



Fast Models Trace Components

Version 11.30

Reference Guide

Non-Confidential

Copyright © 2023–2025 Arm Limited (or its affiliates).
All rights reserved.

Issue 00

107925_1130_00_en



Fast Models Trace Components Reference Guide

This document is Non-Confidential.

Copyright © 2023–2025 Arm Limited (or its affiliates). All rights reserved.

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

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

This document (107925_1130_00_en) was issued on 2025-11-19. There might be a later issue at <https://developer.arm.com/documentation/107925>

The product version is 11.30.

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

Start reading

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

Intended audience

This document is written for software developers using the Fast Models Model Trace Interface (MTI) or writing plug-ins to capture trace from a Fast Models simulation.

Inclusive language commitment

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

This document includes language that can be offensive. We will replace this language in a future issue of this document.

To report offensive language in this document, email terms@arm.com.

Feedback

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

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

Contents

1. Introduction to trace components.....	9
2. Fast Models trace components.....	11
2.1 AHCI_SATA.....	11
2.2 AMBAPV2PVBUS.....	17
2.3 AMBAPVACE2PVBUS.....	20
2.4 AMBAPVSignal2SGSignalx8.....	23
2.5 ARM_AEM-A_MP.....	23
2.6 ARM_AEMv8-R_MP.....	88
2.7 ARM_AEMv8M.....	154
2.8 ARM_C1-Nano.....	203
2.9 ARM_C1-Premium.....	279
2.10 ARM_C1-Pro.....	356
2.11 ARM_C1-Ultra.....	433
2.12 ARM_Cortex-A32.....	510
2.13 ARM_Cortex-A320.....	575
2.14 ARM_Cortex-A34.....	650
2.15 ARM_Cortex-A35.....	715
2.16 ARM_Cortex-A510.....	780
2.17 ARM_Cortex-A520.....	856
2.18 ARM_Cortex-A520AE.....	931
2.19 ARM_Cortex-A53.....	1007
2.20 ARM_Cortex-A55.....	1072
2.21 ARM_Cortex-A57.....	1137
2.22 ARM_Cortex-A65.....	1202
2.23 ARM_Cortex-A65AE.....	1267
2.24 ARM_Cortex-A710.....	1332
2.25 ARM_Cortex-A715.....	1407
2.26 ARM_Cortex-A72.....	1483
2.27 ARM_Cortex-A720.....	1548
2.28 ARM_Cortex-A720AE.....	1623
2.29 ARM_Cortex-A725.....	1699

2.30 ARM_Cortex-A73.....	1774
2.31 ARM_Cortex-A75.....	1839
2.32 ARM_Cortex-A76.....	1904
2.33 ARM_Cortex-A76AE.....	1970
2.34 ARM_Cortex-A77.....	2036
2.35 ARM_Cortex-A78.....	2102
2.36 ARM_Cortex-A78AE.....	2168
2.37 ARM_Cortex-A78C.....	2235
2.38 ARM_Cortex-M0.....	2301
2.39 ARM_Cortex-M0+.....	2345
2.40 ARM_Cortex-M23.....	2390
2.41 ARM_Cortex-M3.....	2435
2.42 ARM_Cortex-M33.....	2465
2.43 ARM_Cortex-M35P.....	2512
2.44 ARM_Cortex-M4.....	2559
2.45 ARM_Cortex-M52.....	2592
2.46 ARM_Cortex-M55.....	2646
2.47 ARM_Cortex-M7.....	2700
2.48 ARM_Cortex-M85.....	2749
2.49 ARM_Cortex-R4.....	2803
2.50 ARM_Cortex-R5.....	2842
2.51 ARM_Cortex-R52Plus.....	2882
2.52 ARM_Cortex-R7.....	2949
2.53 ARM_Cortex-R8.....	3005
2.54 ARM_Cortex-R82.....	3061
2.55 ARM_Cortex-R82AE.....	3127
2.56 ARM_Cortex-X1.....	3194
2.57 ARM_Cortex-X1C.....	3260
2.58 ARM_Cortex-X2.....	3327
2.59 ARM_Cortex-X3.....	3402
2.60 ARM_Cortex-X4.....	3479
2.61 ARM_Cortex-X925.....	3555
2.62 ARM_CortexR52.....	3632
2.63 ARM_Neoverse-E1.....	3700
2.64 ARM_Neoverse-N1.....	3765
2.65 ARM_Neoverse-N2.....	3831

2.66 ARM_Neoverse-N3.....	3915
2.67 ARM_Neoverse-V1.....	3990
2.68 ARM_Neoverse-V2.....	4068
2.69 ARM_Neoverse-V3.....	4152
2.70 ARM_Neoverse-V3AE.....	4237
2.71 ARM_SC000.....	4322
2.72 ARM_SC300.....	4366
2.73 AsyncCacheFlushUnit.....	4396
2.74 Base_PowerController.....	4396
2.75 C1-DSU.....	4397
2.76 CCI400.....	4400
2.77 CCI500.....	4400
2.78 CCI550.....	4407
2.79 CCNCache.....	4414
2.80 CCNRegisterSet.....	4421
2.81 CI700.....	4423
2.82 CMN600.....	4432
2.83 CMN600AE.....	4441
2.84 CMN600CML.....	4452
2.85 CMN600Cache.....	4454
2.86 CMN650.....	4463
2.87 CMN650R2.....	4471
2.88 CMN700.....	4480
2.89 CMNTAGCACHEDADI.....	4489
2.90 CMN_S3.....	4490
2.91 ClockDivider.....	4499
2.92 DMA350.....	4500
2.93 DMC-500.....	4514
2.94 DMC-520.....	4516
2.95 DMC-620.....	4519
2.96 DSU.....	4522
2.97 DSU-110.....	4522
2.98 DSU-120.....	4525
2.99 DualClusterSystemConfigurationBlock.....	4527
2.100 EthosU55.....	4529
2.101 EthosU65.....	4529

2.102 EthosU85.....	4530
2.103 FMU.....	4531
2.104 Firewall.....	4532
2.105 GIC_400.....	4535
2.106 GICv3CPUInterface.....	4540
2.107 GICv3CPUInterfaceDecoder.....	4563
2.108 GICv3CommsLogger.....	4564
2.109 GICv3CommsPVBUS.....	4569
2.110 GICv3Distributor.....	4571
2.111 GICv3InterruptTranslationService.....	4584
2.112 GICv3ProtocolChecker.....	4613
2.113 GICv3Redistributor.....	4613
2.114 GICv3RedistributorInternal.....	4637
2.115 GICv4InterruptTranslationService.....	4646
2.116 GICv4RedistributorInternal.....	4679
2.117 GICv5.....	4688
2.118 IRS.....	4694
2.119 IRSDomain.....	4698
2.120 ITS.....	4723
2.121 ITSDomain.....	4724
2.122 IWB.....	4731
2.123 MMU_400_BASE.....	4736
2.124 MMU_600.....	4777
2.125 MMU_700.....	4879
2.126 MMU_720AE.....	4982
2.127 MMU_L1.....	5115
2.128 MMU_S3.....	5218
2.129 Mali_C10.....	5350
2.130 Mali_C55.....	5350
2.131 Mali_C71.....	5351
2.132 Mali_C720AE.....	5352
2.133 Mali_C78.....	5353
2.134 Mali_G1.....	5353
2.135 Mali_G71.....	5356
2.136 Mali_G710.....	5358
2.137 Mali_G715.....	5361

2.138 Mali_G720.....	5363
2.139 Mali_G725.....	5366
2.140 Mali_G76.....	5368
2.141 Mali_G78AE.....	5371
2.142 Mali_T624.....	5374
2.143 MessageHandlingUnitV2.....	5377
2.144 MessageHandlingUnitV3.....	5380
2.145 MetaDataController.....	5382
2.146 NI700.....	5383
2.147 NI710AE.....	5387
2.148 NOC_S3.....	5392
2.149 PL011_Uart.....	5397
2.150 PL310_L2CC.....	5398
2.151 PPUv0.....	5401
2.152 PPUv1.....	5402
2.153 PVBUS2AMBAPV.....	5404
2.154 PVBUS2AMBAPVACE.....	5407
2.155 PVBUSBridge.....	5411
2.156 PVBUSExclusiveMonitor.....	5411
2.157 PVBUSGICv3Comms.....	5412
2.158 PVBUSLogger.....	5414
2.159 PVBUSMapper.....	5417
2.160 PVBUSMaster.....	5417
2.161 PVBUSSlave.....	5417
2.162 PVCACHE.....	5419
2.163 PVCACHE64.....	5428
2.164 PVMemoryProtectionEngine.....	5437
2.165 PVWriteBuffer.....	5438
2.166 SMMUv3AEM.....	5440
2.167 SMMUv3TestEngine.....	5541
2.168 SSU.....	5549
2.169 SYSTEM_FMU.....	5550
2.170 SharedR.....	5552
2.171 SignalDriver.....	5553
2.172 SignalLogger.....	5553
2.173 TLB.....	5554

2.174 Value64Logger..... 5559

2.175 ValueLogger..... 5560

2.176 VirtioBlockDevice..... 5560

2.177 VirtioPCIBlockDevice..... 5562

2.178 VirtualEthernetCrossover..... 5565

2.179 atc..... 5565

2.180 v7_VGIC..... 5580

Proprietary notice..... 5585

Product and document information..... 5587

Product status..... 5587

Revision history..... 5587

Conventions..... 5588

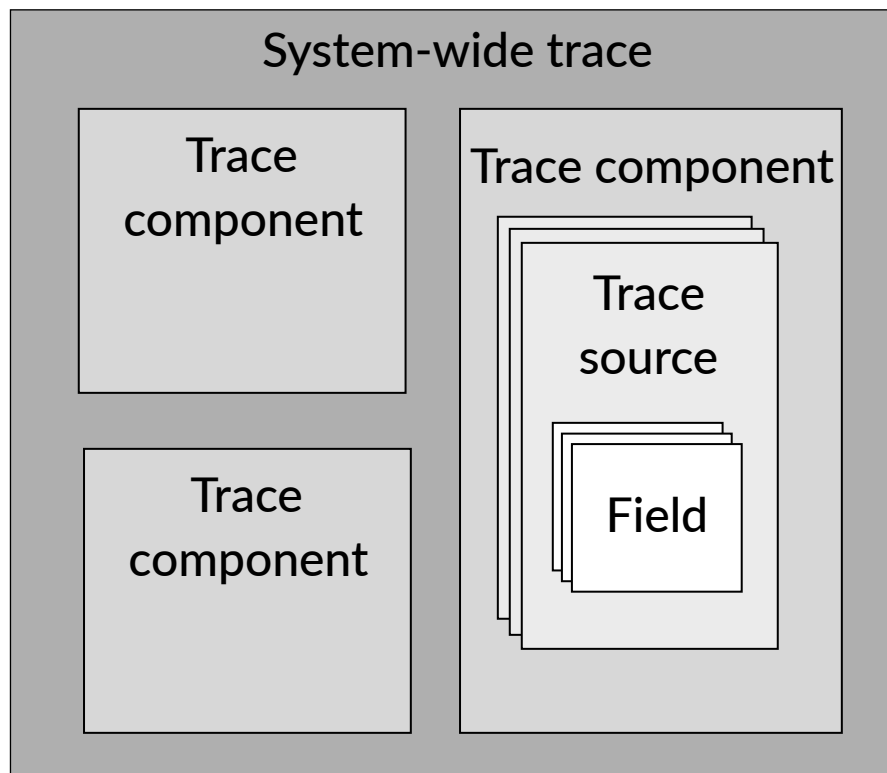
Useful resources..... 5590

1. Introduction to trace components

Fast Models provides an interface called Model Trace Interface (MTI) which enables the generation and capture of trace events during a simulation.

MTI represents trace using a hierarchy, as shown in this figure:

Figure 1-1: MTI hierarchy



Each platform contains a set of trace components. Each trace component can contain one or more trace sources. Each trace source can contain one or more data fields.

A trace source provides information about a specific event that occurred, for example an instruction that was executed, a cache miss, or an MMU translation. It can also be a model-specific event, such as a `sync` event, which occurs at every quantum boundary. While the simulation is running, each trace source generates a stream of trace events.

The fields within the trace source give more information about the event, including a text description. When the field is of type `MTI_ENUM`, the values for this field are also listed.

For example, the `pvcache` trace component has several trace sources, one of which is `CACHE_READ_MISS`, which has the following fields:

```
Source CACHE_READ_MISS (Read access cache miss and cache miss latency.)
  Field IS_SHARED type:MTI_ENUM size:1 (Is the access shared)
    0x0 = NON_SHARED
    0x1 = SHARED
  Field IS_PRELOAD type:MTI_ENUM size:1 (Is the access a preload)
    0x0 = NOT_PRELOAD
    0x1 = PRELOAD
  Field MANAGER_ID type:MTI_UNSIGNED_INT size:4 (Indicates the manager ID
associated with this monitor)
  Field LATENCY type:MTI_UNSIGNED_INT size:8 (Cache miss latency in ticks per
access)
```

Most consumers of trace events are MTI plug-ins, which might respond to events by logging them, increasing a counter, or updating the state of the model. Arm provides several pre-built MTI plug-ins for capturing trace or for listing the available trace sources. They are documented in [Plug-ins for Fast Models](#) in *Fast Models Reference Guide*. Source code for some of these plug-ins and for other example plug-ins is provided in `$PVLIB_HOME/examples/MTI/`.

2. Fast Models trace components

This chapter describes all trace components in Fast Models.

2.1 AHCI_SATA

This section describes the trace sources.

AHCI_REG_GHC_READ

GHC Register read access to AHCI. Fields:

ADDRESS unsigned int

Address Offset within GHC space.

DATA unsigned int

Data returned by this register.

GHC_REGISTER enum

Name of the GHC register.

AHCI_REG_GHC_WRITE

GHC Register write access to AHCI. Fields:

ADDRESS unsigned int

Address Offset within GHC space.

DATA unsigned int

Data written to this register.

GHC_REGISTER enum

Name of the GHC register.

AHCI_REG_PORT_READ

PORT Register read access to AHCI. Fields:

ADDRESS unsigned int

Address Offset within PORTx space.

DATA unsigned int

Data returned by this register.

PORT unsigned int

SATA port number.

PORTx_REGISTER enum

Name of the PORTx register.

AHCI_REG_PORT_WRITE

PORT Register write access to AHCI. Fields:

ADDRESS unsigned int

Address Offset within PORTx space.

DATA unsigned int

Data written to this register.

PORT unsigned int

SATA port number.

PORTx_REGISTER enum

Name of the PORTx register.

ATA_COMMAND

ATA command received by SATA device. Fields:

COMMAND enum

ATA Command.

FIS_COUNT unsigned int

Raw value of the Count field in the FIS. (Sector count for non-NCQ commands.).

FIS_DEVICE unsigned int

Raw value of the Device fields on the FIS. Bit 6 indicates LBA (28 or 48 bit access).

FIS_FEATURE unsigned int

Raw value of the Feature field in the FIS. (Sector count for NCQ commands.).

LBA unsigned int

Starting sector index.

PORT unsigned int

SATA port number (0-31).

SECTOR_COUNT unsigned int

Sector count.

ATA_COMMAND_HEADER

ATA Command header. Fields:

CFIS_LEN unsigned int

CFIS length.

CMD_TABLE_BASE unsigned int

Command table Base.

IS_WRITE bool

Write?.

PORT unsigned int

SATA port number (0-31).

PRD_COUNT unsigned int

Number of PRDs.

PRD_TABLE_LEN unsigned int

PRD table length.

ArchMsg.Error.invalid_access_width

Access with invalid access width in AHCI. Fields:

ACCESS_WIDTH unsigned int

Access Width to a register.

ADDRESS unsigned int

Address Offset within GHC space.

DATA unsigned int

Data with access width of 32 bits.

ArchMsg.Error.invalid_register

Access to invalid register offsets in AHCI. Fields:

ADDRESS unsigned int

Address Offset within GHC or port space.

PORT unsigned int

SATA port number or 0xff for GHC.

CFIS

Command FIS. Fields:

CFIS unsigned int

CFIS content.

PORT unsigned int

SATA port number (0-31).

CMD_SLOT

Command Slot in Command List. Fields:

CMD_SLOT unsigned int

Command Slot.

PORT unsigned int

SATA port number (0-31).

DATA_READ

Data read from ahci_dma_m. Fields:

ADDRESS unsigned int

Address.

DATA unsigned int

Data Read.

PORT unsigned int

SATA port number (0-31).

DATA_READ_ABORT

The data read by the device aborted. Fields:

ADDR **unsigned int**

Address.

NS **bool**

Non-secure access.

PORT **unsigned int**

SATA port number (0-31).

SIZE **unsigned int**

The data size in bytes.

DATA_WRITE

Data write from ahci_dma_m. Fields:

ADDRESS **unsigned int**

Address.

DATA **unsigned int**

Data Write.

PORT **unsigned int**

SATA port number (0-31).

DATA_WRITE_ABORT

The data write by the device aborted. Fields:

ADDR **unsigned int**

Address.

NS **bool**

Non-secure access.

PORT **unsigned int**

SATA port number (0-31).

SIZE **unsigned int**

The data size in bytes.

FIS

FIS received by AHCI controller from SATA device. Fields:

DATA **unsigned int**

FIS content (first byte is the FIS type).

PORT **unsigned int**

SATA port number (0-31).

MSI_INTERRUPT

MSI interrupt generated by AHCI. Fields:

GHC_IS unsigned int

Contents of GHC.IS register.

MSI_VECTOR unsigned int

MSI vector number.

PORT_INTERRUPT

Interrupt generated by port. This is emitted only when the corresponding interrupt is enabled in PxlE, but regardless whether interrupts are globally enabled in GHC.IE. Fields:

INTRRUPT_TYPE enum

Interrupt type.

PORT unsigned int

SATA port number (0-31).

PRDT_ENTRY

PRDT_ENTRY. Fields:

ADDR unsigned int

PRD entry address.

COUNT unsigned int

Bytes for this PRD entry.

IRQ bool

Raise an interrupt upon completion.

PORT unsigned int

SATA port number (0-31).

SATA_BS_PERF

SATA backend storage periodic performance indicator. This allows to monitor the data throughput in terms of host time. This emits a trace event, at most once per second, which indicates the data rate for reads, writes and the sum (throughput) in the interval since the last event. An event is only generated when there was at least one read/write access in the last second. Fields:

PERF string

Throughput/read/write performance as a formatted string.

SATA_PORT unsigned int

Index of the SATA port this backend storage is attached to.

SATA_BS_READ

SATA backend storage read. Fields:

SATA_PORT unsigned int

Index of the SATA port this backend storage is attached to.

SECTOR_COUNT unsigned int

Number of sectors read (512 byte sectors).

SECTOR_INDEX unsigned int

Access starts at this sector index (512 byte sectors).

SATA_BS_READ_DATA

SATA backend storage read. This trace is issued for each individual 512-byte sector of the original access size. Fields:

DATA unsigned int

Sector Data (always 512 bytes).

SATA_PORT unsigned int

Index of the SATA port this backend storage is attached to.

SECTOR_INDEX unsigned int

Sector index of this data (512 byte sectors).

SATA_BS_TRIM

SATA backend storage trim (marking sectors as unused). Fields:

SATA_PORT unsigned int

Index of the SATA port this backend storage is attached to.

SECTOR_COUNT unsigned int

Number of sectors read (512 byte sectors).

SECTOR_INDEX unsigned int

Access starts at this sector index (512 byte sectors).

SATA_BS_WRITE

SATA backend storage write. Fields:

SATA_PORT unsigned int

Index of the SATA port this backend storage is attached to.

SECTOR_COUNT unsigned int

Number of sectors read (512 byte sectors).

SECTOR_INDEX unsigned int

Access starts at this sector index (512 byte sectors).

SATA_BS_WRITE_DATA

SATA backend storage write. This trace is issued for each individual 512-byte sector of the original access size. Fields:

DATA unsigned int

Sector Data (always 512 bytes).

SATA_PORT unsigned int

Index of the SATA port this backend storage is attached to.

SECTOR_INDEX unsigned int

Sector index of this data (512 byte sectors).

SET_FEATURE

SET FEATURE. Fields:

PARAMETER unsigned int

Sub command parameter.

PORT unsigned int

SATA port number (0-31).

SUB_COMMAND enum

Set Feature sub command.

2.2 AMBAPV2PVBUS

This section describes the trace sources.

READ_ACCESS

Logger. Fields:

ACCESS_SIZE unsigned int

Log2 of access size (i.e. 0=byte, 1=halfword, 2=word, 3=doubleword ...).

ACE enum

Encodes the ACE operation.

ATTR unsigned int

Transaction Attributes: [12] Instruction fetch [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

DATA unsigned int

The data transferred.

EXTENDED_ID unsigned int

The Extended ID.

LATENCY unsigned int

Time downstream (ticks).

MANAGER_ID unsigned int

The AXI Manager ID.

NUMBER_OF_BEATS unsigned int

The number of data transfers (beats) in this burst.

PADDR unsigned int

Physical address of access.

RESPONSE enum

Whether the transaction was successful, or an error occurred.

USER_FLAGS unsigned int

Core specific additional signals.

READ_ACCESS_START

Logger. Fields:

ACCESS_SIZE unsigned int

Log2 of access size (i.e. 0=byte, 1=halfword, 2=word, 3=doubleword ...).

ACE enum

Encodes the ACE operation.

ATTR unsigned int

Transaction Attributes: [12] Instruction fetch [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

DATA unsigned int

The data transferred.

EXTENDED_ID unsigned int

The Extended ID.

MANAGER_ID unsigned int

The AXI Manager ID.

NUMBER_OF_BEATS unsigned int

The number of data transfers (beats) in this burst.

PADDR unsigned int

Physical address of access.

USER_FLAGS unsigned int

Core specific additional signals.

WRITE_ACCESS

Logger. Fields:

ACCESS_SIZE unsigned int

Log2 of access size (i.e. 0=byte, 1=halfword, 2=word, 3=doubleword ...).

ACE enum

Encodes the ACE operation.

ATTR unsigned int

Transaction Attributes: [12] Instruction fetch [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

DATA unsigned int

The data transferred.

EXTENDED_ID unsigned int

The Extended ID.

LATENCY unsigned int

Time downstream (ticks).

MANAGER_ID unsigned int

The AXI Manager ID.

NUMBER_OF_BEATS unsigned int

The number of data transfers (beats) in this burst.

PADDR unsigned int

Physical address of access.

RESPONSE enum

Whether the transaction was successful, or an error occurred.

USER_FLAGS unsigned int

Core specific additional signals.

WRITE_ACCESS_START

Logger. Fields:

ACCESS_SIZE unsigned int

Log2 of access size (i.e. 0=byte, 1=halfword, 2=word, 3=doubleword ...).

ACE enum

Encodes the ACE operation.

ATTR unsigned int

Transaction Attributes: [12] Instruction fetch [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

DATA unsigned int

The data transferred.

EXTENDED_ID unsigned int

The Extended ID.

MANAGER_ID unsigned int

The AXI Manager ID.

NUMBER_OF_BEATS unsigned int

The number of data transfers (beats) in this burst.

PADDR unsigned int

Physical address of access.

USER_FLAGS unsigned int

Core specific additional signals.

2.3 AMBAPVACE2PVBUS

This section describes the trace sources.

READ_ACCESS

Logger. Fields:

ACCESS_SIZE unsigned int

Log2 of access size (i.e. 0=byte, 1=halfword, 2=word, 3=doubleword ...).

ACE enum

Encodes the ACE operation.

ATTR unsigned int

Transaction Attributes: [12] Instruction fetch [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

DATA unsigned int

The data transferred.

EXTENDED_ID unsigned int

The Extended ID.

LATENCY unsigned int

Time downstream (ticks).

MANAGER_ID unsigned int

The AXI Manager ID.

NUMBER_OF_BEATS unsigned int

The number of data transfers (beats) in this burst.

PADDR unsigned int

Physical address of access.

RESPONSE enum

Whether the transaction was successful, or an error occurred.

USER_FLAGS unsigned int

Core specific additional signals.

READ_ACCESS_START

Logger. Fields:

ACCESS_SIZE unsigned int

Log2 of access size (i.e. 0=byte, 1=halfword, 2=word, 3=doubleword ...).

ACE enum

Encodes the ACE operation.

ATTR unsigned int

Transaction Attributes: [12] Instruction fetch [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

DATA unsigned int

The data transferred.

EXTENDED_ID unsigned int

The Extended ID.

MANAGER_ID unsigned int

The AXI Manager ID.

NUMBER_OF_BEATS unsigned int

The number of data transfers (beats) in this burst.

PADDR unsigned int

Physical address of access.

USER_FLAGS unsigned int

Core specific additional signals.

WRITE_ACCESS

Logger. Fields:

ACCESS_SIZE unsigned int

Log2 of access size (i.e. 0=byte, 1=halfword, 2=word, 3=doubleword ...).

ACE enum

Encodes the ACE operation.

ATTR unsigned int

Transaction Attributes: [12] Instruction fetch [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

DATA unsigned int

The data transferred.

EXTENDED_ID unsigned int

The Extended ID.

LATENCY unsigned int

Time downstream (ticks).

MANAGER_ID unsigned int

The AXI Manager ID.

NUMBER_OF_BEATS unsigned int

The number of data transfers (beats) in this burst.

PADDR unsigned int

Physical address of access.

RESPONSE enum

Whether the transaction was successful, or an error occurred.

USER_FLAGS unsigned int

Core specific additional signals.

WRITE_ACCESS_START

Logger. Fields:

ACCESS_SIZE unsigned int

Log2 of access size (i.e. 0=byte, 1=halfword, 2=word, 3=doubleword ...).

ACE enum

Encodes the ACE operation.

ATTR unsigned int

Transaction Attributes: [12] Instruction fetch [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

DATA unsigned int

The data transferred.

EXTENDED_ID unsigned int

The Extended ID.

MANAGER_ID unsigned int

The AXI Manager ID.

NUMBER_OF_BEATS unsigned int

The number of data transfers (beats) in this burst.

PADDR unsigned int

Physical address of access.

USER_FLAGS unsigned int

Core specific additional signals.

2.4 AMBAPVSignal2SGSignalx8

This section describes the trace sources.

ArchMsg.Warning.AMBAPVSignal2SGSignal_unconnected_warning

Unconnected Warning. DISPLAY An unconnected signal with index: %{index} and export_id: %{export_id} was toggled to state: %{state}. Fields:

export_id signed int

Received export_id.

index unsigned int

The index of unconnected toggled signal.

state bool

The toggled to Signal state.

2.5 ARM_AEM-A_MP

This section describes the trace sources.

AA64_ASE_SVE_REGS

Modification of AArch64 Advanced SIMD or SVE Vector Registers (via SIMD or Scalar operations), SVE Predicate Registers and ZTO. Fields:

ID enum

The register identifier.

MASK unsigned int

Bitmask of the register to signal the modified bits in the VALUE field.

SM bool

Whether the PE is in Streaming Mode.

VALUE unsigned int

Value of AArch64 vector register.

ASYNC_MEMORY_FAULT

Context ID Register write. Fields:

FAULT unsigned int

Fault status present in ESR.ISS encoding for Serror in ARCH64 and present in DFSR encoding for external abort format in ARCH32.

PADDR unsigned int

Physical Address (or 0 if unavailable).

VADDR signed int

Virtual Address (or 0 if unavailable).

ATOMIC_END_ACCESS

Bus trace access for atomics. Fields:

ACCESS_FAIL bool

Memory access failed.

ADDR unsigned int

The virtual address of the access.

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

COMPARE_VALUE unsigned int

Compare value for CAS.

LOAD_VALUE unsigned int

Loaded values.

NSDESC unsigned int

The security state of the access.

OPERAND_VALUE unsigned int

Operation's operand.

OPERATION enum

Operation type.

PADDR unsigned int

The physical address of the access.

PRIV bool

Is this a privileged access?.

ATOMIC_START_ACCESS

Bus trace access for atomics. Fields:

ADDR unsigned int

The virtual address of the access.

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

COMPARE_VALUE unsigned int

Compare value for CAS.

NSDESC unsigned int

The security state of the access.

OPERAND_VALUE unsigned int

Operation's operand.

OPERATION enum

Operation type.

PADDR unsigned int

The physical address of the access.

PRIV bool

Is this a privileged access?.

ArchMsg.Error.exit_code

DISPLAY exit code: %{EXITCODE} component kind: %{KIND} reason: %{REASON}. Fields:

EXITCODE signed int

Exit code to be returned.

KIND string

Component kind that invoked the exit code trace.

REASON string

Reason why the exit code trace was invoked (optional).

ArchMsg.Warning.IgnoredWarmResetRequest

Event fires when software attempts to request warm reset through write to deprecated DBGPRCR.CWRR.

ArchMsg.Warning.ReportedRevisionRegisterRead

DISPLAY Reported revision and patch do not match that of the cpu. Fields:

ORIGINAL_VALUE unsigned int

The value the register would have based on the modelled revision.

REGISTER string

The name of the register that has ben read.

REPORTED_VALUE unsigned int

The value the register has been overridden to.

ArchMsg.Warning.Unpredictably Indexed PM Event Register

TAGS **UNPREDICTABLE** DISPLAY Event %{IsDirect:(Selector value|Register)} %{SEL} is >= %{N}. Fields:

IsDirect unsigned int

Direct Access rather than indirect.

N unsigned int

Number of Accessible Registers.

SEL unsigned int

Selector.

ArchMsg.Warning.cache_contents_unknown

A cache has been activated without being invalidated and may contain **UNKNOWN** content. DISPLAY %{SIDE:(D|I)}-Cache was enabled in %{REGIME:(SECURE|NON_SECURE|HYP)} regime but was not invalidated since power-on: cache lines could contain **UNKNOWN** content %{SIDE:(Note: D-side caches must invalidate SECURE lines even if the cache is only used by NON_SECURE code in order to ensure that unexpected dirty lines tagged SECURE are not naturally evicted)}. Fields:

REGIME unsigned int

Current translation regime.

SIDE unsigned int

0 d-side cache or unified cache, 1 i-side.

ArchMsg.Warning.dcimvac_matches_watchpoint

TAGS **UNPREDICTABLE** DISPLAY Watchpoints matching an AArch32 DCIMVA are implementation defined.

ArchMsg.Warning.decode_fieldmismatch

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with different fields %{FIELD1} and %{FIELD2}. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_fracbitsnegative

TAGS **UNPREDICTABLE** DISPLAY VCVT Instruction is unpredictable with a negative number of fraction bits.

ArchMsg.Warning.decode_initblock

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable in an IT block.

ArchMsg.Warning.decode_initblocknotlast

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable in an IT block when not the last.

ArchMsg.Warning.decode_invalid_condition

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable when the condition is not AL.

ArchMsg.Warning.decode_invalidfieldcombination

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with the values in %{FIELD1} and %{FIELD2}. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_invalidvalue

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with the value in %{FIELD}. Fields:

FIELD enum

Field name.

ArchMsg.Warning.decode_mem_hint_unallocated

TAGS **UNPREDICTABLE** DISPLAY There is no memory hint allocated to this bit pattern.

ArchMsg.Warning.decode_registermatch

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with %{FIELD1} and %{FIELD2} as the same register. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_registermismatch

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with %{FIELD1} and %{FIELD2} as different registers. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_registeroutofrange

TAGS **UNPREDICTABLE** DISPLAY Floating-point instruction is unpredictable indexing beyond the end of the register bank.

ArchMsg.Warning.decode_sbzsbo

TAGS **UNPREDICTABLE** DISPLAY Reserved bits in the instruction are not canonical.

ArchMsg.Warning.decode_transactiontoobig

TAGS **UNPREDICTABLE** DISPLAY Attempt to perform a memory transaction of more than 128 bytes.

ArchMsg.Warning.decode_unpred_other

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable.

ArchMsg.Warning.decode_unpreduseofpc

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable when %{FIELD} is PC. Fields:

FIELD enum

Field name.

ArchMsg.Warning.decode_unpreduseofr13

TAGS UNPREDICTABLE DISPLAY Instruction is unpredictable when %{FIELD} is R13. Fields:

FIELD enum

Field name.

ArchMsg.Warning.decode_unpreduseofr13inreglist

TAGS UNPREDICTABLE DISPLAY Instruction is unpredictable with R13 in register-list.

ArchMsg.Warning.decode_useofd16orhigher

TAGS UNPREDICTABLE DISPLAY Attempt to use register D16 or higher where it is not allowed.

ArchMsg.Warning.decode_writebackandbaseinlist

TAGS UNPREDICTABLE DISPLAY Instruction is unpredictable with write back when the base register is in the transfer list.

ArchMsg.Warning.decode_zeroregistersinlist

TAGS UNPREDICTABLE DISPLAY Instruction is unpredictable when no registers are to be transferred.

ArchMsg.Warning.recursive_branch

An instruction is performing a branch that targets the same instruction. If this was intended then a WFI might be more effective DISPLAY recursive branch detected at PC=%{PC}.

PRIMARY KEY PC. Fields:

PC unsigned int

Address of instruction causing the branch.

ArchMsg.Warning.recursive_exception

An instruction has generated an exception that targets the same instruction. DISPLAY recursive %{TYPE} exception detected at PC=%{PC} (CPSR=%{CPSR}). PRIMARY KEY PC. Fields:

CPSR unsigned int

Processor state.

PC unsigned int

Address of instruction causing the exception.

TYPE enum

Type of exception.

ArchMsg.Warning.reentrant_vector_catch

TAGS UNPREDICTABLE DISPLAY Vector catch debug event on Prefetch or Data abort vector %{ADDR}. Fields:

ADDR unsigned int

vector address.

ArchMsg.Warning.reserved_it_state

TAGS UNPREDICTABLE DISPLAY Execution with reserved IT state %{ITSTATE}. Fields:

ITSTATE unsigned int

reserved IT state.

ArchMsg.Warning.tlb_contents_unknown

Translation has been enabled without invalidating the TLB. DISPLAY Translation was enabled in the `%{PAS:(Secure|Non-secure|Root|Realm)} EL%{EL:d}` translation regime (`vmid=%{VMID:x}`) without invalidating `%{INVALIDITY:(|d-side |i-side |)}` entries since power-on: `%{INVALIDITY:(|D|I)}` TLB could contain **UNKNOWN** content. Fields:

EL unsigned int

Exception level of translation regime.

INVALIDITY unsigned int

Which TLB may be invalid (if TLBs are separate).

NON_SECURE bool

Whether the translation regime is non-secure.

PAS unsigned int

Physical address space of translation regime.

VMID unsigned int

Current VMID, if applicable.

ArchMsg.Warning.unknown_DLR_DSPSR

Exiting debug state with unknown DLR or DSPSR.

ArchMsg.Warning.unknown_ELR_SPSR

Returning from debug exception with unknown ELR or SPSR.

ArchMsg.Warning.unpred_FAR_on_LRCPC3_non_atomic_mem_access

TAGS **UNPREDICTABLE** Unpredictable FAR/FSTATUS on LRCPC3 pair instructions resulting in two separate memory accesses. DISPLAY Unpredictable FAR on LRCPC3 non atomic memory access at `%{ADDR1}` with FSTATUS `%{FSTATUS1}` or `%{ADDR2}` with FSTATUS `%{FSTATUS2}`. Fields:

ADDR1 unsigned int

Possible FAR address 1.

ADDR2 unsigned int

Possible FAR address 2.

FSTATUS1 unsigned int

Possible fault status value 1.

FSTATUS2 unsigned int

Possible fault status value 2.

ArchMsg.Warning.unpred_async_tag_check_fail_on_sve_speculative_load_do_update_ffr

Unpredictable set to FALSE of the FFR element associated to SVE speculative load with an async tag check fail TAGS **UNPREDICTABLE**.

ArchMsg.Warning.unpredictable_a32_breakpoint

TAGS **UNPREDICTABLE** DISPLAY Unpredictable breakpoint on a A32 instruction at %{ADDR1} Offset %{OFFSET} BAS %{BAS} for a %{IS_ADDRESS_MISMATCH_BREAKPOINT:(Mismatch|Match)} Breakpoint number %{N:d} Breakpoint address %{ADDR2}. Fields:

ADDR1 unsigned int

Address.

ADDR2 unsigned int

Breakpoint address.

BAS unsigned int

BAS field.

IS_ADDRESS_MISMATCH_BREAKPOINT bool

Is a Mismatch breakpoint, not a Match breakpoint.

N unsigned int

Breakpoint number.

OFFSET unsigned int

Always has a value of zero.

ArchMsg.Warning.unpredictable_banked_register_access

TAGS **UNPREDICTABLE** DISPLAY Access to banked register (R=%{R:d} SYSm=%{SYSM:2x}) is unpredictable. Fields:

R unsigned int

R bit.

SYSM unsigned int

SYSm.

ArchMsg.Warning.unpredictable_ccfail_undef

TAGS **UNPREDICTABLE** DISPLAY An instruction %{OPCODE} at VA=%{VADDR} failed its condition codes check but would otherwise have caused a non-data-dependent trap or undefined exception. Behaviour is implementation-defined choice of exception or nop. Fields:

OPCODE unsigned int

opcode of the instruction.

VADDR unsigned int

Virtual address of the instruction being executed.

ArchMsg.Warning.unpredictable_ici_bits

TAGS **UNPREDICTABLE** DISPLAY ICI bits do not apply or are architecturally unpredictable for instruction.

ArchMsg.Warning.unpredictable_t32_breakpoint

TAGS **UNPREDICTABLE** DISPLAY Unpredictable breakpoint on a T32 instruction at %{ADDR1} Offset %{OFFSET} BAS %{BAS} for a %{IS_ADDRESS_MISMATCH_BREAKPOINT:(Mismatch|Match)} Breakpoint number %{N:d} Breakpoint address %{ADDR2}. Fields:

ADDR1 unsigned int

Address.

ADDR2 unsigned int

Breakpoint address.

BAS unsigned int

BAS field.

IS_ADDRESS_MISMATCH_BREAKPOINT bool

Is a Mismatch breakpoint, not a Match breakpoint.

N unsigned int

Breakpoint number.

OFFSET unsigned int

Indicates which halfword of a 32-bit Thumb instruction the breakpoint is set on. 0 for the first halfword or if the instruction is 16 bits wide, or 1 for the second halfword.

ArchMsg.Warning.unpredictable_t32_vector_catch

TAGS **UNPREDICTABLE** DISPLAY Unpredictable Vector catch exception on a T32 instruction at %{ADDR} Offset %{OFFSET}. Fields:

ADDR unsigned int

Address.

OFFSET unsigned int

Indicates which halfword of a 32-bit Thumb instruction the vector catch exception is set on. 0 for the first halfword or if the instruction is 16 bits wide, or 1 for the second halfword.

ArchMsg.Warning.unpredictable_watchpoint_far

TAGS **UNPREDICTABLE** DISPLAY The value for FAR/EDWAR for hitting this watchpoint may be anywhere in the range %{LOWER_BOUND} to %{UPPER_BOUND}. Fields:

LOWER_BOUND unsigned int

the lowest address accessed by the instruction that triggered the watchpoint.

UPPER_BOUND unsigned int

the highest watchpointed address accessed by that instruction.

ArchMsg.Warning.unpredictable_watchpoint_far_isscmp

DISPLAY The value for FAR/EDWAR for hitting this watchpoint may be anywhere in the range %{LOWER_BOUND} to %{UPPER_BOUND}. Fields:

LOWER_BOUND unsigned int

the lowest address accessed by the instruction that triggered the watchpoint.

UPPER_BOUND unsigned int

the highest watchpointed address accessed by that instruction.

ArchMsg.Warning.warning_AdvSIMDExpandImmUnexpectedZero

TAGS **UNPREDICTABLE** DISPLAY AdvSIMDExpandImm may treat this immediate value as **UNPREDICTABLE**.

ArchMsg.Warning.warning_ConditionalSMC

TAGS **UNPREDICTABLE** DISPLAY SMC instruction has **UNPREDICTABLE** effects when conditional and when combined with traps.

ArchMsg.Warning.warning_TLBID_out_of_range

TLBID out of range in TLBI according to the configuration of NIS, NVIS, NOS and NVOS. Fields:

NIS unsigned int

NIS.

NOS unsigned int

NOS.

NVIS unsigned int

NVIS.

NVOS unsigned int

NVOS.

TLBID unsigned int

The TLBID value used in the TLBI instruction.

ArchMsg.Warning.warning_TTL_or_TG_mismatches_current_translation_granule

TTL or TG mismatches current translation granule in TLBI VA/RVA/IPA/RIPA. Fields:

CURRENT_TG unsigned int

Current TG as indicated by TCR_ELx or VTCR_EL2.

TLBI_TG unsigned int

TG (or TTL[3:2]) used in TLBI instruction.

ArchMsg.Warning.warning_access_crosses_page_boundary

TAGS **UNPREDICTABLE** DISPLAY Access crosses page boundary.

ArchMsg.Warning.warning_access_crosses_page_boundary_has_fault

TAGS **UNPREDICTABLE** DISPLAY Access that crosses page boundary has fault(s).

ArchMsg.Warning.warning_access_crosses_page_boundary_has_fault_on_both_pages

TAGS **UNPREDICTABLE** DISPLAY Access that crosses page boundary has a fault on each page. Fields:

ADDR1 unsigned int

FAR address on first page.

ADDR2 unsigned int

FAR address on second page.

FSTATUS1 unsigned int

Fault status on first page.

FSTATUS2 unsigned int

Fault status on second page.

ArchMsg.Warning.warning_access_crosses_page_boundary_spanning_different_memory

TAGS **UNPREDICTABLE** DISPLAY Access at address %{ADDR} crosses page from MEM_TYPE %{MEMTYPE_PAGE1} to %{MEMTYPE_PAGE2}. Fields:

ADDR unsigned int

address of access crossing page.

MEMTYPE_PAGE1 signed int

First Page's memory type (0-Normal, 1-Non-Normal).

MEMTYPE_PAGE2 signed int

Second Page's memory type (0-Normal, 1-Non-Normal).

ArchMsg.Warning.warning_access_wraps_around_memory

TAGS **UNPREDICTABLE** DISPLAY Access wraps around memory in %{WIDTH:d}bit mode. Fields:

WIDTH unsigned int

Address width.

ArchMsg.Warning.warning_bcr_linking_status

TAGS **UNPREDICTABLE** DISPLAY DBGBCR.{BT2,BT} combination gives unpredictable behaviour.

ArchMsg.Warning.warning_bcr_mask_reserved

TAGS **UNPREDICTABLE** DISPLAY DBGBCR_ELn.MASK is set to a reserved value.

ArchMsg.Warning.warning_bvr_masked_bit_not_zero

TAGS **UNPREDICTABLE** DISPLAY DBGBVRn_EL1.VA contains a bit masked by DBGBCRn_EL1.MASK that is not zero.

ArchMsg.Warning.warning_ccsidr_unimplemented_level

TAGS **UNPREDICTABLE** DISPLAY CCSIDR read while CSSELR %{CSSELR:x} points to an unimplemented cache level. Fields:

CSSELR unsigned int

current effective value of CSSELR.

ArchMsg.Warning.warning_change_to_ns_when_tge_set

TAGS **UNPREDICTABLE** DISPLAY Attempting to change NS to 1 when HCR.TGE = 1.

ArchMsg.Warning.warning_contiguous_bit_check_abort

TAGS **UNPREDICTABLE** DISPLAY An abort occurred whilst attempting to check TLB entry is contiguous at %{ENTRY_ADDR}. Fields:

ENTRY_ADDR unsigned int

address of conflicting TLB entry.

ArchMsg.Warning.warning_contiguous_bit_error

TAGS **UNPREDICTABLE** DISPLAY TLB entry at %{ENTRY_ADDR} was not contiguous with entry at %{CONTIG_ADDR} Contents are %{ENTRY_DATA} but expected %{CONTIG_DATA}. Fields:

CONTIG_ADDR unsigned int

address of the TLB first read, with which this entry is expected to be contiguous.

CONTIG_DATA unsigned int

expected contents based on the entry at CONTIG_ADDR.

ENTRY_ADDR unsigned int

address of conflicting TLB entry.

ENTRY_DATA unsigned int

contents of conflicting TLB entry.

ArchMsg.Warning.warning_contiguous_bit_error_D128

TAGS **UNPREDICTABLE** DISPLAY TLB entry at %{ENTRY_ADDR} was not contiguous with entry at %{CONTIG_ADDR} Contents are %{ENTRY_DATA_HI} %{ENTRY_DATA_LO} but expected %{CONTIG_DATA}. Fields:

CONTIG_ADDR unsigned int

address of the TLB first read, with which this entry is expected to be contiguous.

CONTIG_DATA unsigned int

expected contents based on the entry at CONTIG_ADDR.

ENTRY_ADDR unsigned int

address of conflicting TLB entry.

ENTRY_DATA_HI unsigned int

Higher bit contents of conflicting TLB entry.

ENTRY_DATA_LO unsigned int

Lower bit contents of conflicting TLB entry.

ArchMsg.Warning.warning_cp10_cp11_mismatch

TAGS **UNPREDICTABLE** DISPLAY Unpredictable configuration of CP10 and CP11 controls at EL %{EL:d}. Fields:

EL unsigned int

EL owning the control.

ArchMsg.Warning.warning_cp10_cp11_reserved_value

TAGS **UNPREDICTABLE** DISPLAY Unpredictable configuration of CPACR CP10 (%{CP10}) or CP11 (%{CP11}). Fields:

CP10 unsigned int

CP10 access.

CP11 unsigned int

CP11 access.

ArchMsg.Warning.warning_csselr_level_out_of_range

TAGS **UNPREDICTABLE** DISPLAY CSSELR.Level out of range, written value %{LEVEL} with only %{IMPLEMENTED:d} levels implemented. Fields:

IMPLEMENTED unsigned int

number of cache levels implemented.

LEVEL unsigned int

written value of CSSELR.Level.

TYPE enum

The type of cache selected by CSSELR.InD and TnD.

ArchMsg.Warning.warning_debug_flow_control_bits_not_obeyed

TAGS **UNPREDICTABLE** DISPLAY Flow control bits for %{REGISTER} were not obeyed, causing %{ERROR}. Fields:

ERROR enum

type of error resulting.

REGISTER enum

register under consideration.

ArchMsg.Warning.warning_debug_register_access_during_reset

TAGS **UNPREDICTABLE** DISPLAY Unpredictable %{IS_WRITE:(read|write)} access to debug register offset %{OFFSET} during reset. Fields:

IS_WRITE unsigned int

Write Not Read.

OFFSET unsigned int

Register Offset.

ArchMsg.Warning.warning_decode_cps_inconsistent_fields

TAGS **UNPREDICTABLE** DISPLAY The mode setting fields in the CPS instruction are inconsistent.

ArchMsg.Warning.warning_decode_invalid_state

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable in current exception-level/security state.

ArchMsg.Warning.warning_default_cacheable_mmu_on

TAGS **UNPREDICTABLE** DISPLAY Default cacheable is enabled (HCR.DC = 1) while MMU is enabled.

ArchMsg.Warning.warning_default_cacheable_vmmu_off

TAGS **UNPREDICTABLE** DISPLAY Default cacheable is enabled (HCR.DC = 1) while HCR.VM = 0.

ArchMsg.Warning.warning_deprecated_wvr_bit_2_set

TAGS **UNPREDICTABLE** DISPLAY DBGWVR%{N:d}_EL1=%{WVR:h} has bit 2 set, deprecated in ARMv8. Fields:

N unsigned int

index.

WVR unsigned int

DBGWVR.

ArchMsg.Warning.warning_eret_while_software_step_active_pending

Missing ISB between setting MDSCR_EL1.SS and ERET TAGS UNPREDICTABLE.

ArchMsg.Warning.warning_exclusive_to_non_normal

TAGS UNPREDICTABLE DISPLAY Exclusive Access to Non-Normal Memory %{ADDR}. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_exclusive_to_non_writeback

TAGS UNPREDICTABLE DISPLAY Exclusive Access to Non-Writeback-Cacheable Memory %{ADDR}. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_execute_from_device_memory

TAGS UNPREDICTABLE DISPLAY An attempt was made to execute from Device memory.

ArchMsg.Warning.warning_illegal_cpsr_mode

TAGS UNPREDICTABLE DISPLAY Writing an illegal or unimplement mode (%{NEW_MODE:h}) to CPSR was unpredictable. Fields:

NEW_MODE unsigned int

New value of CPSR.M.

ArchMsg.Warning.warning_illegal_srs_mode

TAGS UNPREDICTABLE DISPLAY Illegal or UNPREDICTABLE mode (%{MODE:h}) used for SRS instruction. Fields:

MODE unsigned int

mode for Banked SP.

ArchMsg.Warning.warning_implementation_defined_read_debug_register_in_SCS

Read Access to Debug registers from the processor is IMPLEMENTATION DEFINED TAGS IMP_DEF. Fields:

PPB_OFFSET unsigned int

debug register offset.

ArchMsg.Warning.warning_implementation_defined_sequential_security_transitions_supported

The behavior of sequential instruction fetches that cross from non-secure to secure memory and contain an SG instruction is CONSTRAINED UNPREDICTABLE TAGS UNPREDICTABLE.

ArchMsg.Warning.warning_implementation_defined_stack_limit_check_supported

It is **IMPLEMENTATION DEFINED** whether stack pointer limit checking is performed for this instruction TAGS IMP_DEF. Fields:

ADDRESS unsigned int

Address of instruction.

OPCODE unsigned int

instruction opcode.

ArchMsg.Warning.warning_implementation_defined_taggednonsharedmemorytype

TAGS IMP_DEF DISPLAY Memory attribute has Tagged, non shared memory type; it is **IMPLEMENTATION DEFINED** whether the memory location is treated as Tagged or non-Tagged.

ArchMsg.Warning.warning_implementation_defined_write_debug_register_in_SCS

Write Access to Debug registers from the processor is **IMPLEMENTATION DEFINED** TAGS IMP_DEF. Fields:

DATA unsigned int

data attempted to be written.

PPB_OFFSET unsigned int

debug register offset.

ArchMsg.Warning.warning_invalid_tcr_granule

TAGS **UNPREDICTABLE** DISPLAY TCR.TG{%TG_ID:(0|1)} bits have been set to a granule size {%REQUEST:d}K not implemented - using {%SUBSTITUTE:d}K. Fields:

REQUEST signed int

page size requested (or 0 for reserved).

SUBSTITUTE signed int

best guess available page size.

TG_ID bool

bits TG0 or TG1.

ArchMsg.Warning.warning_load_multiple_user_registers_from_user_mode

TAGS **UNPREDICTABLE** DISPLAY An LDM(user registers) instruction executed from user mode.

ArchMsg.Warning.warning_mte_in_mem_attr_not_supported

TAGS **UNPREDICTABLE** DISPLAY Memory attribute {%ATTR} has Tagged memory type but Armv8.5-MTE is not implemented. Fields:

ATTR unsigned int

Memory attribute.

ArchMsg.Warning.warning_reserved_breakpoint_state_match

TAGS **UNPREDICTABLE** DISPLAY {%IS_BREAKPOINT:(Watchpoint|Breakpoint)} programmed with a reserved combination of HMC, SSC and {%IS_BREAKPOINT:(PAC|PMC)}. Fields:

IS_BREAKPOINT unsigned int

Is a breakpoint, not a watchpoint.

ArchMsg.Warning.warning_shareability

TAGS **UNPREDICTABLE** DISPLAY Unpredictable: combination of the 1st and 2nd stages of translation is Normal Inner Non-Cacheable, Outer Non-Cacheable.

ArchMsg.Warning.warning_software_step_set_while_enabled

TAGS **UNPREDICTABLE** DISPLAY MDSCR_EL1.SS set to 1 while software step debug exceptions are enabled.

ArchMsg.Warning.warning_thumb_instruction_wraps_around_memory

TAGS **UNPREDICTABLE** DISPLAY Thumb instruction wraps around memory.

ArchMsg.Warning.warning_ttbr_sbz_bits_are_not_zero

TAGS **UNPREDICTABLE** DISPLAY %{TTBR}[%{MSB:d}:0] should be zero, but are not. Fields:

MSB unsigned int

x-1.

TTBR enum

which TTBR register.

ArchMsg.Warning.warning_unaligned_address_dbgdtrrx_write

TAGS **UNPREDICTABLE** DISPLAY Unaligned address %{ADDR} in X0/R0 when DBGDTRRX written in memory access mode. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unaligned_address_dbgdtrtx_read

TAGS **UNPREDICTABLE** DISPLAY Unaligned address %{ADDR} in X0/R0 when DBGDTRTX read in memory access mode. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unaligned_to_device

TAGS **UNPREDICTABLE** DISPLAY Unaligned Access to Device Memory %{ADDR}. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unaligned_to_strongly_ordered

TAGS **UNPREDICTABLE** DISPLAY Unaligned Access to Strongly Ordered Memory %{ADDR}. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unknown_sau_rnr

TAGS UNPREDICTABLE DISPLAY SAU_RNR was set to an unsupported value.

ArchMsg.Warning.warning_unpredictable_AIRCR_PRIS_and_BFHFNMINs

TAGS UNPREDICTABLE DISPLAY The effect of setting both AIRCR.BFHFNMINs and AIRCR.PRIS to 1 is UNPREDICTABLE.

ArchMsg.Warning.warning_unpredictable_AIRCR_VECTCLRACTIVE_when_not_in_debug

TAGS UNPREDICTABLE DISPLAY The effect of writing a 1 to AIRCR.VECTCLRACTIVE if the processor is not halted in Debug state is UNPREDICTABLE.

ArchMsg.Warning.warning_unpredictable_AIRCR_VECTRESET_and_SYSRESETREQ

TAGS UNPREDICTABLE DISPLAY When the processor is halted in Debug state, if a write to the register writes a 1 to both VECTRESET and SYSRESETREQ, the behavior is UNPREDICTABLE.

ArchMsg.Warning.warning_unpredictable_AIRCR_VECTRESET_when_not_in_debug

TAGS UNPREDICTABLE DISPLAY The effect of writing a 1 to AIRCR.VECTRESET if the processor is not halted in Debug state is UNPREDICTABLE.

ArchMsg.Warning.warning_unpredictable_AIRCR_incorrect_VKEY

TAGS UNPREDICTABLE DISPLAY The value 0x05FA must be written to AIRCR.VKEY, otherwise the register write is UNPREDICTABLE.

ArchMsg.Warning.warning_unpredictable_EXC_RETURN_Reserved_Bit

EXC_RETURN[23:7] are reserved with the special condition that all bits should be written as one. Values other than all 1s are UNPREDICTABLE. TAGS UNPREDICTABLE. Fields:

EXC_RETURN unsigned int

Special PC value.

SBO_MASK unsigned int

Which bits Should Be One to not UNPRED.

ArchMsg.Warning.warning_unpredictable_PMEVTYPEPn_ELO_VS_reserved_value

TAGS UNPREDICTABLE DISPLAY Unpredictable reserved value of PMEVTYPEn<n>_ELO.VS.

ArchMsg.Warning.warning_unpredictable_change_to_DEMCR_MON_STEP_at_insufficient_priority

TAGS UNPREDICTABLE DISPLAY The effect of changing DEMCR.MON_STEP at an execution priority that is lower than the priority of the DebugMonitor exception is **CONSTRAINED** UNPREDICTABLE.

ArchMsg.Warning.warning_unpredictable_change_to_priority_of_active_exception

TAGS UNPREDICTABLE DISPLAY Changing the priority of an active exception is UNPREDICTABLE.

ArchMsg.Warning.warning_unpredictable_clear_lspact

TAGS UNPREDICTABLE DISPLAY UNPREDICTABLE state on context switch. Fields:

DRVEXC_TAKEN unsigned int

Derived exception taken, not pended.

REACHED_LAST_STORE unsigned int

Only the last faulted.

ArchMsg.Warning.warning_unpredictable_dbg_exit_unaligned_dlr

TAGS **UNPREDICTABLE** DISPLAY Debug exit to AArch32 with DLR[0] set to 1. PC[0] can be set to 0 or to DLR[0].

ArchMsg.Warning.warning_unpredictable_exception_catch

TAGS **UNPREDICTABLE** DISPLAY Generation of an exception catch event is unpredictable.

ArchMsg.Warning.warning_unpredictable_exception_return_inconsistent_state

TAGS **UNPREDICTABLE** DISPLAY state on exception return is unpredictable.

ArchMsg.Warning.warning_unpredictable_exception_return_instruction

TAGS **UNPREDICTABLE** DISPLAY exception return instruction is unpredictable.

ArchMsg.Warning.warning_unpredictable_in_debug_state

TAGS **UNPREDICTABLE** DISPLAY Instruction is **UNPREDICTABLE** when executed in debug state.

ArchMsg.Warning.warning_unpredictable_mair_encoding

TAGS **UNPREDICTABLE** DISPLAY Memory attribute %{ATTR} has invalid MAIR encoding; this may lead to unpredictable results. Fields:

ATTR unsigned int

Memory attribute.

ArchMsg.Warning.warning_unpredictable_pmu_counter_access

TAGS **UNPREDICTABLE** DISPLAY Access to PMU counters from non-secure EL0 or EL1 is **UNPREDICTABLE** with MDCR_EL2.HPMN set to 0.

ArchMsg.Warning.warning_unpredictable_prioritization_breakpoint_match_vector_catch

TAGS **UNPREDICTABLE** DISPLAY Two events with the same priority occurred at the same instruction: Address Matching Vector Catch debug event, and Breakpoint debug event. It is constrained unpredictable which is taken.

ArchMsg.Warning.warning_unpredictable_stack_selection

TAGS **UNPREDICTABLE** DISPLAY **UNPREDICTABLE** state on context switch.

ArchMsg.Warning.warning_unpredictable_tsize_out_of_range

TAGS **UNPREDICTABLE** DISPLAY %{TCR}.T%{TTBR:d}SZ (%{TSIZE:d} bits) is out of range. Must be between 25 and %{TSIZE_MAX:d}. Fields:

TCR enum

Which TCR.

TSIZE signed int

Value of TSize.

TSIZE_MAX signed int

The maximum allowed value.

TTBR unsigned int

Which TTBR region.

ArchMsg.Warning.warning_unpredictable_unaligned_pc_as_base_register

TAGS **UNPREDICTABLE** DISPLAY The PC value must be word-aligned, otherwise the behavior of the instruction is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_unaligned_pop_stack

TAGS **UNPREDICTABLE** DISPLAY **CONSTRAINED UNPREDICTABLE** unaligned stack frame `{FRAME_ADDR:x}` in exception-return. Fields:

FRAME_ADDR unsigned int

unaligned stack frame address.

ArchMsg.Warning.warning_unpredictable_unaligned_to_non_writeback

TAGS **UNPREDICTABLE** DISPLAY Unaligned Access to Non-Writeback-Cacheable Memory.

ArchMsg.Warning.warning_unpredictable_vmsa_memattrib

TAGS **UNPREDICTABLE** DISPLAY Unpredictable vmsa memory attribute with texcb value programmed to `{TEXCB}` when tex remap is `{TEX_REMAP}`. Fields:

TEXCB unsigned int

texcb value.

TEX_REMAP bool

use tex remap.

ArchMsg.Warning.warning_unpredictable_vtcr_t0sz_sl0

TAGS **UNPREDICTABLE** DISPLAY The combination of `VTCR.T0SZ={T0SZ:d}` and `VTCR.SL0={SL0:d}` is unpredictable. Fields:

SL0 signed int

`VTCR.SL0`, starting level for stage 2 translation table walks.

T0SZ signed int

`VTCR.T0SZ`, required input address range.

ArchMsg.Warning.warning_unpredictable_write_DHCSR

TAGS **UNPREDICTABLE** DISPLAY **UNPREDICTABLE** set of bit-changes in DHCSR. Fields:

NEW unsigned int

data attempted to be written.

OLD unsigned int

previous value.

ArchMsg.Warning.warning_unpredictable_write_SHCSR

TAGS **UNPREDICTABLE** DISPLAY **UNPREDICTABLE** setting a pending bit to 1 for an exception with priority \geq execution priority.

ArchMsg.Warning.warning_unsupported_access_to_memory_mapped_register

TAGS **UNPREDICTABLE** DISPLAY Access to memory mapped register is unsupported.

ArchMsg.Warning.warning_user_jmcr_access

TAGS UNPREDICTABLE DISPLAY User %{WRITE:(write|read)} access to JMCR is unpredictable.
Fields:

WRITE unsigned int
access is write.

ArchMsg.Warning.warning_wcr_mask_and_bas

TAGS UNPREDICTABLE DISPLAY DBGWCR_EL1.MASK is non-zero and BAS is not 0xff.

ArchMsg.Warning.warning_wcr_mask_reserved

TAGS UNPREDICTABLE DISPLAY DBGWCR_ELn.MASK is set to a reserved value.

ArchMsg.Warning.warning_wcr_non_configuous_bas

TAGS UNPREDICTABLE DISPLAY DBGWCR_EL1.BAS has non-contiguous set of ones.

ArchMsg.Warning.warning_wvr_masked_bit_not_zero

TAGS UNPREDICTABLE DISPLAY DBGWVRn_EL1.VA contains a bit masked by
DBGWCRn_EL1.MASK that is not zero.

ArchMsg.Why.why_branch_target_exception

A branch target exception has occurred. DISPLAY branch target exception: target pc=%{PC},
target opcode=%{OPCODE}. PRIMARY KEY PC. Fields:

BTYPE unsigned int
pstate.BTYPE at time branch taken.

OPCODE unsigned int
opcode at branch target.

PC unsigned int
Virtual Address of branch target.

ArchMsg.Why.why_illegal_state

An illegal mode change has occurred. DISPLAY illegal mode change: %{MESSAGE}. Fields:

MESSAGE string
The message.

BRANCH_MISPREDICT

Simulating branch mispredict. Fields:

PC unsigned int
Origin address (or 0 if unavailable).

BRA_DIR

Direct branches, to immediate address. Fields:

CORE_NUM unsigned int
Core number in a multi processor.

INST_COUNT unsigned int
The core's instruction counter, starting at 1 for the first instruction.

ISSET enum

The instruction set of the branch instruction.

IS_COND bool

Indicates if this is a conditional branch.

IS_HINTED bool

Whether the conditional instruction is hinted.

IS_ISB bool

Whether the instruction is ISB.

IS_LINK bool

Indicates if this is a branch with link.

PC unsigned int

The address of the branch instruction.

TARGET_ISSET enum

The instruction set after the branch.

TARGET_PC unsigned int

The address the instruction branches to.

BRA_INDIR

Indirect branches, perhaps to a register. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISSET enum

The instruction set of the branch instruction.

IS_COND bool

Indicates if this is a conditional branch.

IS_HINTED bool

Whether the conditional instruction is hinted.

IS_ISB bool

Whether the instruction is ISB.

IS_LINK bool

Indicates if this is a branch with link.

PC unsigned int

The address of the branch instruction.

TARGET_ISSET enum

The instruction set after the branch.

TARGET_PC unsigned int

The address the instruction branches to.

BRA_INDIR_CCFAIL

Indirect branches fail, perhaps to a register.

BRA_RET_CCFAIL

Branch return fail, perhaps to a register.

CACHE_MAINTENANCE_OP

Cache Maintenance Operation. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

DATA unsigned int

Specified MVA or Set/way.

FUNCTION enum

Clean / Invalidate.

SCOPE enum

Affected region.

SIDE enum

Inst / Data.

CCFAIL

Conditional instruction condition check fail. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CCFAIL_UNC

Unconditional instruction condition check fail. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CCPASS

Conditional instruction condition check pass. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CCPASS_UNC

Unconditional instruction condition check pass. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CODE_CACHE_MAINT

Code cache maintenance. Fields:

END_INCL unsigned int

Inclusive end address for VA or PA requests.

NS bool

Non-secure world for PA.

START unsigned int

Start address for VA or PA requests.

TYPE enum

Request type.

COMPILE_BLOCK_END

Last instruction of basic block translated. Fields:

VADDR unsigned int

Address of next instruction after this basic block.

COMPILE_BLOCK_START

First instruction of basic block translated. Fields:

VADDR unsigned int

Address of first instruction in basic block.

COMPILE_INST

ARM instruction compiled. Fields:

DISASS string

Disassembly of the instruction.

ISSET enum

The instruction set of this instruction.

ITSTATE unsigned int

The ITSTATE current for the instruction.

OPCODE unsigned int

The opcode of the instruction.

PC unsigned int

The address of the instruction.

SIZE unsigned int

The size of the instruction in bytes.

CONTEXTIDR

Context ID Register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

NS bool

Secure or nonsecure banked register is accessed.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The new value written.

CONTEXT_SYNC

Called for every context synchronization event. Fields:

ARCH_CSE bool

Architectural Context Synchronization Event.

PC unsigned int

The address of the synchronization instruction.

SYSREG_SYNC bool

System register synchronization event.

CORE_ENDIAN

Core BE8 Big-Endian state changed. Fields:

BE8 bool

Core BE8 Big-Endian state.

CORE_INFO

Static processor attributes. Only triggered by a call to DumpState(). Fields:

ARCH_PROFILE enum

The architecture profile of the core.

CLUSTER_ID unsigned int

The cluster ID of this processor.

CORE_NUM unsigned int

The number of this core in an MP processor.

DETAILED_CORE_FEATURES enum

Detailed core features as a bitfield. Enum values are bitmasks to extract the info from the field.

FPU_VERSION enum

The VFP version implemented by the core.

MEM_ARCH enum

The memory architecture of the core.

MIDR_EL1 unsigned int

MIDR_EL1 value.

NUM_CORES unsigned int

The number of cores in this MP processor.

QUANTUM_SIZE unsigned int

The default quantum size of the core (in ticks).

SECURITY_FEATURES bool

Does the core have security features?.

CORE_LOADS

Processor load accesses. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ACQREL enum

Is this an acquire/release.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DATA unsigned int

The data read or written.

ELEMENT_SIZE unsigned int

Width of each element.

ENDIANNESS enum

Byte ordering of this access.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

LOCK enum

Normal, exclusive or locked access.

MECID unsigned int

Memory Encryption Context ID, if applicable.

MECID2 unsigned int

Memory Encryption Context ID of the second page, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PAGE_SIZE unsigned int

Page size as $\log_2(\text{size in bytes})$.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK, 2=Exclusive Failed, 3=Unsupported.

SIZE unsigned int

Width of the access in bytes. Only required if DATA is not traced.

TRANS bool

Is this a translated access.

VADDR unsigned int

The virtual address of the access.

CORE_REGS

Changes of the core registers R0 to R14. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number, 0 to 14.

MODE enum

Bank of the register accessed.

OLD_VALUE unsigned int

The old value overwritten.

PHYS_ID enum

The physical register accessed.

VALUE unsigned int

The new value written to the register.

CORE_REGS64

Changes of the core registers X0..X30, SP_ELn. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ID enum

The register number, 0-30 for X0-X30, >=32 for SP_ELn.

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

CORE_REGS64_READ

General purpose AArch64 register has been read X0..X30, SP_ELn. Fields:

ID enum

The register number, 0-30 for X0-X30, >=32 for SP_ELn.

VALUE unsigned int

The value read from the register.

CORE_STALLED_TRANSITION

Detect when a core is prevented from progressing PChannel states. Fields:

CURRENT_STATE enum

Current power mode of core.

REASON string

The reason a transition was stalled, and what must be done to unblock it.

TARGET_STATE enum

Desired power mode of core after transition.

CORE_STORES

Processor store accesses. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ACQREL enum

Is this an acquire/release.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DATA unsigned int

The data read or written.

ELEMENT_SIZE unsigned int

Width of each element.

ENDIANNESS enum

Byte ordering of this access.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

LOCK enum

Normal, exclusive or locked access.

MECID unsigned int

Memory Encryption Context ID, if applicable.

MECID2 unsigned int

Memory Encryption Context ID of the second page, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PAGE_SIZE unsigned int

Page size as $\log_2(\text{size in bytes})$.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK, 2=Exclusive Failed, 3=Unsupported.

SIZE unsigned int

Width of the access in bytes. Only required if DATA is not traced.

TRANS bool

Is this a translated access.

VADDR unsigned int

The virtual address of the access.

CP14_READ

System Coprocessor register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP14 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc1 unsigned int

opcode 1.

opc2 unsigned int

opcode 2.

CP14_WRITE

System Coprocessor register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP14 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc1 unsigned int

opcode 1.

opc2 unsigned int
opcode 2.

CP15_READ

System Coprocessor register read. Fields:

CORE_NUM unsigned int
Core number in a multi processor.

CRm unsigned int
CRm.

CRn unsigned int
CRn.

NS bool
Secure or nonsecure banked register is accessed.

REG_NAME string
Name of the CP15 register accessed.

SECURITY_STATE enum
The current security state.

UNDEF bool
The register accessed is undefined.

VALUE unsigned int
The value read.

opc1 unsigned int
opcode 1.

opc2 unsigned int
opcode 2.

CP15_READ64

System Coprocessor register read. Fields:

CORE_NUM unsigned int
Core number in a multi processor.

CRm unsigned int
CRm.

NS bool
Secure or nonsecure banked register is accessed.

REG_NAME string
Name of the CP15 register accessed.

SECURITY_STATE enum
The current security state.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc unsigned int

opcode 1.

CP15_WRITE

System Control Coprocessor register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP15 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc1 unsigned int

opcode 1.

opc2 unsigned int

opcode 2.

CP15_WRITE64

System Coprocessor register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP15 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc unsigned int

opcode 1.

CPSR

CPSR change. (In AArch64 PSTATE represented in SPSR format). Fields:

CORE_NUM unsigned int

Core number in a multi processor.

OLD_VALUE unsigned int

The old CPSR value.

UNKNOWN unsigned int

Bits within the register that have unknown value.

VALUE unsigned int

The new CPSR value.

CRYPTO_SPEC

Every crypto instruction speculatively executed.

CorePowerStateChange

Core power-up state has changed. DISPLAY Core powered %{PU:(up|down)}. Fields:

PU bool

Whether the core powered up.

DATA_CACHE_SET_ALLOC_TAGS

Set allocation tags from DC GVA instruction. Fields:

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate

when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

NSDESC unsigned int

The security state of the access.

PADDR unsigned int

The physical address of the access.

PAS enum

The physical address space of the page.

RESPONSE enum

0=Aborted, 1=OK.

SIZE unsigned int

The size of memory in bytes.

TAG unsigned int

The value of the tag being set.

TAGGING_TYPE enum

Type of Memory tagging (Physical or Virtual).

TAG_VADDR unsigned int

The virtual address of the Virtual Tag if Virtual tagging is used; otherwise 0.

VADDR unsigned int

Virtual address used for setting allocation tags.

DATA_CACHE_ZERO

Zero and invalidate cache line from DC ZVA instruction. Fields:

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

NSDESC unsigned int

The security state of the access.

PADDR unsigned int

The physical address of the access.

PADDR_TAG unsigned int

If tag checked, the physical address tag for the access.

PAS enum

The physical address space of the page.

RESPONSE enum

0=Aborted, 1=OK.

SIZE unsigned int

The size of zero'd memory in bytes.

TAG_CHECKED bool

Is this access tag checked.

VADDR unsigned int

Virtual address of the zero'd cache line.

DEBUG_EVENT

Hardware debug support event. Fields:

EVENT enum

Description of event.

VALUE unsigned int

data value.

DEBUG_TARGET_ISET

Hardware debug target ISA. Fields:

TARGET_ISET enum

The target instruction set of the debug state.

DMI_ALLOCATE

DMI allocated. Fields:

ACCESS_DIRECTION enum

READ / WRITE.

PADDR unsigned int

The physical address of the DMI.

PTR unsigned int

DMI range pointer.

SIDE enum

DSIDE / ISIDE.

SIZE unsigned int

The size of DMI.

VADDR unsigned int

The virtual address of the DMI..

DMI_HIT

DMI hit, access used DMI (only fire for small DMI regions i.e. less than a full page). Fields:

ACCESS_DIRECTION enum

READ / WRITE.

PADDR unsigned int

The physical address of the DMI.

SIDE enum

DSIDE / ISIDE.

SIZE unsigned int

The size of the DMI.

VADDR unsigned int

The virtual address of the DMI..

DMI_REVOKE

DMI revoked. Fields:

ACCESS_DIRECTION enum

READ / WRITE.

PADDR unsigned int

The physical address of the DMI.

SIDE enum

DSIDE / ISIDE.

SIZE unsigned int

The size of DMI.

VADDR unsigned int

The virtual address of the DMI..

END_COMPILE

Compilation end. Fields:

END_OF_PAGE_COUNT unsigned int

Number of basic blocks exited due to page boundary since START_COMPILE.

FETCHFAIL_COUNT unsigned int

Number of basic blocks exited due to ifetch failure START_COMPILE.

INST_COUNT unsigned int

Number of instructions compiled since START_COMPILE.

NONSEQ_COUNT unsigned int

Number of basic blocks exited due to non-sequential instructions since START_COMPILE.

PAGE_STRADDLE_COUNT unsigned int

Number of basic blocks exited due to unaligned instructions crossing page since START_COMPILE.

ESB

Error Synchronization Barrier hit. Fields:

ASYNC bool

ESB is async.

IMPLICIT bool

ESB is implicit.

EXCEPTION

Exception taken. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ESR unsigned int

The value assigned to ESR when exception is taken.

ESR_EC unsigned int

The Value assigned to ESR.EC when exception is taken.

IS_PHYSICAL bool

Physical or Virtual exception.

IS_TRAP bool

Exception coming from a trap.

LR unsigned int

The value assigned to the link register.

PC unsigned int

The location where the exception occurred.

PREFERRED_RETURN unsigned int

The preferred return address for the exception.

TAKEN_LOCALLY bool

Whether exception is taken locally(same exception level).

TARGET_ISET enum

The instruction set of the exception handler code.

TARGET_PC unsigned int

The address the exception branches to.

VECTOR enum

The exception vector.

VECTOR_OFFSET unsigned int

The offset of an exception vector from the base address.

EXCEPTION_END

Every exception completed.

EXCEPTION_INFO

Trace human-readable reason of the current exception for data/instruction aborts, floating point exceptions and unknown exceptions. Fields:

DESCRIPTION string

Reason for current exception.

ESR_EC unsigned int

The value of ESR.EC when exception is taken.

EXCEPTION_RAISE

Every exception raised.

EXCEPTION_RETURN

Branches on leaving exception. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISET enum

The instruction set of the branch instruction.

IS_COND bool

Indicates if this is a conditional branch.

IS_HINTED bool

Whether the conditional instruction is hinted.

IS_ISB bool

Whether the instruction is ISB.

IS_LINK bool

Indicates if this is a branch with link.

PC unsigned int

The address of the branch instruction.

TARGET_ISET enum

The instruction set after the branch.

TARGET_PC unsigned int

The address the instruction branches to.

EXCEPTION_RETURN_PREBRANCH

Exception return event that must occur before PC and processor state have updated.

EXCEPTION_START

Every exception started.

FIQ_TAKEN

FIQ taken exception.

FP_EXCEPTIONS_SIGNALLED

Trace which FP exceptions were signalled as a part of the FP operations performed by an instruction. Fields:

FP_EXCEPTIONS_SIGNALLED_MASK unsigned int

Indicates which FP exceptions were signalled by an instruction. This uses the same encoding as the cumulative floating-point exception bitfields in FPSR.

FP_STATE

Floating point state. Fields:

FPSR unsigned int

FPSR register value.

MASK unsigned int

The mask indicating updated cumulative bits.

FREQ_CHANGED

Core frequency changed. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The instruction count of this core.

NEW_FREQ signed int

The new frequency of this core (expressed in Hz).

OLD_FREQ signed int

The old frequency of this core (expressed in Hz).

HLT

HLT instruction occurred. Fields:

IMM16 unsigned int

The 16-bit immediate value encoded in HLT instruction.

INST

Every instruction executed. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DEBUG_STATE bool

The instruction is executed in debug state.

DISASS string

Disassembly of instruction.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISSET enum

The current instruction set.

ITSTATE unsigned int

The current ITSTATE.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

MODE enum

The mode the core is in.

NS unsigned int

The core's non-secure bit.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

OPCODE unsigned int

The opcode of the instruction.

PADDR unsigned int

The physical address of the instruction.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the instruction.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the page.

PC unsigned int

The address of the instruction.

SECURITY_STATE enum

The core's security state.

SIZE unsigned int

The size of the instruction in bytes.

INST_END

Every instruction completed.

INST_START

Every instruction started. Fields:

DEBUG_STATE bool

The instruction is executed in debug state.

ISSET enum

The current instruction set.

MODE enum

The mode the core is in.

NS enum

The current Secure State.

PC unsigned int

The address of the conditional instruction.

SECURITY_STATE enum

The current Security State.

INST_STRADDLE

Instruction straddles boundary. Fields:

DEBUG_STATE bool

The instruction is executed in debug state.

ISSET enum

The current instruction set.

MODE enum

The mode the core is in.

NS enum

The current Secure State.

PC unsigned int

The address of the conditional instruction.

SECURITY_STATE enum

The current Security State.

IRQ_TAKEN

IRQ taken exception.

ISB

ISB instruction occurred. Fields:

PC unsigned int

The address of the ISB instruction.

LOAD_MULTIPLE_REGS_LIST

List of destination registers (not source) of a multiple registers load instruction. Fields:

REG string

Register to add into the list.

LOCAL_MONITOR

Local monitor activity. Fields:

PADDR unsigned int

Local Monitor Address.

State enum

State of the monitor (Open/Exclusive).

MAPPING_CREATE

A memory mapping has been created. This is an implementation detail of the model and so may change without notice. Fields:

ASID unsigned int

The ASID used during translation, or 0 if no ASID was used.

IS_VALID bool

Whether the mapping is valid or not.

PADDR unsigned int

Base physical address of region, or 0 if not known.

SIDE enum

I-side or D-side.

SIZE unsigned int

Size of the page, or 0 if not known.

VADDR unsigned int

Base virtual address of region.

VMID unsigned int

The VMID used during translation, or 0 if no VMID was used.

MAPPING_DESTROY

A memory mapping has been destroyed. This is an implementation detail of the model and so may change without notice. Fields:

ASID unsigned int

The ASID used during translation.

CAUSE enum

The cause of invalidation.

PADDR unsigned int

Base physical address of page.

SIDE enum

I-side or D-side.

SIZE unsigned int

Size of the page.

VADDR unsigned int

Base virtual address of page.

VMID unsigned int

The VMID used during translation.

MEMMAP_DEBUG_READ

Memory mapped reads to the debug registers. Fields:

ADDR unsigned int

address.

CORE_NUM unsigned int

Core number in a multi processor.

EXT bool

Whether access is from an external device (such as the DAP).

NS enum

Secure state of the access.

REG_NAME string

Name of the debug register accessed.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

MEMMAP_DEBUG_WRITE

Memory mapped writes to the debug registers. Fields:

ADDR unsigned int

address.

CORE_NUM unsigned int

Core number in a multi processor.

EXT bool

Whether access is from an external device (such as the DAP).

NS enum

Secure state of the access.

REG_NAME string

Name of the debug register accessed.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The new value written.

MMU_D128_TTB_READ

This event is triggered by reads caused by a D128 translation table walk. Fields:

ABORTED bool

Was the walk successful.

DATA_HIGH unsigned int

The data upper bits read.

DATA_LOW unsigned int

The data lower bits read.

IPA unsigned int

For stage 1, the IPA of the read.

LEVEL unsigned int

Translation table level.

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

OUTADDR unsigned int

The output physical address.

PADDR unsigned int

The physical address of the read.

PASmem enum

Physical Address Space of the accessed memory.

PASreq enum

Physical Address Space of the requested lookup.

REGIME_EL enum

Entry matches in this translation regime.

REQUESTER enum

The requester of this table walk.

SIDE enum

Inst / Data.

STAGE unsigned int

Translation stage.

MMU_D128_TTB_WRITE

This event is triggered by writes caused by a D128 translation table walk. Fields:

ABORTED bool

Was the walk successful.

DATA_HIGH unsigned int

The data upper bits written.

DATA_LOW unsigned int

The data lower bits written.

IPA unsigned int

For stage 1, the IPA of the write.

IS_HACDBS_ACCESS bool

If the access was due to HACDBS_CLEAN.

LEVEL unsigned int

Translation table level.

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

OUTADDR unsigned int

The output physical address.

PADDR unsigned int

The physical address of the write.

PASmem enum

Physical Address Space of the accessed memory.

PASreq enum

Physical Address Space of the requested lookup.

REGIME_EL enum

Entry matches in this translation regime.

SIDE enum

Inst / Data.

STAGE unsigned int

Translation stage.

MMU_TRANS

Address translation information. Fields:

ASID unsigned int

Address space identifier.

CORE_NUM unsigned int

Core number in a multi processor.

Hyp bool

Entry matches in Hyp state only.

INNERCACHE_RA bool

Is the inner cache allocate on read.

INNERCACHE_TRANSIENT bool

Is the inner write-through transient.

INNERCACHE_TYPE enum

Inner Caching scheme (NC/MB/WA).

INNERCACHE_WA bool

Is the inner cache allocate on write.

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MEMTYPE enum

Memory type.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC enum

Is secure side supposed to access secure or nonsecure memory.

OUTERCACHE_RA bool

Is the outer cache allocate on read.

OUTERCACHE_TRANSIENT bool

Is the outer write-through transient.

OUTERCACHE_TYPE enum

Outer Caching scheme (NC/MB/WA).

OUTERCACHE_WA bool

Is the outer cache allocate on write.

PADDR unsigned int

Physical address of the access.

PAGESIZE unsigned int

Page size as log2(size).

PAS enum

PAS supposed to be accessed is secure or nonsecure memory or Root/Realm if RME is implemented.

SH enum

Shareability.

SIDE enum

Inst / Data.

STAGE1_PERM unsigned int

Stage 1 permissions, [5:3] Privileged access (XWR) [2:0] User access (XWR).

STAGE2_PERM unsigned int

Stage 2 permission mask [5:3] Privileged access (XWR) [2:0] User access (XWR).

VADDR unsigned int

Virtual address of the access.

VMID unsigned int

Virtual machine identifier.

XS bool

XS Attribute.

nG enum

Flag indicating whether ASID will be matched.

MMU_TTB_READ

This event is triggered by reads caused by a translation table walk. Fields:

ABORTED bool

Was the walk successful.

DATA unsigned int

The data read.

DESC_KIND enum

The encoding scheme of the descriptors used for this walk.

IPA unsigned int

For stage 1, the IPA of the read.

LEVEL unsigned int

Translation table level.

LPAAE bool

Is this for an LPAAE translation (*DEPRECATED*, use DESC_KIND).

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSmem bool

Non secure memory access.

NSreq bool

Non secure request.

OUTADDR unsigned int

The output physical address.

PADDR unsigned int

The physical address of the read.

PASmem enum

Physical Address Space of the accessed memory.

PASreq enum

Physical Address Space of the requested lookup.

REGIME_EL enum

Entry matches in this translation regime.

REQUESTER enum

The requester of this table walk.

SIDE enum

Inst / Data.

STAGE unsigned int

Translation stage.

MMU_TTB_WRITE

This event is triggered by writes caused by a translation table walk. Fields:

ABORTED bool

Was the walk successful.

DATA unsigned int

The data written.

DESC_KIND enum

The encoding scheme of the descriptors used for this walk.

IPA unsigned int

For stage 1, the IPA of the write.

IS_HACDBS_ACCESS bool

If the access was due to HACDBS_CLEAN.

LEVEL unsigned int

Translation table level.

LPAE bool

Is this for an LPAE translation (*DEPRECATED*, use DESC_KIND).

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSmem bool

Non secure memory access.

NSreq bool

Non secure request.

OUTADDR unsigned int

The output physical address.

PADDR unsigned int

The physical address of the write.

PASmem enum

Physical Address Space of the accessed memory.

PASreq enum

Physical Address Space of the requested lookup.

REGIME_EL enum

Entry matches in this translation regime.

SIDE enum

Inst / Data.

STAGE unsigned int

Translation stage.

MODE_CHANGE

Mode change. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

MODE enum

The new mode.

NON_SECURE enum

The core's new non-secure bit.

OLD_MODE enum

The old mode.

SECURITY_STATE enum

The new security state.

NOT_TAG_CHECKED_INST

MTE Tag not checked for current instruction.

PERIODIC

Called for every quantum. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The instruction count of this CPU.

PC unsigned int

The address of the next instruction to be executed on this CPU.

PMU_COUNTER_OVERFLOW

PMU counter overflow. Fields:

COUNTER_GROUP enum

Counter group.

COUNTER_RANGE enum

Counter range.

EVENT_ID unsigned int

Event Identifier, field is valid only for Event counter group.

INDEX unsigned int

Counter index (as selected by PMSELR), field is valid only for Event/Cycle counter groups.

INTERRUPT bool

Is interrupt enabled on overflow for this counter.

POTINDEX_UPDATE

Value of POT index for current pc changed. Fields:

NEW_POTINDEX unsigned int

New POTIndex.

OLD_POTINDEX unsigned int

Old POTIndex.

PREFETCH_MEMORY64

Prefetch from PRFM or PRFUM instructions. Fields:

PRFOP enum

Prefetch hint.

VADDR unsigned int

Virtual address of the location that should be prefetched.

PRELOAD64_REQUEST

Preload requests from PRFM or PRFUM instructions. Fields:

PRFOP enum

Prefetch hint.

VADDR unsigned int

Virtual address of the location that should be prefetched.

PRELOAD_DATA

Data preload from PLD instruction. Fields:

VADDR unsigned int

Virtual address of the data that should be preloaded.

PRELOAD_INST

Instruction preload from PLI instruction. Fields:

VADDR unsigned int

Virtual address of the instruction that should be preloaded.

PRE_CORE_LOAD

Trace just before a core load. Fields:

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

PRE_CORE_STORE

Trace just before a core store. Fields:

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

PRE_TTB_READ

This event is triggered before reads caused by a translation table walk. Fields:

ADDR unsigned int

The physical address of the read.

PSTATE

PSTATE fields update. Fields:

ALLINT unsigned int

All interrupt mask bit.

BTYPE unsigned int

Branch target identification bit.

CORE_NUM unsigned int

Core number in a multi processor.

DAIF unsigned int

Exception mask bits.

DIT unsigned int

Data Independent Timing bit.

EL unsigned int

Current Exception level field.

EXLOCK unsigned int

Exception return state lock bit.

IL unsigned int

Illegal Execution state bit.

NZCV unsigned int

Condition flags.

PACM unsigned int

PACM bit.

PAN unsigned int

Privileged Access Never state bit.

PM unsigned int

PMU exception mask bit.

PPEND unsigned int

PMU exception pending bit.

SM unsigned int

StreamingMode status bit.

SP unsigned int

Stack pointer register selection bit.

SS unsigned int

Software Step bit.

SSBS unsigned int

Speculative Store Bypass Safe bit.

TCO unsigned int

Tag Check Override bit.

TOUCHED unsigned int

Mask of the fields that have been touched since this trace has last been emitted. Note that touched does not necessarily mean the value changed. The index of the field is the bit index in the mask plus the index of this field plus one.

UAO unsigned int

User Access Override bit.

UINJ unsigned int

Undefined Instruction exceptions bit.

ZA unsigned int

ZA storage and ZT0 register enable bit.

nRW unsigned int

Current Execution state bit.

PSTATE_PACTIVE_OPMODE_TRANSITION

Trace pactive and Opmode mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Operating mode after the transition.

OLD_STATE enum

Operating mode before the transition.

PSTATE_PACTIVE_OPMODE_TRANSITION_DENIED

Trace the denied pactive and Opmode mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Operating mode after the transition.

OLD_STATE enum

Operating mode before the transition.

REASON string

Reason for the transition being denied.

PSTATE_PACTIVE_POWER_TRANSITION

Trace pactive and power mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Power mode after the transition.

OLD_STATE enum

Power mode before the transition.

PSTATE_PACTIVE_POWER_TRANSITION_DENIED

Trace the denied pactive and power mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Power mode after the transition.

OLD_STATE enum

Power mode before the transition.

REASON string

Reason for the transition being denied.

PSTATE_PACTIVE_THREAD_OPMODE_TRANSITION

Trace pactive and Opmode mode transitions for threads on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Operating mode after the transition.

OLD_STATE enum

Operating mode before the transition.

QUANTUM_START

Called at the start of every quantum.

RANGE_PRELOAD_REQUEST

Preload requests from RPRFM. Fields:

COUNT unsigned int

Number of blocks of data to access.

LENGTH unsigned int

Number of contiguous bytes to be accessed in each block.

OP unsigned int

Range prefetch operation .

REUSE unsigned int

Maximum number of bytes to be accessed before next RPRFM .

STRIDE signed int

Stride to advance after accessing LENGTH bytes.

VADDR unsigned int

Base virtual address of range prefetch request.

RUN_STATE

Run state transition. Fields:

INST_COUNT unsigned int

Ticks count at point of transition.

NEW enum

New run state.

OLD enum

Old run state.

SEMIHOSTING_CALL

Call of a semihost function occurred. Fields:

PC unsigned int

The program counter after the semihosting call.

SEMIHOSTING_PRECALL

About to call semihost. Fields:

EL2 bool

Use EL2 translation regime.

NS bool

Is Non-Secure.

REG_WIDTH unsigned int

The current register width in bytes.

SIGNAL

External signal state change. Fields:

SIGNAL enum

Signal that changed.

STATE bool

Signal asserted state.

SOFTWARE_STEP

Return the Debug Software Step state. Fields:

STATE enum

Software Step state.

SPSR

SPSR change. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

MODE enum

Which of the banked SPSR registers is written.

OLD_VALUE unsigned int

The old SPSR value.

VALUE unsigned int

The new SPSR value.

START_COMPILE

Compilation started. Fields:

VADDR unsigned int

Instruction where compilation begins.

STG_MULTIPLE_MICRO_OPS_ADDRESS_RANGE

Address range for MTE stores with multiple uops. Fields:

ADDR_END unsigned int

Final address.

ADDR_START unsigned int

Starting address.

STORE_MULTIPLE_REGS_ADDRESS_RANGE

Address range for stores with multiple registers. Fields:

ADDR_END unsigned int

Final address.

ADDR_START unsigned int

Starting address.

SUBOPTIMAL_LDST_RETIRED

suboptimal load/store. Fields:

ACCESS_SIZE unsigned int

Log2 of the access width in bytes used for alignment checking (1: half-word, 2: word ...).

OPTIMAL_ALIGN_SELECTOR unsigned int

Selector to decide the optimal alignment size (1: access width, 2: 4-bytes, 3: 8-bytes, ..., 12: 4-kbytes).

VADDR unsigned int

The virtual address of the access.

SYNC

Called for every synchronization, that is every quantum break. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The instruction count of this CPU.

LOCAL_QUANTUM unsigned int

The local quantum of this CPU.

LOCAL_TIME unsigned int

The local time of this CPU.

SYNC_REASON enum

Reason for sync.

SYSCALL

System call instruction executed. Fields:

IMM unsigned int

Immediate value of the system call instruction.

TYPE enum

System call type.

VADDR unsigned int

Instruction that caused the system call.

SYSREG_READ128

System Coprocessor 128 bit register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

REGNUM enum

Internal register number.

REGNUM_PHYS enum

Internal register number physically accessed (e.g. in presense of VHE redirection).

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_READ64

System Coprocessor register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REGNUM enum

Internal register number.

REGNUM_OPERAND enum

Register number corresponding to the instruction operand.

REGNUM_PHYS enum

Internal register number physically accessed (e.g. in presense of VHE redirection).

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_UPDATE128

Triggers when the system updates a register. Fields:

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG enum

Register number.

UNKNOWN unsigned int

Bits of the register which became unknown.

VALUE unsigned int

Value written to the register.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_UPDATE32

Triggers when the system updates a register. Fields:

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG enum

Register number.

UNKNOWN unsigned int

Bits of the register which became unknown.

VALUE unsigned int

Value written to the register.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_UPDATE64

Triggers when the system updates a register. Fields:

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG enum

Register number.

UNKNOWN unsigned int

Bits of the register which became unknown.

VALUE unsigned int

Value written to the register.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_WRITE128

System Coprocessor 128 bit register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

REGNUM enum

Internal register number.

REGNUM_PHYS enum

Internal register number physically accessed (e.g. in presense of VHE redirection).

UNDEF bool

The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_WRITE64

System Coprocessor register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REGNUM enum

Internal register number.

REGNUM_OPERAND enum

Register number corresponding to the instruction operand.

REGNUM_PHYS enum

Internal register number physically accessed (e.g. in presense of VHE redirection).

UNDEF bool

The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

UNALIGNED_LDST_RETIRED

Processor unaligned load/store. Fields:

ACCESS_SIZE unsigned int

Log2 of the access width in bytes used for alignment checking (1=>half-word, 2->word ...).

NUMBER_OF_BEATS unsigned int

Number of accesses (beats) in this burst (Total size in bytes is ACCESS_SIZE * NUMBER_OF_BEATS).

VADDR unsigned int

The virtual address of the access.

VFP_D_REGS

VFP/NEON D 64 bit register write. Fields:

ALIASING enum

Type of register aliasing used.

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a write to the corresponding bit in the register. For example, 0x1 would indicate a write to VALUE[0].

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

VFP_D_REGS_READ

VFP/NEON D 64 bit register read. Fields:

ID unsigned int

The register number.

VALUE unsigned int

The current value of the register.

VFP_Q_REGS

VFP/NEON Q 128 bit register write. Fields:

ALIASING enum

Type of register aliasing used.

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a write to the corresponding byte in the register. For example, 0x1 would indicate a write to VALUE[7:0].

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

VFP_Q_REGS_READ

VFP/NEON Q 128 bit register read. Fields:

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a read to the corresponding byte in the register. For example, 0x1 would indicate a read to VALUE[7:0].

VALUE unsigned int

The current value of the register.

VFP_SYS_REGS

Writes to the VFP/NEON units system registers. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ID enum

Which VFP system register is written.

MASK unsigned int

The mask indicating updated cumulative bits.

OLD_VALUE unsigned int

The register's old value overwritten.

VALUE unsigned int

The new value written to the VFP system register.

VFP_S_REGS

VFP/NEON S 32 bit register write. Fields:

ALIASING enum

Type of register aliasing used.

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a write to the corresponding bit in the register. For example, 0x1 would indicate a write to VALUE[0].

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

VFP_S_REGS_READ

VFP/NEON S 32 bit register read. Fields:

ID unsigned int

The register number.

VALUE unsigned int

The current value of the register.

VIRTUAL_TAG_ALLOCATED

Virtual tag information when allocated. Fields:

TAG_VADDR unsigned int

Virtual tag virtual address.

VADDR unsigned int

Data Virtual address.

VIRTUAL_TAG unsigned int

Virtual tag value.

WAYPOINT

Signals end of basic block execution. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ISET enum

Origin instruction set.

IS_COND bool

Indicates if this is a conditional waypoint.

PC unsigned int

Origin address (or 0 if unavailable).

TAKEN bool

Indicates if this waypoint was taken.

TARGET unsigned int

Destination address.

TARGET_ISET enum

Destination instruction set.

WFE_END

WFE ended. Fields:

INST_COUNT unsigned int

Ticks count when leaving WFE.

WFE_EVENT_REGISTER

WFE event register status: set/clear, reason. Fields:

INST_COUNT unsigned int

Ticks count.

REASON enum

Reason for set/clear. Only REASON==0 (Cleared by WFE) clears the bit, all other reasons set it.

WFE_IGNORED

WFE ignored. Fields:

EVENT bool

This WFE was ignored because the event register was set.

INST_COUNT unsigned int

Ticks count when ignoring WFE.

TRAPPED bool

This WFE was trapped.

WFE_START

WFE entered. Fields:

INST_COUNT unsigned int

Ticks count when entering WFE.

WFI_END

WFI ended. Fields:

INST_COUNT unsigned int

Ticks count when leaving WFI.

WFI_IGNORED

WFI ignored. Fields:

DISABLED bool

This WFI was ignored because WFI is disabled.

INST_COUNT unsigned int

Ticks count when ignoring WFI.

REASON enum

specifies reason why WFI trace was ignored.

TRAPPED bool

This WFI was trapped.

WFI_START

WFI entered. Fields:

INST_COUNT unsigned int

Ticks count when entering WFI.

WFI_WAKEUP

WFI wakeup. Fields:

INST_COUNT unsigned int

Ticks count when WFI wakeup occurred.

REASON enum

Reason for wakeup.

2.6 ARM_AEMv8-R_MP

This section describes the trace sources.

AA64_ASE_SVE_REGS

Modification of AArch64 Advanced SIMD or SVE Vector Registers (via SIMD or Scalar operations), SVE Predicate Registers and ZT0. Fields:

ID enum

The register identifier.

MASK unsigned int

Bitmask of the register to signal the modified bits in the VALUE field.

SM bool

Whether the PE is in Streaming Mode.

VALUE unsigned int

Value of AArch64 vector register.

ASYNC_MEMORY_FAULT

Context ID Register write. Fields:

FAULT unsigned int

Fault status present in ESR.ISS encoding for Serror in ARCH64 and present in DFSR encoding for external abort format in ARCH32.

PADDR unsigned int

Physical Address (or 0 if unavailable).

VADDR signed int

Virtual Address (or 0 if unavailable).

ATOMIC_END_ACCESS

Bus trace access for atomics. Fields:

ACCESS_FAIL bool

Memory access failed.

ADDR unsigned int

The virtual address of the access.

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

COMPARE_VALUE unsigned int

Compare value for CAS.

LOAD_VALUE unsigned int

Loaded values.

NSDESC unsigned int

The security state of the access.

OPERAND_VALUE unsigned int

Operation's operand.

OPERATION enum

Operation type.

PADDR unsigned int

The physical address of the access.

PRIV bool

Is this a privileged access?.

ATOMIC_START_ACCESS

Bus trace access for atomics. Fields:

ADDR unsigned int

The virtual address of the access.

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

COMPARE_VALUE unsigned int

Compare value for CAS.

NSDESC unsigned int

The security state of the access.

OPERAND_VALUE unsigned int

Operation's operand.

OPERATION enum

Operation type.

PADDR unsigned int

The physical address of the access.

PRIV bool

Is this a privileged access?.

ArchMsg.Error.exit_code

DISPLAY exit code: %{EXITCODE} component kind: %{KIND} reason: %{REASON}. Fields:

EXITCODE signed int

Exit code to be returned.

KIND string

Component kind that invoked the exit code trace.

REASON string

Reason why the exit code trace was invoked (optional).

ArchMsg.Info.invalid_region_programmed

DISPLAY MPU Stage %{STAGE:d}: Region ID %{REGION_ID:d} has base address greater than end address Base Addr=%{BASE_ADDR:x}, End Addr=%{END_ADDR:x}. Fields:

BASE_ADDR unsigned int

Base Address.

END_ADDR unsigned int

End Address.

REGION_ID unsigned int

Region ID.

STAGE unsigned int

MPU Stage.

ArchMsg.Warning.IgnoredWarmResetRequest

Event fires when software attempts to request warm reset through write to deprecated DBGPRCR.CWRR.

ArchMsg.Warning.ReportedRevisionRegisterRead

DISPLAY Reported revision and patch do not match that of the cpu. Fields:

ORIGINAL_VALUE unsigned int

The value the register would have based on the modelled revision.

REGISTER string

The name of the register that has been read.

REPORTED_VALUE unsigned int

The value the register has been overridden to.

ArchMsg.Warning.Unpredictably Indexed PM Event Register

TAGS **UNPREDICTABLE** DISPLAY Event %{IsDirect:(Selector value|Register)} %{SEL} is >= %{N}.
Fields:

IsDirect unsigned int

Direct Access rather than indirect.

N unsigned int

Number of Accessible Registers.

SEL unsigned int

Selector.

ArchMsg.Warning.cache_contents_unknown

DISPLAY %{SIDE:(D|I)}-Cache was enabled but was not invalidated since power-on: cache lines could contain **UNKNOWN** content. Fields:

REGIME unsigned int

Current translation regime.

SIDE unsigned int

0 d-side cache or unified cache, 1 i-side.

ArchMsg.Warning.dcimvac_matches_watchpoint

TAGS **UNPREDICTABLE** DISPLAY Watchpoints matching an AArch32 DCIMVA are implementation defined.

ArchMsg.Warning.decode_fieldmismatch

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with different fields %{FIELD1} and %{FIELD2}. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_fracbitsnegative

TAGS **UNPREDICTABLE** DISPLAY VCVT Instruction is unpredictable with a negative number of fraction bits.

ArchMsg.Warning.decode_initblock

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable in an IT block.

ArchMsg.Warning.decode_initblocknotlast

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable in an IT block when not the last.

ArchMsg.Warning.decode_invalid_condition

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable when the condition is not AL.

ArchMsg.Warning.decode_invalidfieldcombination

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with the values in %{FIELD1} and %{FIELD2}. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_invalidvalue

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with the value in %{FIELD}. Fields:

FIELD enum

Field name.

ArchMsg.Warning.decode_mem_hint_unallocated

TAGS **UNPREDICTABLE** DISPLAY There is no memory hint allocated to this bit pattern.

ArchMsg.Warning.decode_registermatch

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with %{FIELD1} and %{FIELD2} as the same register. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_registermismatch

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with %{FIELD1} and %{FIELD2} as different registers. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_registeroutofrange

TAGS **UNPREDICTABLE** DISPLAY Floating-point instruction is unpredictable indexing beyond the end of the register bank.

ArchMsg.Warning.decode_sbzsbo

TAGS **UNPREDICTABLE** DISPLAY Reserved bits in the instruction are not canonical.

ArchMsg.Warning.decode_transactiontoobig

TAGS UNPREDICTABLE DISPLAY Attempt to perform a memory transaction of more than 128 bytes.

ArchMsg.Warning.decode_unpred_other

TAGS UNPREDICTABLE DISPLAY Instruction is unpredictable.

ArchMsg.Warning.decode_unpreduseofpc

TAGS UNPREDICTABLE DISPLAY Instruction is unpredictable when %{FIELD} is PC. Fields:

FIELD enum

Field name.

ArchMsg.Warning.decode_unpreduseofr13

TAGS UNPREDICTABLE DISPLAY Instruction is unpredictable when %{FIELD} is R13. Fields:

FIELD enum

Field name.

ArchMsg.Warning.decode_unpreduseofr13inreglist

TAGS UNPREDICTABLE DISPLAY Instruction is unpredictable with R13 in register-list.

ArchMsg.Warning.decode_useofd16orhigher

TAGS UNPREDICTABLE DISPLAY Attempt to use register D16 or higher where it is not allowed.

ArchMsg.Warning.decode_writebackandbaseinlist

TAGS UNPREDICTABLE DISPLAY Instruction is unpredictable with write back when the base register is in the transfer list.

ArchMsg.Warning.decode_zeroregistersinlist

TAGS UNPREDICTABLE DISPLAY Instruction is unpredictable when no registers are to be transferred.

ArchMsg.Warning.recursive_branch

An instruction is performing a branch that targets the same instruction. If this was intended then a WFI might be more effective DISPLAY recursive branch detected at PC=%{PC}. PRIMARY KEY PC. Fields:

PC unsigned int

Address of instruction causing the branch.

ArchMsg.Warning.recursive_exception

An instruction has generated an exception that targets the same instruction. DISPLAY recursive %{TYPE} exception detected at PC=%{PC} (CPSR=%{CPSR}). PRIMARY KEY PC. Fields:

CPSR unsigned int

Processor state.

PC unsigned int

Address of instruction causing the exception.

TYPE enum

Type of exception.

ArchMsg.Warning.reentrant_vector_catch

TAGS **UNPREDICTABLE** DISPLAY Vector catch debug event on Prefetch or Data abort vector %{ADDR}. Fields:

ADDR unsigned int

vector address.

ArchMsg.Warning.reserved_it_state

TAGS **UNPREDICTABLE** DISPLAY Execution with reserved IT state %{ITSTATE}. Fields:

ITSTATE unsigned int

reserved IT state.

ArchMsg.Warning.tlb_contents_unknown

Translation has been enabled without invalidating the TLB. DISPLAY Translation was enabled in the %{PAS:(Secure|Non-secure|Root|Realm)} EL%{EL:d} translation regime (vmid=%{VMID:x}) without invalidating %{INVALIDITY:(|d-side |i-side |)}entries since power-on: %{INVALIDITY:(|D|I|)}TLB could contain **UNKNOWN** content. Fields:

EL unsigned int

Exception level of translation regime.

INVALIDITY unsigned int

Which TLB may be invalid (if TLBs are separate).

NON_SECURE bool

Whether the translation regime is non-secure.

PAS unsigned int

Physical address space of translation regime.

VMID unsigned int

Current VMID, if applicable.

ArchMsg.Warning.unknown_DLR_DSPSR

Exiting debug state with unknown DLR or DSPSR.

ArchMsg.Warning.unknown_ELR_SPSR

Returning from debug exception with unknown ELR or SPSR.

ArchMsg.Warning.unpred_FAR_on_LRCPC3_non_atomic_mem_access

TAGS **UNPREDICTABLE** Unpredictable FAR/FSTATUS on LRCPC3 pair instructions resulting in two separate memory accessesDISPLAY Unpredictable FAR on LRCPC3 non atomic memory access at %{ADDR1} with FSTATUS %{FSTATUS1} or %{ADDR2} with FSTATUS %{FSTATUS2}. Fields:

ADDR1 unsigned int

Possible FAR address 1.

ADDR2 unsigned int

Possible FAR address 2.

FSTATUS1 unsigned int

Possible fault status value 1.

FSTATUS2 unsigned int

Possible fault status value 2.

ArchMsg.Warning.unpred_async_tag_check_fail_on_sve_speculative_load_do_update_ffr

Unpredictable set to FALSE of the FFR element associated to SVE speculative load with an async tag check fail TAGS **UNPREDICTABLE**.

ArchMsg.Warning.unpredictable_a32_breakpoint

TAGS **UNPREDICTABLE** DISPLAY Unpredictable breakpoint on a A32 instruction at %{ADDR1} Offset %{OFFSET} BAS %{BAS} for a %{IS_ADDRESS_MISMATCH_BREAKPOINT:(Mismatch|Match)} Breakpoint number %{N:d} Breakpoint address %{ADDR2}. Fields:

ADDR1 unsigned int

Address.

ADDR2 unsigned int

Breakpoint address.

BAS unsigned int

BAS field.

IS_ADDRESS_MISMATCH_BREAKPOINT bool

Is a Mismatch breakpoint, not a Match breakpoint.

N unsigned int

Breakpoint number.

OFFSET unsigned int

Always has a value of zero.

ArchMsg.Warning.unpredictable_banked_register_access

TAGS **UNPREDICTABLE** DISPLAY Access to banked register (R=%{R:d} SYSm=%{SYSM:2x}) is unpredictable. Fields:

R unsigned int

R bit.

SYSM unsigned int

SYSm.

ArchMsg.Warning.unpredictable_ccfail_undef

TAGS **UNPREDICTABLE** DISPLAY An instruction %{OPCODE} at VA=%{VADDR} failed its condition codes check but would otherwise have caused a non-data-dependent trap or undefined exception. Behaviour is implementation-defined choice of exception or nop. Fields:

OPCODE unsigned int

opcode of the instruction.

VADDR unsigned int

Virtual address of the instruction being executed.

ArchMsg.Warning.unpredictable_ici_bits

TAGS **UNPREDICTABLE** DISPLAY ICI bits do not apply or are architecturally unpredictable for instruction.

ArchMsg.Warning.unpredictable_t32_breakpoint

TAGS **UNPREDICTABLE** DISPLAY Unpredictable breakpoint on a T32 instruction at %{ADDR1} Offset %{OFFSET} BAS %{BAS} for a %{IS_ADDRESS_MISMATCH_BREAKPOINT:(Mismatch|Match)} Breakpoint number %{N:d} Breakpoint address %{ADDR2}. Fields:

ADDR1 unsigned int

Address.

ADDR2 unsigned int

Breakpoint address.

BAS unsigned int

BAS field.

IS_ADDRESS_MISMATCH_BREAKPOINT bool

Is a Mismatch breakpoint, not a Match breakpoint.

N unsigned int

Breakpoint number.

OFFSET unsigned int

Indicates which halfword of a 32-bit Thumb instruction the breakpoint is set on. 0 for the first halfword or if the instruction is 16 bits wide, or 1 for the second halfword.

ArchMsg.Warning.unpredictable_t32_vector_catch

TAGS **UNPREDICTABLE** DISPLAY Unpredictable Vector catch exception on a T32 instruction at %{ADDR} Offset %{OFFSET}. Fields:

ADDR unsigned int

Address.

OFFSET unsigned int

Indicates which halfword of a 32-bit Thumb instruction the vector catch exception is set on. 0 for the first halfword or if the instruction is 16 bits wide, or 1 for the second halfword.

ArchMsg.Warning.unpredictable_watchpoint_far

TAGS **UNPREDICTABLE** DISPLAY The value for FAR/EDWAR for hitting this watchpoint may be anywhere in the range %{LOWER_BOUND} to %{UPPER_BOUND}. Fields:

LOWER_BOUND unsigned int

the lowest address accessed by the instruction that triggered the watchpoint.

UPPER_BOUND unsigned int

the highest watchpointed address accessed by that instruction.

ArchMsg.Warning.unpredictable_watchpoint_far_isscmp

DISPLAY The value for FAR/EDWAR for hitting this watchpoint may be anywhere in the range %{LOWER_BOUND} to %{UPPER_BOUND}. Fields:

LOWER_BOUND unsigned int

the lowest address accessed by the instruction that triggered the watchpoint.

UPPER_BOUND unsigned int

the highest watchpointed address accessed by that instruction.

ArchMsg.Warning.warning_AdvSIMDEExpandImmUnexpectedZero

TAGS UNPREDICTABLE DISPLAY AdvSIMDEExpandImm may treat this immediate value as UNPREDICTABLE.

ArchMsg.Warning.warning_ConditionalSMC

TAGS UNPREDICTABLE DISPLAY SMC instruction has UNPREDICTABLE effects when conditional and when combined with traps.

ArchMsg.Warning.warning_TLBID_out_of_range

TLBID out of range in TLBI according to the configuration of NIS, NVIS, NOS and NVOS. Fields:

NIS unsigned int

NIS.

NOS unsigned int

NOS.

NVIS unsigned int

NVIS.

NVOS unsigned int

NVOS.

TLBID unsigned int

The TLBID value used in the TLBI instruction.

ArchMsg.Warning.warning_TTL_or_TG_mismatches_current_translation_granule

TTL or TG mismatches current translation granule in TLBI VA/RVA/IPA/RIPA. Fields:

CURRENT_TG unsigned int

Current TG as indicated by TCR_ELx or VTCR_EL2.

TLBI_TG unsigned int

TG (or TTL[3:2]) used in TLBI instruction.

ArchMsg.Warning.warning_access_crosses_page_boundary

TAGS UNPREDICTABLE DISPLAY Access crosses page boundary.

ArchMsg.Warning.warning_access_crosses_page_boundary_has_fault

TAGS UNPREDICTABLE DISPLAY Access that crosses page boundary has fault(s).

ArchMsg.Warning.warning_access_crosses_page_boundary_has_fault_on_both_pages

TAGS **UNPREDICTABLE** DISPLAY Access that crosses page boundary has a fault on each page.

Fields:

ADDR1 unsigned int

FAR address on first page.

ADDR2 unsigned int

FAR address on second page.

FSTATUS1 unsigned int

Fault status on first page.

FSTATUS2 unsigned int

Fault status on second page.

ArchMsg.Warning.warning_access_crosses_page_boundary_spanning_different_memory

TAGS **UNPREDICTABLE** DISPLAY Access at address %{ADDR} crosses page from MEM_TYPE %{MEMTYPE_PAGE1} to %{MEMTYPE_PAGE2}. Fields:

ADDR unsigned int

address of access crossing page.

MEMTYPE_PAGE1 signed int

First Page's memory type (0-Normal, 1-Non-Normal).

MEMTYPE_PAGE2 signed int

Second Page's memory type (0-Normal, 1-Non-Normal).

ArchMsg.Warning.warning_access_wraps_around_memory

TAGS **UNPREDICTABLE** DISPLAY Access wraps around memory in %{WIDTH:d}bit mode.

Fields:

WIDTH unsigned int

Address width.

ArchMsg.Warning.warning_bcr_linking_status

TAGS **UNPREDICTABLE** DISPLAY DBGBCR.{BT2,BT} combination gives unpredictable behaviour.

ArchMsg.Warning.warning_bcr_mask_reserved

TAGS **UNPREDICTABLE** DISPLAY DBGBCR_ELn.MASK is set to a reserved value.

ArchMsg.Warning.warning_bvr_masked_bit_not_zero

TAGS **UNPREDICTABLE** DISPLAY DBGBVRn_EL1.VA contains a bit masked by DBGBCRn_EL1.MASK that is not zero.

ArchMsg.Warning.warning_ccsidr_unimplemented_level

TAGS **UNPREDICTABLE** DISPLAY CCSIDR read while CSSELR %{CSSELR:x} points to an unimplemented cache level. Fields:

CSSELR unsigned int

current effective value of CSSELR.

ArchMsg.Warning.warning_change_to_ns_when_tge_set

TAGS **UNPREDICTABLE** DISPLAY Attempting to change NS to 1 when HCR.TGE = 1.

ArchMsg.Warning.warning_contiguous_bit_check_abort

TAGS **UNPREDICTABLE** DISPLAY An abort occurred whilst attempting to check TLB entry is contiguous at %{ENTRY_ADDR}. Fields:

ENTRY_ADDR unsigned int

address of conflicting TLB entry.

ArchMsg.Warning.warning_contiguous_bit_error

TAGS **UNPREDICTABLE** DISPLAY TLB entry at %{ENTRY_ADDR} was not contiguous with entry at %{CONTIG_ADDR} Contents are %{ENTRY_DATA} but expected %{CONTIG_DATA}. Fields:

CONTIG_ADDR unsigned int

address of the TLB first read, with which this entry is expected to be contiguous.

CONTIG_DATA unsigned int

expected contents based on the entry at CONTIG_ADDR.

ENTRY_ADDR unsigned int

address of conflicting TLB entry.

ENTRY_DATA unsigned int

contents of conflicting TLB entry.

ArchMsg.Warning.warning_contiguous_bit_error_D128

TAGS **UNPREDICTABLE** DISPLAY TLB entry at %{ENTRY_ADDR} was not contiguous with entry at %{CONTIG_ADDR} Contents are %{ENTRY_DATA_HI} %{ENTRY_DATA_LO} but expected %{CONTIG_DATA}. Fields:

CONTIG_ADDR unsigned int

address of the TLB first read, with which this entry is expected to be contiguous.

CONTIG_DATA unsigned int

expected contents based on the entry at CONTIG_ADDR.

ENTRY_ADDR unsigned int

address of conflicting TLB entry.

ENTRY_DATA_HI unsigned int

Higher bit contents of conflicting TLB entry.

ENTRY_DATA_LO unsigned int

Lower bit contents of conflicting TLB entry.

ArchMsg.Warning.warning_cp10_cp11_mismatch

TAGS **UNPREDICTABLE** DISPLAY Unpredictable configuration of CP10 and CP11 controls at EL %{EL:d}. Fields:

EL unsigned int

EL owning the control.

ArchMsg.Warning.warning_cp10_cp11_reserved_value

TAGS **UNPREDICTABLE** DISPLAY Unpredictable configuration of CPACR CP10 ({CP10}) or CP11 ({CP11}). Fields:

CP10 unsigned int

CP10 access.

CP11 unsigned int

CP11 access.

ArchMsg.Warning.warning_csselr_level_out_of_range

TAGS **UNPREDICTABLE** DISPLAY CSSELR.Level out of range, written value {LEVEL} with only {IMPLEMENTED:d} levels implemented. Fields:

IMPLEMENTED unsigned int

number of cache levels implemented.

LEVEL unsigned int

written value of CSSELR.Level.

TYPE enum

The type of cache selected by CSSELR.InD and TnD.

ArchMsg.Warning.warning_debug_flow_control_bits_not_obeyed

TAGS **UNPREDICTABLE** DISPLAY Flow control bits for {REGISTER} were not obeyed, causing {ERROR}. Fields:

ERROR enum

type of error resulting.

REGISTER enum

register under consideration.

ArchMsg.Warning.warning_debug_register_access_during_reset

TAGS **UNPREDICTABLE** DISPLAY Unpredictable {IS_WRITE:(read|write)} access to debug register offset {OFFSET} during reset. Fields:

IS_WRITE unsigned int

Write Not Read.

OFFSET unsigned int

Register Offset.

ArchMsg.Warning.warning_decode_cps_inconsistent_fields

TAGS **UNPREDICTABLE** DISPLAY The mode setting fields in the CPS instruction are inconsistent.

ArchMsg.Warning.warning_decode_invalid_state

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable in current exception-level/security state.

ArchMsg.Warning.warning_default_cacheable_mmu_on

TAGS **UNPREDICTABLE** DISPLAY Default cacheable is enabled (HCR.DC = 1) while MMU is enabled.

ArchMsg.Warning.warning_default_cacheable_vmmu_off

TAGS **UNPREDICTABLE** DISPLAY Default cacheable is enabled (HCR.DC = 1) while HCR.VM = 0.

ArchMsg.Warning.warning_deprecated_wvr_bit_2_set

TAGS **UNPREDICTABLE** DISPLAY DBGWVR%{N:d}_EL1=%{WVR:h} has bit 2 set, deprecated in ARMv8. Fields:

N unsigned int

index.

WVR unsigned int

DBGWVR.

ArchMsg.Warning.warning_eret_while_software_step_active_pending

Missing ISB between setting MDSCR_EL1.SS and ERET TAGS **UNPREDICTABLE**.

ArchMsg.Warning.warning_exclusive_to_non_normal

TAGS **UNPREDICTABLE** DISPLAY Exclusive Access to Non-Normal Memory %{ADDR}. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_exclusive_to_non_writeback

TAGS **UNPREDICTABLE** DISPLAY Exclusive Access to Non-Writeback-Cacheable Memory %{ADDR}. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_execute_from_device_memory

TAGS **UNPREDICTABLE** DISPLAY An attempt was made to execute from Device memory.

ArchMsg.Warning.warning_illegal_cpsr_mode

TAGS **UNPREDICTABLE** DISPLAY Writing an illegal or unimplement mode (%{NEW_MODE:h}) to CPSR was unpredictable. Fields:

NEW_MODE unsigned int

New value of CPSR.M.

ArchMsg.Warning.warning_illegal_srs_mode

TAGS **UNPREDICTABLE** DISPLAY Illegal or **UNPREDICTABLE** mode (%{MODE:h}) used for SRS instruction. Fields:

MODE unsigned int

mode for Banked SP.

ArchMsg.Warning.warning_implementation_defined_read_debug_register_in_SCS

Read Access to Debug registers from the processor is IMPLEMENTATION DEFINED TAGS IMP_DEF. Fields:

PPB_OFFSET unsigned int
debug register offset.

ArchMsg.Warning.warning_implementation_defined_sequential_security_transitions_supported

The behavior of sequential instruction fetches that cross from non-secure to secure memory and contain an SG instruction is CONSTRAINED UNPREDICTABLE TAGS UNPREDICTABLE.

ArchMsg.Warning.warning_implementation_defined_stack_limit_check_supported

It is IMPLEMENTATION DEFINED whether stack pointer limit checking is performed for this instruction TAGS IMP_DEF. Fields:

ADDRESS unsigned int
Address of instruction.

OPCODE unsigned int
instruction opcode.

ArchMsg.Warning.warning_implementation_defined_taggednonsharedmemorytype

TAGS IMP_DEF DISPLAY Memory attribute has Tagged, non shared memory type; it is IMPLEMENTATION DEFINED whether the memory location is treated as Tagged or non-Tagged.

ArchMsg.Warning.warning_implementation_defined_write_debug_register_in_SCS

Write Access to Debug registers from the processor is IMPLEMENTATION DEFINED TAGS IMP_DEF. Fields:

DATA unsigned int
data attempted to be written.

PPB_OFFSET unsigned int
debug register offset.

ArchMsg.Warning.warning_invalid_tcr_granule

TAGS UNPREDICTABLE DISPLAY TCR.TG%{TG_ID:(0|1)} bits have been set to a granule size %{{REQUEST:d}}K not implemented - using %{{SUBSTITUTE:d}}K. Fields:

REQUEST signed int
page size requested (or 0 for reserved).

SUBSTITUTE signed int
best guess available page size.

TG_ID bool
bits TG0 or TG1.

ArchMsg.Warning.warning_load_multiple_user_registers_from_user_mode

TAGS UNPREDICTABLE DISPLAY An LDM(user registers) instruction executed from user mode.

ArchMsg.Warning.warning_mte_in_mem_attr_not_supported

TAGS UNPREDICTABLE DISPLAY Memory attribute %{ATTR} has Tagged memory type but Armv8.5-MTE is not implemented. Fields:

ATTR unsigned int

Memory attribute.

ArchMsg.Warning.warning_reserved_breakpoint_state_match

TAGS UNPREDICTABLE DISPLAY %{IS_BREAKPOINT:(Watchpoint|Breakpoint)} programmed with a reserved combination of HMC, SSC and %{IS_BREAKPOINT:(PAC|PMC)}. Fields:

IS_BREAKPOINT unsigned int

Is a breakpoint, not a watchpoint.

ArchMsg.Warning.warning_shareability

TAGS UNPREDICTABLE DISPLAY Unpredictable: combination of the 1st and 2nd stages of translation is Normal Inner Non-Cacheable, Outer Non-Cacheable.

ArchMsg.Warning.warning_software_step_set_while_enabled

TAGS UNPREDICTABLE DISPLAY MDSCR_EL1.SS set to 1 while software step debug exceptions are enabled.

ArchMsg.Warning.warning_thumb_instruction_wraps_around_memory

TAGS UNPREDICTABLE DISPLAY Thumb instruction wraps around memory.

ArchMsg.Warning.warning_ttbr_sbz_bits_are_not_zero

TAGS UNPREDICTABLE DISPLAY %{TTBR}[%{MSB:d}:0] should be zero, but are not. Fields:

MSB unsigned int

x-1.

TTBR enum

which TTBR register.

ArchMsg.Warning.warning_unaligned_address_dbgdtrrx_write

TAGS UNPREDICTABLE DISPLAY Unaligned address %{ADDR} in X0/R0 when DBGDTRRX written in memory access mode. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unaligned_address_dbgdtrtx_read

TAGS UNPREDICTABLE DISPLAY Unaligned address %{ADDR} in X0/R0 when DBGDTRTX read in memory access mode. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unaligned_to_device

TAGS UNPREDICTABLE DISPLAY Unaligned Access to Device Memory %{ADDR}. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unaligned_to_strongly_ordered

TAGS **UNPREDICTABLE** DISPLAY Unaligned Access to Strongly Ordered Memory %{ADDR}.

Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unknown_sau_rnr

TAGS **UNPREDICTABLE** DISPLAY SAU_RNR was set to an unsupported value.

ArchMsg.Warning.warning_unpredictable_AIRCR_PRIS_and_BFHFNMINs

TAGS **UNPREDICTABLE** DISPLAY The effect of setting both AIRCR.BFHFNMINs and AIRCR.PRIS to 1 is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_AIRCR_VECTCLRACTIVE_when_not_in_debug

TAGS **UNPREDICTABLE** DISPLAY The effect of writing a 1 to AIRCR.VECTCLRACTIVE if the processor is not halted in Debug state is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_AIRCR_VECTRESET_and_SYSRESETREQ

TAGS **UNPREDICTABLE** DISPLAY When the processor is halted in Debug state, if a write to the register writes a 1 to both VECTRESET and SYSRESETREQ, the behavior is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_AIRCR_VECTRESET_when_not_in_debug

TAGS **UNPREDICTABLE** DISPLAY The effect of writing a 1 to AIRCR.VECTRESET if the processor is not halted in Debug state is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_AIRCR_incorrect_VKEY

TAGS **UNPREDICTABLE** DISPLAY The value 0x05FA must be written to AIRCR.VKEY, otherwise the register write is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_EXC_RETURN_Reserved_Bit

EXC_RETURN[23:7] are reserved with the special condition that all bits should be written as one. Values other than all 1s are **UNPREDICTABLE**. TAGS **UNPREDICTABLE**. Fields:

EXC_RETURN unsigned int

Special PC value.

SBO_MASK unsigned int

Which bits Should Be One to not UNPRED.

ArchMsg.Warning.warning_unpredictable_PMEVTYPEPn_ELO_VS_reserved_value

TAGS **UNPREDICTABLE** DISPLAY Unpredictable reserved value of PMEVTYPEN<n>_ELO.VS.

ArchMsg.Warning.warning_unpredictable_change_to_DEMCR_MON_STEP_at_insufficient_priority

TAGS **UNPREDICTABLE** DISPLAY The effect of changing DEMCR.MON_STEP at an execution priority that is lower than the priority of the DebugMonitor exception is **CONSTRAINED UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_change_to_priority_of_active_exception

TAGS UNPREDICTABLE DISPLAY Changing the priority of an active exception is UNPREDICTABLE.

ArchMsg.Warning.warning_unpredictable_clear_lspact

TAGS UNPREDICTABLE DISPLAY UNPREDICTABLE state on context switch. Fields:

DRVEXC_TAKEN unsigned int

Derived exception taken, not pended.

REACHED_LAST_STORE unsigned int

Only the last faulted.

ArchMsg.Warning.warning_unpredictable_dbg_exit_unaligned_dlr

TAGS UNPREDICTABLE DISPLAY Debug exit to AArch32 with DLR[0] set to 1. PC[0] can be set to 0 or to DLR[0].

ArchMsg.Warning.warning_unpredictable_exception_catch

TAGS UNPREDICTABLE DISPLAY Generation of an exception catch event is unpredictable.

ArchMsg.Warning.warning_unpredictable_exception_return_inconsistent_state

TAGS UNPREDICTABLE DISPLAY state on exception return is unpredictable.

ArchMsg.Warning.warning_unpredictable_exception_return_instruction

TAGS UNPREDICTABLE DISPLAY exception return instruction is unpredictable.

ArchMsg.Warning.warning_unpredictable_in_debug_state

TAGS UNPREDICTABLE DISPLAY Instruction is UNPREDICTABLE when executed in debug state.

ArchMsg.Warning.warning_unpredictable_mair_encoding

TAGS UNPREDICTABLE DISPLAY Memory attribute %{ATTR} has invalid MAIR encoding; this may lead to unpredictable results. Fields:

ATTR unsigned int

Memory attribute.

ArchMsg.Warning.warning_unpredictable_pmu_counter_access

TAGS UNPREDICTABLE DISPLAY Access to PMU counters from non-secure EL0 or EL1 is UNPREDICTABLE with MDCR_EL2.HPMN set to 0.

ArchMsg.Warning.warning_unpredictable_prioritization_breakpoint_match_vector_catch

TAGS UNPREDICTABLE DISPLAY Two events with the same priority occurred at the same instruction: Address Matching Vector Catch debug event, and Breakpoint debug event. It is constrained unpredictable which is taken.

ArchMsg.Warning.warning_unpredictable_stack_selection

TAGS UNPREDICTABLE DISPLAY UNPREDICTABLE state on context switch.

ArchMsg.Warning.warning_unpredictable_tsize_out_of_range

TAGS UNPREDICTABLE DISPLAY %{TCR}.T%{TTBR:d}SZ (%{TSIZE:d} bits) is out of range. Must be between 25 and %{TSIZE_MAX:d}. Fields:

TCR enum

Which TCR.

TSize signed int

Value of TSize.

TSize_MAX signed int

The maximum allowed value.

TTBR unsigned int

Which TTBR region.

ArchMsg.Warning.warning_unpredictable_unaligned_pc_as_base_register

TAGS **UNPREDICTABLE** DISPLAY The PC value must be word-aligned, otherwise the behavior of the instruction is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_unaligned_pop_stack

TAGS **UNPREDICTABLE** DISPLAY **CONSTRAINED UNPREDICTABLE** unaligned stack frame `%{FRAME_ADDR:x}` in exception-return. Fields:

FRAME_ADDR unsigned int

unaligned stack frame address.

ArchMsg.Warning.warning_unpredictable_unaligned_to_non_writeback

TAGS **UNPREDICTABLE** DISPLAY Unaligned Access to Non-Writeback-Cacheable Memory.

ArchMsg.Warning.warning_unpredictable_vmsa_memattrib

TAGS **UNPREDICTABLE** DISPLAY Unpredictable vmsa memory attribute with texcb value programmed to `%{TEXCB}` when tex remap is `%{TEX_REMAP}`. Fields:

TEXCB unsigned int

texcb value.

TEX_REMAP bool

use tex remap.

ArchMsg.Warning.warning_unpredictable_vtcr_t0sz_sl0

TAGS **UNPREDICTABLE** DISPLAY The combination of VTCR.TOSZ=`%{TOSZ:d}` and VTCR.SL0=`%{SL0:d}` is unpredictable. Fields:

SL0 signed int

VTCR.SL0, starting level for stage 2 translation table walks.

TOSZ signed int

VTCR.TOSZ, required input address range.

ArchMsg.Warning.warning_unpredictable_write_DHCSR

TAGS **UNPREDICTABLE** DISPLAY **UNPREDICTABLE** set of bit-changes in DHCSR. Fields:

NEW unsigned int

data attempted to be written.

OLD unsigned int

previous value.

ArchMsg.Warning.warning_unpredictable_write_SHCSR

TAGS UNPREDICTABLE DISPLAY UNPREDICTABLE setting a pending bit to 1 for an exception with priority \geq execution priority.

ArchMsg.Warning.warning_unsupported_access_to_memory_mapped_register

TAGS UNPREDICTABLE DISPLAY Access to memory mapped register is unsupported.

ArchMsg.Warning.warning_user_jmcr_access

TAGS UNPREDICTABLE DISPLAY User %{WRITE:(write|read)} access to JMCR is unpredictable. Fields:

WRITE unsigned int
access is write.

ArchMsg.Warning.warning_wcr_mask_and_bas

TAGS UNPREDICTABLE DISPLAY DBGWCR_EL1.MASK is non-zero and BAS is not 0xff.

ArchMsg.Warning.warning_wcr_mask_reserved

TAGS UNPREDICTABLE DISPLAY DBGWCR_ELn.MASK is set to a reserved value.

ArchMsg.Warning.warning_wcr_non_configuous_bas

TAGS UNPREDICTABLE DISPLAY DBGWCR_EL1.BAS has non-contiguous set of ones.

ArchMsg.Warning.warning_wvr_masked_bit_not_zero

TAGS UNPREDICTABLE DISPLAY DBGWVRn_EL1.VA contains a bit masked by DBGWCRn_EL1.MASK that is not zero.

ArchMsg.Why.why_branch_target_exception

A branch target exception has occurred. DISPLAY branch target exception: target pc=%{PC}, target opcode=%{OPCODE}. PRIMARY KEY PC. Fields:

BTYPE unsigned int
pstate.BTYPE at time branch taken.

OPCODE unsigned int
opcode at branch target.

PC unsigned int
Virtual Address of branch target.

ArchMsg.Why.why_illegal_state

An illegal mode change has occurred. DISPLAY illegal mode change: %{MESSAGE}. Fields:

MESSAGE string
The message.

BRANCH_MISPREDICT

Simulating branch mispredict. Fields:

PC unsigned int
Origin address (or 0 if unavailable).

BRA_DIR

Direct branches, to immediate address. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISET enum

The instruction set of the branch instruction.

IS_COND bool

Indicates if this is a conditional branch.

IS_HINTED bool

Whether the conditional instruction is hinted.

IS_ISB bool

Whether the instruction is ISB.

IS_LINK bool

Indicates if this is a branch with link.

PC unsigned int

The address of the branch instruction.

TARGET_ISET enum

The instruction set after the branch.

TARGET_PC unsigned int

The address the instruction branches to.

BRA_INDIR

Indirect branches, perhaps to a register. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISET enum

The instruction set of the branch instruction.

IS_COND bool

Indicates if this is a conditional branch.

IS_HINTED bool

Whether the conditional instruction is hinted.

IS_ISB bool

Whether the instruction is ISB.

IS_LINK bool

Indicates if this is a branch with link.

PC unsigned int

The address of the branch instruction.

TARGET_ISET enum

The instruction set after the branch.

TARGET_PC unsigned int

The address the instruction branches to.

BRA_INDIR_CCFAIL

Indirect branches fail, perhaps to a register.

BRA_RET_CCFAIL

Branch return fail, perhaps to a register.

CACHE_MAINTENANCE_OP

Cache Maintenance Operation. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

DATA unsigned int

Specified MVA or Set/way.

FUNCTION enum

Clean / Invalidate.

SCOPE enum

Affected region.

SIDE enum

Inst / Data.

CCFAIL

Conditional instruction condition check fail. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CCFAIL_UNC

Unconditional instruction condition check fail. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CCPASS

Conditional instruction condition check pass. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CCPASS_UNC

Unconditional instruction condition check pass. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CODE_CACHE_MAINT

Code cache maintenance. Fields:

END_INCL unsigned int

Inclusive end address for VA or PA requests.

NS bool

Non-secure world for PA.

START unsigned int

Start address for VA or PA requests.

TYPE enum

Request type.

COMPILE_BLOCK_END

Last instruction of basic block translated. Fields:

VADDR unsigned int

Address of next instruction after this basic block.

COMPILE_BLOCK_START

First instruction of basic block translated. Fields:

VADDR unsigned int

Address of first instruction in basic block.

COMPILE_INST

ARM instruction compiled. Fields:

DISASS string

Disassembly of the instruction.

ISSET enum

The instruction set of this instruction.

ITSTATE unsigned int

The ITSTATE current for the instruction.

OPCODE unsigned int

The opcode of the instruction.

PC unsigned int

The address of the instruction.

SIZE unsigned int

The size of the instruction in bytes.

CONTEXTIDR

Context ID Register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

NS bool

Secure or nonsecure banked register is accessed.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The new value written.

CONTEXT_SYNC

Called for every context synchronization event. Fields:

ARCH_CSE bool

Architectural Context Synchronization Event.

PC unsigned int

The address of the synchronization instruction.

SYSREG_SYNC bool

System register synchronization event.

CORE_ENDIAN

Core BE8 Big-Endian state changed. Fields:

BE8 bool

Core BE8 Big-Endian state.

CORE_INFO

Static processor attributes. Only triggered by a call to DumpState(). Fields:

ARCH_PROFILE enum

The architecture profile of the core.

CLUSTER_ID unsigned int

The cluster ID of this processor.

CORE_NUM unsigned int

The number of this core in an MP processor.

DETAILED_CORE_FEATURES enum

Detailed core features as a bitfield. Enum values are bitmasks to extract the info from the field.

FPU_VERSION enum

The VFP version implemented by the core.

MEM_ARCH enum

The memory architecture of the core.

MIDR_EL1 unsigned int

MIDR_EL1 value.

NUM_CORES unsigned int

The number of cores in this MP processor.

QUANTUM_SIZE unsigned int

The default quantum size of the core (in ticks).

SECURITY_FEATURES bool

Does the core have security features?.

CORE_LOADS

Processor load accesses. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ACQREL enum

Is this an acquire/release.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DATA unsigned int

The data read or written.

ELEMENT_SIZE unsigned int

Width of each element.

ENDIANNESS enum

Byte ordering of this access.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

LOCK enum

Normal, exclusive or locked access.

MECID unsigned int

Memory Encryption Context ID, if applicable.

MECID2 unsigned int

Memory Encryption Context ID of the second page, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PAGE_SIZE unsigned int

Page size as $\log_2(\text{size in bytes})$.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK, 2=Exclusive Failed, 3=Unsupported.

SIZE unsigned int

Width of the access in bytes. Only required if DATA is not traced.

TRANS bool

Is this a translated access.

VADDR unsigned int

The virtual address of the access.

CORE_REGS

Changes of the core registers R0 to R14. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number, 0 to 14.

MODE enum

Bank of the register accessed.

OLD_VALUE unsigned int

The old value overwritten.

PHYS_ID enum

The physical register accessed.

VALUE unsigned int

The new value written to the register.

CORE_REGS64

Changes of the core registers X0..X30, SP_ELn. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ID enum

The register number, 0-30 for X0-X30, >=32 for SP_ELn.

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

CORE_REGS64_READ

General purpose AArch64 register has been read X0..X30, SP_ELn. Fields:

ID enum

The register number, 0-30 for X0-X30, >=32 for SP_ELn.

VALUE unsigned int

The value read from the register.

CORE_STALLED_TRANSITION

Detect when a core is prevented from progressing PChannel states. Fields:

CURRENT_STATE enum

Current power mode of core.

REASON string

The reason a transition was stalled, and what must be done to unblock it.

TARGET_STATE enum

Desired power mode of core after transition.

CORE_STORES

Processor store accesses. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ACQREL enum

Is this an acquire/release.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DATA unsigned int

The data read or written.

ELEMENT_SIZE unsigned int

Width of each element.

ENDIANNESS enum

Byte ordering of this access.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

LOCK enum

Normal, exclusive or locked access.

MECID unsigned int

Memory Encryption Context ID, if applicable.

MECID2 unsigned int

Memory Encryption Context ID of the second page, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PAGE_SIZE unsigned int

Page size as $\log_2(\text{size in bytes})$.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK, 2=Exclusive Failed, 3=Unsupported.

SIZE unsigned int

Width of the access in bytes. Only required if DATA is not traced.

TRANS bool

Is this a translated access.

VADDR unsigned int

The virtual address of the access.

CP14_READ

System Coprocessor register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP14 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc1 unsigned int

opcode 1.

opc2 unsigned int

opcode 2.

CP14_WRITE

System Coprocessor register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP14 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc1 unsigned int

opcode 1.

opc2 unsigned int

opcode 2.

CP15_READ

System Coprocessor register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP15 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc1 unsigned int

opcode 1.

opc2 unsigned int

opcode 2.

CP15_READ64

System Coprocessor register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP15 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc unsigned int

opcode 1.

CP15_WRITE

System Control Coprocessor register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP15 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc1 unsigned int

opcode 1.

opc2 unsigned int
opcode 2.

CP15_WRITE64

System Coprocessor register write. Fields:

CORE_NUM unsigned int
Core number in a multi processor.

CRm unsigned int
CRm.

NS bool
Secure or nonsecure banked register is accessed.

REG_NAME string
Name of the CP15 register accessed.

SECURITY_STATE enum
The current security state.

UNDEF bool
The register accessed is undefined.

UPDATED_VALUE unsigned int
Updated value of the register now it has been written.

VALUE unsigned int
The new value written.

opc unsigned int
opcode 1.

CPSR

CPSR change. (In AArch64 PSTATE represented in SPSR format). Fields:

CORE_NUM unsigned int
Core number in a multi processor.

OLD_VALUE unsigned int
The old CPSR value.

UNKNOWN unsigned int
Bits within the register that have unknown value.

VALUE unsigned int
The new CPSR value.

CRYPTO_SPEC

Every crypto instruction speculatively executed.

CorePowerStateChange

Core power-up state has changed. DISPLAY Core powered %{PU:(up|down)}. Fields:

PU bool

Whether the core powered up.

DATA_CACHE_SET_ALLOC_TAGS

Set allocation tags from DC GVA instruction. Fields:

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

NSDESC unsigned int

The security state of the access.

PADDR unsigned int

The physical address of the access.

PAS enum

The physical address space of the page.

RESPONSE enum

0=Aborted, 1=OK.

SIZE unsigned int

The size of memory in bytes.

TAG unsigned int

The value of the tag being set.

TAGGING_TYPE enum

Type of Memory tagging (Physical or Virtual).

TAG_VADDR unsigned int

The virtual address of the Virtual Tag if Virtual tagging is used; otherwise 0.

VADDR unsigned int

Virtual address used for setting allocation tags.

DATA_CACHE_ZERO

Zero and invalidate cache line from DC ZVA instruction. Fields:

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

NSDESC unsigned int

The security state of the access.

PADDR unsigned int

The physical address of the access.

PADDR_TAG unsigned int

If tag checked, the physical address tag for the access.

PAS enum

The physical address space of the page.

RESPONSE enum

0=Aborted, 1=OK.

SIZE unsigned int

The size of zero'd memory in bytes.

TAG_CHECKED bool

Is this access tag checked.

VADDR unsigned int

Virtual address of the zero'd cache line.

DEBUG_EVENT

Hardware debug support event. Fields:

EVENT enum

Description of event.

VALUE unsigned int

data value.

DEBUG_TARGET_ISET

Hardware debug target ISA. Fields:

TARGET_ISET enum

The target instruction set of the debug state.

DMI_ALLOCATE

DMI allocated. Fields:

ACCESS_DIRECTION enum

READ / WRITE.

PADDR unsigned int

The physical address of the DMI.

PTR unsigned int

DMI range pointer.

SIDE enum

DSIDE / ISIDE.

SIZE unsigned int

The size of DMI.

VADDR unsigned int

The virtual address of the DMI..

DMI_HIT

DMI hit, access used DMI (only fire for small DMI regions i.e. less than a full page). Fields:

ACCESS_DIRECTION enum

READ / WRITE.

PADDR unsigned int

The physical address of the DMI.

SIDE enum

DSIDE / ISIDE.

SIZE unsigned int

The size of the DMI.

VADDR unsigned int

The virtual address of the DMI..

DMI_REVOKE

DMI revoked. Fields:

ACCESS_DIRECTION enum

READ / WRITE.

PADDR unsigned int

The physical address of the DMI.

SIDE enum

DSIDE / ISIDE.

SIZE unsigned int

The size of DMI.

VADDR unsigned int

The virtual address of the DMI..

END_COMPILE

Compilation end. Fields:

END_OF_PAGE_COUNT unsigned int

Number of basic blocks exited due to page boundary since START_COMPILE.

FETCHFAIL_COUNT unsigned int

Number of basic blocks exited due to ifetch failure START_COMPILE.

INST_COUNT unsigned int

Number of instructions compiled since START_COMPILE.

NONSEQ_COUNT unsigned int

Number of basic blocks exited due to non-sequential instructions since START_COMPILE.

PAGE_STRADDLE_COUNT unsigned int

Number of basic blocks exited due to unaligned instructions crossing page since START_COMPILE.

ESB

Error Synchronization Barrier hit. Fields:

ASYNC bool

ESB is async.

IMPLICIT bool

ESB is implicit.

EXCEPTION

Exception taken. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ESR unsigned int

The value assigned to ESR when exception is taken.

ESR_EC unsigned int

The Value assigned to ESR.EC when exception is taken.

IS_PHYSICAL bool

Physical or Virtual exception.

IS_TRAP bool

Exception coming from a trap.

LR unsigned int

The value assigned to the link register.

PC unsigned int

The location where the exception occurred.

PREFERRED_RETURN unsigned int

The preferred return address for the exception.

TAKEN_LOCALLY bool

Whether exception is taken locally(same exception level).

TARGET_ISET enum

The instruction set of the exception handler code.

TARGET_PC unsigned int

The address the exception branches to.

VECTOR enum

The exception vector.

VECTOR_OFFSET unsigned int

The offset of an exception vector from the base address.

EXCEPTION_END

Every exception completed.

EXCEPTION_INFO

Trace human-readable reason of the current exception for data/instruction aborts, floating point exceptions and unknown exceptions. Fields:

DESCRIPTION string

Reason for current exception.

ESR_EC unsigned int

The value of ESR.EC when exception is taken.

EXCEPTION_RAISE

Every exception raised.

EXCEPTION_RETURN

Branches on leaving exception. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISSET enum

The instruction set of the branch instruction.

IS_COND bool

Indicates if this is a conditional branch.

IS_HINTED bool

Whether the conditional instruction is hinted.

IS_ISB bool

Whether the instruction is ISB.

IS_LINK bool

Indicates if this is a branch with link.

PC unsigned int

The address of the branch instruction.

TARGET_ISET enum

The instruction set after the branch.

TARGET_PC unsigned int

The address the instruction branches to.

EXCEPTION_RETURN_PREBRANCH

Exception return event that must occur before PC and processor state have updated.

EXCEPTION_START

Every exception started.

FIQ_TAKEN

FIQ taken exception.

FP_EXCEPTIONS_SIGNALLED

Trace which FP exceptions were signalled as a part of the FP operations performed by an instruction. Fields:

FP_EXCEPTIONS_SIGNALLED_MASK unsigned int

Indicates which FP exceptions were signalled by an instruction. This uses the same encoding as the cumulative floating-point exception bitfields in FPSR.

FP_STATE

Floating point state. Fields:

FPSR unsigned int

FPSR register value.

MASK unsigned int

The mask indicating updated cumulative bits.

FREQ_CHANGED

Core frequency changed. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The instruction count of this core.

NEW_FREQ signed int

The new frequency of this core (expressed in Hz).

OLD_FREQ signed int

The old frequency of this core (expressed in Hz).

HLT

HLT instruction occurred. Fields:

IMM16 unsigned int

The 16-bit immediate value encoded in HLT instruction.

INST

Every instruction executed. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DEBUG_STATE bool

The instruction is executed in debug state.

DISASS string

Disassembly of instruction.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISET enum

The current instruction set.

ITSTATE unsigned int

The current ITSTATE.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

MODE enum

The mode the core is in.

NS unsigned int

The core's non-secure bit.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

OPCODE unsigned int

The opcode of the instruction.

PADDR unsigned int

The physical address of the instruction.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the instruction.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the page.

PC unsigned int

The address of the instruction.

SECURITY_STATE enum

The core's security state.

SIZE unsigned int

The size of the instruction in bytes.

INST_END

Every instruction completed.

INST_START

Every instruction started. Fields:

DEBUG_STATE **bool**

The instruction is executed in debug state.

ISET **enum**

The current instruction set.

MODE **enum**

The mode the core is in.

NS **enum**

The current Secure State.

PC **unsigned int**

The address of the conditional instruction.

SECURITY_STATE **enum**

The current Security State.

INST_STRADDLE

Instruction straddles boundary. Fields:

DEBUG_STATE **bool**

The instruction is executed in debug state.

ISET **enum**

The current instruction set.

MODE **enum**

The mode the core is in.

NS **enum**

The current Secure State.

PC **unsigned int**

The address of the conditional instruction.

SECURITY_STATE **enum**

The current Security State.

IRQ_TAKEN

IRQ taken exception.

ISB

ISB instruction occurred. Fields:

PC **unsigned int**

The address of the ISB instruction.

LOAD_MULTIPLE_REGS_LIST

List of destination registers (not source) of a multiple registers load instruction. Fields:

REG string

Register to add into the list.

LOCAL_MONITOR

Local monitor activity. Fields:

PADDR unsigned int

Local Monitor Address.

State enum

State of the monitor (Open/Exclusive).

MAPPING_CREATE

A memory mapping has been created. This is an implementation detail of the model and so may change without notice. Fields:

ASID unsigned int

The ASID used during translation, or 0 if no ASID was used.

IS_VALID bool

Whether the mapping is valid or not.

PADDR unsigned int

Base physical address of region, or 0 if not known.

SIDE enum

I-side or D-side.

SIZE unsigned int

Size of the page, or 0 if not known.

VADDR unsigned int

Base virtual address of region.

VMID unsigned int

The VMID used during translation, or 0 if no VMID was used.

MAPPING_DESTROY

A memory mapping has been destroyed. This is an implementation detail of the model and so may change without notice. Fields:

ASID unsigned int

The ASID used during translation.

CAUSE enum

The cause of invalidation.

PADDR unsigned int

Base physical address of page.

SIDE enum

I-side or D-side.

SIZE unsigned int

Size of the page.

VADDR unsigned int

Base virtual address of page.

VMID unsigned int

The VMID used during translation.

MEMMAP_DEBUG_READ

Memory mapped reads to the debug registers. Fields:

ADDR unsigned int

address.

CORE_NUM unsigned int

Core number in a multi processor.

EXT bool

Whether access is from an external device (such as the DAP).

NS enum

Secure state of the access.

REG_NAME string

Name of the debug register accessed.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

MEMMAP_DEBUG_WRITE

Memory mapped writes to the debug registers. Fields:

ADDR unsigned int

address.

CORE_NUM unsigned int

Core number in a multi processor.

EXT bool

Whether access is from an external device (such as the DAP).

NS enum

Secure state of the access.

REG_NAME string

Name of the debug register accessed.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The new value written.

MMU_D128_TTB_READ

This event is triggered by reads caused by a D128 translation table walk. Fields:

ABORTED bool

Was the walk successful.

DATA_HIGH unsigned int

The data upper bits read.

DATA_LOW unsigned int

The data lower bits read.

IPA unsigned int

For stage 1, the IPA of the read.

LEVEL unsigned int

Translation table level.

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

OUTADDR unsigned int

The output physical address.

PADDR unsigned int

The physical address of the read.

PASmem enum

Physical Address Space of the accessed memory.

PASreq enum

Physical Address Space of the requested lookup.

REGIME_EL enum

Entry matches in this translation regime.

REQUESTER enum

The requester of this table walk.

SIDE enum

Inst / Data.

STAGE unsigned int

Translation stage.

MMU_D128_TTB_WRITE

This event is triggered by writes caused by a D128 translation table walk. Fields:

ABORTED bool

Was the walk successful.

DATA_HIGH unsigned int

The data upper bits written.

DATA_LOW unsigned int

The data lower bits written.

IPA unsigned int

For stage 1, the IPA of the write.

IS_HACDBS_ACCESS bool

If the access was due to HACDBS_CLEAN.

LEVEL unsigned int

Translation table level.

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

OUTADDR unsigned int

The output physical address.

PADDR unsigned int

The physical address of the write.

PASmem enum

Physical Address Space of the accessed memory.

PASreq enum

Physical Address Space of the requested lookup.

REGIME_EL enum

Entry matches in this translation regime.

SIDE enum

Inst / Data.

STAGE unsigned int

Translation stage.

MMU_TRANS

Address translation information. Fields:

ASID unsigned int

Address space identifier.

CORE_NUM unsigned int

Core number in a multi processor.

Hyp bool

Entry matches in Hyp state only.

INNERCACHE_RA bool

Is the inner cache allocate on read.

INNERCACHE_TRANSIENT bool

Is the inner write-through transient.

INNERCACHE_TYPE enum

Inner Caching scheme (NC/MB/WA).

INNERCACHE_WA bool

Is the inner cache allocate on write.

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MEMTYPE enum

Memory type.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC enum

Is secure side supposed to access secure or nonsecure memory.

OUTERCACHE_RA bool

Is the outer cache allocate on read.

OUTERCACHE_TRANSIENT bool

Is the outer write-through transient.

OUTERCACHE_TYPE enum

Outer Caching scheme (NC/MB/WA).

OUTERCACHE_WA bool

Is the outer cache allocate on write.

PADDR unsigned int

Physical address of the access.

PAGESIZE unsigned int

Page size as log2(size).

PAS enum

PAS supposed to be accessed is secure or nonsecure memory or Root/Realm if RME is implemented.

SH enum

Shareability.

SIDE enum

Inst / Data.

STAGE1_PERM unsigned int

Stage 1 permissions, [5:3] Privileged access (XWR) [2:0] User access (XWR).

STAGE2_PERM unsigned int

Stage 2 permission mask [5:3] Privileged access (XWR) [2:0] User access (XWR).

VADDR unsigned int

Virtual address of the access.

VMID unsigned int

Virtual machine identifier.

XS bool

XS Attribute.

nG enum

Flag indicating whether ASID will be matched.

MMU_TTB_READ

This event is triggered by reads caused by a translation table walk. Fields:

ABORTED bool

Was the walk successful.

DATA unsigned int

The data read.

DESC_KIND enum

The encoding scheme of the descriptors used for this walk.

IPA unsigned int

For stage 1, the IPA of the read.

LEVEL unsigned int

Translation table level.

LPAE bool

Is this for an LPAE translation (*DEPRECATED*, use *DESC_KIND*).

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSmem bool

Non secure memory access.

NSreq bool

Non secure request.

OUTADDR unsigned int

The output physical address.

PADDR unsigned int

The physical address of the read.

PASmem enum

Physical Address Space of the accessed memory.

PASreq enum

Physical Address Space of the requested lookup.

REGIME_EL enum

Entry matches in this translation regime.

REQUESTER enum

The requester of this table walk.

SIDE enum

Inst / Data.

STAGE unsigned int

Translation stage.

MMU_TTB_WRITE

This event is triggered by writes caused by a translation table walk. Fields:

ABORTED bool

Was the walk successful.

DATA unsigned int

The data written.

DESC_KIND enum

The encoding scheme of the descriptors used for this walk.

IPA unsigned int

For stage 1, the IPA of the write.

IS_HACDBS_ACCESS bool

If the access was due to HACDBS_CLEAN.

LEVEL unsigned int

Translation table level.

LPAE bool

Is this for an LPAE translation (*DEPRECATED*, use DESC_KIND).

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSmem bool

Non secure memory access.

NSreq bool

Non secure request.

OUTADDR unsigned int

The output physical address.

PADDR unsigned int

The physical address of the write.

PASmem enum

Physical Address Space of the accessed memory.

PASreq enum

Physical Address Space of the requested lookup.

REGIME_EL enum

Entry matches in this translation regime.

SIDE enum

Inst / Data.

STAGE unsigned int

Translation stage.

MODE_CHANGE

Mode change. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

MODE enum

The new mode.

NON_SECURE enum

The core's new non-secure bit.

OLD_MODE enum

The old mode.

SECURITY_STATE enum

The new security state.

MPU_TRANS

Address translation information. Fields:

ACCESS_STATUS enum

Access status(0xAA/0xBB/0xDD/0xEE=overlap/bg hit/default/bg fault).

CORE_NUM unsigned int

Core number in a multi processor.

EXEC_PERM enum

Execution Permission.

INNERCACHE_RA bool

Is the inner cache allocate on read.

INNERCACHE_TYPE enum

Inner Caching scheme (NC/MB/WA).

INNERCACHE_WA bool

Is the inner cache allocate on write.

MEMTYPE enum

Memory type.

NSDESC enum

Is non secure access.

OUTERCACHE_RA bool

Is the outer cache allocate on read.

OUTERCACHE_TYPE enum

Outer Caching scheme (NC/MB/WA).

OUTERCACHE_WA bool

Is the outer cache allocate on write.

PADDR unsigned int

Address of the access.

PAGESIZE unsigned int

Page size as $\log_2(\text{size})$.

PAS enum

PAS accessed is secure or nonsecure memory.

READ_PERM enum

Read Permission.

REG_NUM unsigned int

MPU region-number 0-31.

SH enum

Shareability.

SIDE enum

Inst / Data.

STAGE unsigned int

The translation stage.

WRITE_PERM enum

Write Permission.

NOT_TAG_CHECKED_INST

MTE Tag not checked for current instruction.

PERIODIC

Called for every quantum. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The instruction count of this CPU.

PC unsigned int

The address of the next instruction to be executed on this CPU.

PMU_COUNTER_OVERFLOW

PMU counter overflow. Fields:

COUNTER_GROUP enum

Counter group.

COUNTER_RANGE enum

Counter range.

EVENT_ID unsigned int

Event Identifier, field is valid only for Event counter group.

INDEX unsigned int

Counter index (as selected by PMSELR), field is valid only for Event/Cycle counter groups.

INTERRUPT bool

Is interrupt enabled on overflow for this counter.

POTINDEX_UPDATE

Value of POT index for current pc changed. Fields:

NEW_POTINDEX unsigned int

New POTIndex.

OLD_POTINDEX unsigned int

Old POTIndex.

PREFETCH_MEMORY64

Prefetch from PRFM or PRFUM instructions. Fields:

PRFOP enum

Prefetch hint.

VADDR unsigned int

Virtual address of the location that should be prefetched.

PRELOAD64_REQUEST

Preload requests from PRFM or PRFUM instructions. Fields:

PRFOP enum

Prefetch hint.

VADDR unsigned int

Virtual address of the location that should be prefetched.

PRELOAD_DATA

Data preload from PLD instruction. Fields:

VADDR unsigned int

Virtual address of the data that should be preloaded.

PRELOAD_INST

Instruction preload from PLI instruction. Fields:

VADDR unsigned int

Virtual address of the instruction that should be preloaded.

PRE_CORE_LOAD

Trace just before a core load. Fields:

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

PRE_CORE_STORE

Trace just before a core store. Fields:

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

PRE_TTB_READ

This event is triggered before reads caused by a translation table walk. Fields:

ADDR unsigned int

The physical address of the read.

PSTATE

PSTATE fields update. Fields:

ALLINT unsigned int

All interrupt mask bit.

BTYP unsigned int

Branch target identification bit.

CORE_NUM unsigned int

Core number in a multi processor.

DAIF unsigned int

Exception mask bits.

DIT unsigned int

Data Independent Timing bit.

EL unsigned int

Current Exception level field.

EXLOCK unsigned int

Exception return state lock bit.

IL unsigned int

Illegal Execution state bit.

NZCV unsigned int

Condition flags.

PACM unsigned int

PACM bit.

PAN unsigned int

Privileged Access Never state bit.

PM unsigned int

PMU exception mask bit.

PPEND unsigned int

PMU exception pending bit.

SM unsigned int

StreamingMode status bit.

SP unsigned int

Stack pointer register selection bit.

SS unsigned int

Software Step bit.

SSBS unsigned int

Speculative Store Bypass Safe bit.

TCO unsigned int

Tag Check Override bit.

TOUCHED unsigned int

Mask of the fields that have been touched since this trace has last been emitted. Note that touched does not necessarily mean the value changed. The index of the field is the bit index in the mask plus the index of this field plus one.

UAO unsigned int

User Access Override bit.

UINJ unsigned int

Undefined Instruction exceptions bit.

ZA unsigned int

ZA storage and ZT0 register enable bit.

nRW unsigned int

Current Execution state bit.

PSTATE_PACTIVE_OPMODE_TRANSITION

Trace pactive and Opmode mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Operating mode after the transition.

OLD_STATE enum

Operating mode before the transition.

PSTATE_PACTIVE_OPMODE_TRANSITION_DENIED

Trace the denied pactive and Opmode mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Operating mode after the transition.

OLD_STATE enum

Operating mode before the transition.

REASON string

Reason for the transition being denied.

PSTATE_PACTIVE_POWER_TRANSITION

Trace pactive and power mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Power mode after the transition.

OLD_STATE enum

Power mode before the transition.

PSTATE_PACTIVE_POWER_TRANSITION_DENIED

Trace the denied pactive and power mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Power mode after the transition.

OLD_STATE enum

Power mode before the transition.

REASON string

Reason for the transition being denied.

PSTATE_PACTIVE_THREAD_OPMODE_TRANSITION

Trace pactive and Opmode mode transitions for threads on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Operating mode after the transition.

OLD_STATE enum

Operating mode before the transition.

QUANTUM_START

Called at the start of every quantum.

RANGE_PRELOAD_REQUEST

Preload requests from RPRFM. Fields:

COUNT unsigned int

Number of blocks of data to access.

LENGTH unsigned int

Number of contiguous bytes to be accessed in each block.

OP unsigned int

Range prefetch operation .

REUSE unsigned int

Maximum number of bytes to be accessed before next RPRFM .

STRIDE signed int

Stride to advance after accessing LENGTH bytes.

VADDR unsigned int

Base virtual address of range prefetch request.

RUN_STATE

Run state transition. Fields:

INST_COUNT unsigned int

Ticks count at point of transition.

NEW enum

New run state.

OLD enum

Old run state.

SEMIHOSTING_CALL

Call of a semihost function occurred. Fields:

PC unsigned int

The program counter after the semihosting call.

SEMIHOSTING_PRECALL

About to call semihost. Fields:

EL2 bool

Use EL2 translation regime.

NS bool

Is Non-Secure.

REG_WIDTH unsigned int

The current register width in bytes.

SIGNAL

External signal state change. Fields:

SIGNAL enum

Signal that changed.

STATE bool

Signal asserted state.

SOFTWARE_STEP

Return the Debug Software Step state. Fields:

STATE enum

Software Step state.

SPSR

SPSR change. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

MODE enum

Which of the banked SPSR registers is written.

OLD_VALUE unsigned int

The old SPSR value.

VALUE unsigned int

The new SPSR value.

START_COMPILE

Compilation started. Fields:

VADDR unsigned int

Instruction where compilation begins.

STG_MULTIPLE_MICRO_OPS_ADDRESS_RANGE

Address range for MTE stores with multiple uops. Fields:

ADDR_END unsigned int

Final address.

ADDR_START unsigned int

Starting address.

STORE_MULTIPLE_REGS_ADDRESS_RANGE

Address range for stores with multiple registers. Fields:

ADDR_END unsigned int

Final address.

ADDR_START unsigned int

Starting address.

SUBOPTIMAL_LDST_RETIRED

suboptimal load/store. Fields:

ACCESS_SIZE unsigned int

Log2 of the access width in bytes used for alignment checking (1: half-word, 2: word ...).

OPTIMAL_ALIGN_SELECTOR unsigned int

Selector to decide the optimal alignment size (1: access width, 2: 4-bytes, 3: 8-bytes, ..., 12: 4-kbytes).

VADDR unsigned int

The virtual address of the access.

SYNC

Called for every synchronization, that is every quantum break. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The instruction count of this CPU.

LOCAL_QUANTUM unsigned int

The local quantum of this CPU.

LOCAL_TIME unsigned int

The local time of this CPU.

SYNC_REASON enum

Reason for sync.

SYSCALL

System call instruction executed. Fields:

IMM unsigned int

Immediate value of the system call instruction.

TYPE enum

System call type.

VADDR unsigned int

Instruction that caused the system call.

SYSREG_READ128

System Coprocessor 128 bit register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

REGNUM enum

Internal register number.

REGNUM_PHYS enum

Internal register number physically accessed (e.g. in presense of VHE redirection).

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_READ64

System Coprocessor register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REGNUM enum

Internal register number.

REGNUM_OPERAND enum

Register number corresponding to the instruction operand.

REGNUM_PHYS enum

Internal register number physically accessed (e.g. in presense of VHE redirection).

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_UPDATE128

Triggers when the system updates a register. Fields:

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG enum

Register number.

UNKNOWN unsigned int

Bits of the register which became unknown.

VALUE unsigned int

Value written to the register.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_UPDATE32

Triggers when the system updates a register. Fields:

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG enum

Register number.

UNKNOWN unsigned int

Bits of the register which became unknown.

VALUE unsigned int

Value written to the register.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_UPDATE64

Triggers when the system updates a register. Fields:

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG enum

Register number.

UNKNOWN unsigned int

Bits of the register which became unknown.

VALUE unsigned int

Value written to the register.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_WRITE128

System Coprocessor 128 bit register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

REGNUM enum

Internal register number.

REGNUM_PHYS enum

Internal register number physically accessed (e.g. in presense of VHE redirection).

UNDEF bool

The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_WRITE64

System Coprocessor register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REGNUM enum

Internal register number.

REGNUM_OPERAND enum

Register number corresponding to the instruction operand.

REGNUM_PHYS enum

Internal register number physically accessed (e.g. in presense of VHE redirection).

UNDEF bool

The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

UNALIGNED_LDST_RETIRE

Processor unaligned load/store. Fields:

ACCESS_SIZE unsigned int

Log2 of the access width in bytes used for alignment checking (1=>half-word, 2->word ...).

NUMBER_OF_BEATS unsigned int

Number of accesses (beats) in this burst (Total size in bytes is ACCESS_SIZE * NUMBER_OF_BEATS).

VADDR unsigned int

The virtual address of the access.

VFP_D_REGS

VFP/NEON D 64 bit register write. Fields:

ALIASING enum

Type of register aliasing used.

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a write to the corresponding bit in the register. For example, 0x1 would indicate a write to VALUE[0].

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

VFP_D_REGS_READ

VFP/NEON D 64 bit register read. Fields:

ID unsigned int

The register number.

VALUE unsigned int

The current value of the register.

VFP_Q_REGS

VFP/NEON Q 128 bit register write. Fields:

ALIASING enum

Type of register aliasing used.

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a write to the corresponding byte in the register. For example, 0x1 would indicate a write to VALUE[7:0].

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

VFP_Q_REGS_READ

VFP/NEON Q 128 bit register read. Fields:

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a read to the corresponding byte in the register. For example, 0x1 would indicate a read to VALUE[7:0].

VALUE unsigned int

The current value of the register.

VFP_SYS_REGS

Writes to the VFP/NEON units system registers. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ID enum

Which VFP system register is written.

MASK unsigned int

The mask indicating updated cumulative bits.

OLD_VALUE unsigned int

The register's old value overwritten.

VALUE unsigned int

The new value written to the VFP system register.

VFP_S_REGS

VFP/NEON S 32 bit register write. Fields:

ALIASING enum

Type of register aliasing used.

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a write to the corresponding bit in the register. For example, 0x1 would indicate a write to VALUE[0].

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

VFP_S_REGS_READ

VFP/NEON S 32 bit register read. Fields:

ID unsigned int

The register number.

VALUE unsigned int

The current value of the register.

VIRTUAL_TAG_ALLOCATED

Virtual tag information when allocated. Fields:

TAG_VADDR unsigned int

Virtual tag virtual address.

VADDR unsigned int

Data Virtual address.

VIRTUAL_TAG unsigned int

Virtual tag value.

WAYPOINT

Signals end of basic block execution. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ISET enum

Origin instruction set.

IS_COND bool

Indicates if this is a conditional waypoint.

PC unsigned int

Origin address (or 0 if unavailable).

TAKEN bool

Indicates if this waypoint was taken.

TARGET unsigned int

Destination address.

TARGET_ISET enum

Destination instruction set.

WFE_END

WFE ended. Fields:

INST_COUNT unsigned int

Ticks count when leaving WFE.

WFE_EVENT_REGISTER

WFE event register status: set/clear, reason. Fields:

INST_COUNT unsigned int

Ticks count.

REASON enum

Reason for set/clear. Only REASON==0 (Cleared by WFE) clears the bit, all other reasons set it.

WFE_IGNORED

WFE ignored. Fields:

EVENT bool

This WFE was ignored because the event register was set.

INST_COUNT unsigned int

Ticks count when ignoring WFE.

TRAPPED bool

This WFE was trapped.

WFE_START

WFE entered. Fields:

INST_COUNT unsigned int

Ticks count when entering WFE.

WFI_END

WFI ended. Fields:

INST_COUNT unsigned int

Ticks count when leaving WFI.

WFI_IGNORED

WFI ignored. Fields:

DISABLED bool

This WFI was ignored because WFI is disabled.

INST_COUNT unsigned int

Ticks count when ignoring WFI.

REASON enum

specifies reason why WFI trace was ignored.

TRAPPED bool

This WFI was trapped.

WFI_START

WFI entered. Fields:

INST_COUNT unsigned int

Ticks count when entering WFI.

WFI_WAKEUP

WFI wakeup. Fields:

INST_COUNT unsigned int

Ticks count when WFI wakeup occurred.

REASON enum

Reason for wakeup.

2.7 ARM_AEMv8M

This section describes the trace sources.

ASYNC_MEMORY_FAULT

Context ID Register write. Fields:

FAULT unsigned int

Fault status present in ESR.ISS encoding for Serror in ARCH64 and present in DFSR encoding for external abort format in ARCH32.

PADDR unsigned int

Physical Address (or 0 if unavailable).

ATOMIC_END_ACCESS

Bus trace access for atomics. Fields:

ACCESS_FAIL bool

Memory access failed.

ADDR unsigned int

The virtual address of the access.

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

COMPARE_VALUE unsigned int

Compare value for CAS.

LOAD_VALUE unsigned int

Loaded values.

NSDESC unsigned int

The security state of the access.

OPERAND_VALUE unsigned int

Operation's operand.

OPERATION enum

Operation type.

PADDR unsigned int

The physical address of the access.

PRIV bool

Is this a privileged access?.

ATOMIC_SLAVE_ACCESS

Information on the atomic operation at slave side. Fields:

ACCESS_RESULT unsigned int

Result of atomic access.

COMPARE_VALUE unsigned int

Value to be compared with memory by CAS.

LOAD_VALUE unsigned int

Value to be loaded from memory as a result of atomic operation.

MANAGER unsigned int

ID of manager which initiated the transaction.

NS bool

Transaction is Non-Secure.

OPERAND_VALUE unsigned int

Incoming operand value provided by core.

OPERATION enum

Type of atomic operation.

PADDR unsigned int

Full physical address of TCM.

STORE_VALUE unsigned int

Value to be stored to memory as a result of atomic operation.

ATOMIC_START_ACCESS

Bus trace access for atomics. Fields:

ADDR unsigned int

The virtual address of the access.

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

COMPARE_VALUE unsigned int

Compare value for CAS.

NSDESC unsigned int

The security state of the access.

OPERAND_VALUE unsigned int

Operation's operand.

OPERATION enum

Operation type.

PADDR unsigned int

The physical address of the access.

PRIV bool

Is this a privileged access?.

ArchMsg.Error.exit_code

DISPLAY exit code: %{EXITCODE} component kind: %{KIND} reason: %{REASON}. Fields:

EXITCODE signed int

Exit code to be returned.

KIND string

Component kind that invoked the exit code trace.

REASON string

Reason why the exit code trace was invoked (optional).

ArchMsg.Warning.Unpredictably Indexed PM Event Register

TAGS **UNPREDICTABLE** DISPLAY Event %{IsDirect:(Selector value|Register)} %{SEL} is >= %{N}.

Fields:

IsDirect unsigned int

Direct Access rather than indirect.

N unsigned int

Number of Accessible Registers.

SEL unsigned int

Selector.

ArchMsg.Warning.cache_contents_unknown

A cache has been activated without being invalidated and may contain **UNKNOWN** content. DISPLAY %{SIDE:(D|I)}-Cache was enabled in %{REGIME:(SECURE|NON_SECURE|HYP)} regime but was not invalidated since power-on: cache lines could contain **UNKNOWN** content %{SIDE:(Note: D-side caches must invalidate SECURE lines even if the cache is only used by NON_SECURE code in order to ensure that unexpected dirty lines tagged SECURE are not naturally evicted)}. Fields:

REGIME unsigned int

Current translation regime.

SIDE unsigned int

0 d-side cache or unified cache, 1 i-side.

ArchMsg.Warning.dap_csw_bad_size

A write to CM3DAP CSW has an invalid size field. DISPLAY Write %{DATA} to CM3DAP CSW has invalid size field. Fields:

DATA unsigned int

bits[2:0] are size.

ArchMsg.Warning.dcimvac_matches_watchpoint

TAGS **UNPREDICTABLE** DISPLAY Watchpoints matching an AArch32 DCIMVA are implementation defined.

ArchMsg.Warning.decode_fieldmismatch

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with different fields %{FIELD1} and %{FIELD2}. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_frachitsnegative

TAGS **UNPREDICTABLE** DISPLAY VCVT Instruction is unpredictable with a negative number of fraction bits.

ArchMsg.Warning.decode_initblock

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable in an IT block.

ArchMsg.Warning.decode_initblocknotlast

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable in an IT block when not the last.

ArchMsg.Warning.decode_invalid_condition

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable when the condition is not AL.

ArchMsg.Warning.decode_invalidfieldcombination

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with the values in %{FIELD1} and %{FIELD2}. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_invalidvalue

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with the value in %{FIELD}. Fields:

FIELD enum

Field name.

ArchMsg.Warning.decode_mem_hint_unallocated

TAGS **UNPREDICTABLE** DISPLAY There is no memory hint allocated to this bit pattern.

ArchMsg.Warning.decode_registermatch

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with %{FIELD1} and %{FIELD2} as the same register. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_registermismatch

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with %{FIELD1} and %{FIELD2} as different registers. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_registeroutofrange

TAGS **UNPREDICTABLE** DISPLAY Floating-point instruction is unpredictable indexing beyond the end of the register bank.

ArchMsg.Warning.decode_sbzsbo

TAGS **UNPREDICTABLE** DISPLAY Reserved bits in the instruction are not canonical.

ArchMsg.Warning.decode_transactiontoobig

TAGS **UNPREDICTABLE** DISPLAY Attempt to perform a memory transaction of more than 128 bytes.

ArchMsg.Warning.decode_unpred_other

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable.

ArchMsg.Warning.decode_unpreduseofpc

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable when %{FIELD} is PC. Fields:

FIELD enum

Field name.

ArchMsg.Warning.decode_unpreduseofr13

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable when %{FIELD} is R13. Fields:

FIELD enum

Field name.

ArchMsg.Warning.decode_unpreduseofr13inreglist

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with R13 in register-list.

ArchMsg.Warning.decode_useofd16orhigher

TAGS **UNPREDICTABLE** DISPLAY Attempt to use register D16 or higher where it is not allowed.

ArchMsg.Warning.decode_writebackandbaseinlist

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with write back when the base register is in the transfer list.

ArchMsg.Warning.decode_zeroregistersinlist

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable when no registers are to be transferred.

ArchMsg.Warning.recursive_branch

An instruction is performing a branch that targets the same instruction. If this was intended then a WFI might be more effective DISPLAY recursive branch detected at PC=%{PC}. PRIMARY KEY PC. Fields:

PC unsigned int

Address of instruction causing the branch.

ArchMsg.Warning.recursive_exception

An instruction has generated an exception that targets the same instruction. DISPLAY recursive %{TYPE} exception detected at PC=%{PC} (CPSR=%{CPSR}). PRIMARY KEY PC. Fields:

CPSR unsigned int

Processor state.

PC unsigned int

Address of instruction causing the exception.

TYPE enum

Type of exception.

ArchMsg.Warning.reentrant_vector_catch

TAGS **UNPREDICTABLE** DISPLAY Vector catch debug event on Prefetch or Data abort vector %**{ADDR}**. Fields:

ADDR unsigned int

vector address.

ArchMsg.Warning.reserved_it_state

TAGS **UNPREDICTABLE** DISPLAY Execution with reserved IT state %**{ITSTATE}**. Fields:

ITSTATE unsigned int

reserved IT state.

ArchMsg.Warning.secure_vector_fetch_from_nonsecure

DISPLAY Secure vector table is being fetched from Non-secure memory.

ArchMsg.Warning.sg_in_it_block

Arm recommends that software does not place SG (security gate instruction) inside an IT instruction block.

ArchMsg.Warning.unpred_FAR_on_LRCPC3_non_atomic_mem_access

TAGS **UNPREDICTABLE** Unpredictable FAR/FSTATUS on LRCPC3 pair instructions resulting in two separate memory accesses
DISPLAY Unpredictable FAR on LRCPC3 non atomic memory access at %**{ADDR1}** with FSTATUS %**{FSTATUS1}** or %**{ADDR2}** with FSTATUS %**{FSTATUS2}**. Fields:

ADDR1 unsigned int

Possible FAR address 1.

ADDR2 unsigned int

Possible FAR address 2.

FSTATUS1 unsigned int

Possible fault status value 1.

FSTATUS2 unsigned int

Possible fault status value 2.

ArchMsg.Warning.unpred_async_tag_check_fail_on_sve_speculative_load_do_update_ffr

Unpredictable set to FALSE of the FFR element associated to SVE speculative load with an async tag check fail TAGS **UNPREDICTABLE**.

ArchMsg.Warning.unpredictable_a32_breakpoint

TAGS **UNPREDICTABLE** DISPLAY Unpredictable breakpoint on a A32 instruction at %{ADDR1} Offset %{OFFSET} BAS %{BAS} for a %{IS_ADDRESS_MISMATCH_BREAKPOINT:(Mismatch|Match)} Breakpoint number %{N:d} Breakpoint address %{ADDR2}. Fields:

ADDR1 unsigned int

Address.

ADDR2 unsigned int

Breakpoint address.

BAS unsigned int

BAS field.

IS_ADDRESS_MISMATCH_BREAKPOINT bool

Is a Mismatch breakpoint, not a Match breakpoint.

N unsigned int

Breakpoint number.

OFFSET unsigned int

Always has a value of zero.

ArchMsg.Warning.unpredictable_banked_register_access

TAGS **UNPREDICTABLE** DISPLAY Access to banked register (R=%{R:d} SYSm=%{SYSM:2x}) is unpredictable. Fields:

R unsigned int

R bit.

SYSM unsigned int

SYSm.

ArchMsg.Warning.unpredictable_ccfail_undef

TAGS **UNPREDICTABLE** DISPLAY An instruction %{OPCODE} at VA=%{VADDR} failed its condition codes check but would otherwise have caused a non-data-dependent trap or undefined exception. Behaviour is implementation-defined choice of exception or nop. Fields:

OPCODE unsigned int

opcode of the instruction.

VADDR unsigned int

Virtual address of the instruction being executed.

ArchMsg.Warning.unpredictable_ici_bits

TAGS **UNPREDICTABLE** DISPLAY ICI bits do not apply or are architecturally unpredictable for instruction.

ArchMsg.Warning.unpredictable_t32_breakpoint

TAGS **UNPREDICTABLE** DISPLAY Unpredictable breakpoint on a T32 instruction at %{ADDR1} Offset %{OFFSET} BAS %{BAS} for a %{IS_ADDRESS_MISMATCH_BREAKPOINT:(Mismatch|Match)} Breakpoint number %{N:d} Breakpoint address %{ADDR2}. Fields:

ADDR1 unsigned int

Address.

ADDR2 unsigned int

Breakpoint address.

BAS unsigned int

BAS field.

IS_ADDRESS_MISMATCH_BREAKPOINT bool

Is a Mismatch breakpoint, not a Match breakpoint.

N unsigned int

Breakpoint number.

OFFSET unsigned int

Indicates which halfword of a 32-bit Thumb instruction the breakpoint is set on. 0 for the first halfword or if the instruction is 16 bits wide, or 1 for the second halfword.

ArchMsg.Warning.unpredictable_t32_vector_catch

TAGS **UNPREDICTABLE** DISPLAY Unpredictable Vector catch exception on a T32 instruction at %{ADDR} Offset %{OFFSET}. Fields:

ADDR unsigned int

Address.

OFFSET unsigned int

Indicates which halfword of a 32-bit Thumb instruction the vector catch exception is set on. 0 for the first halfword or if the instruction is 16 bits wide, or 1 for the second halfword.

ArchMsg.Warning.unpredictable_watchpoint_far

TAGS **UNPREDICTABLE** DISPLAY The value for FAR/EDWAR for hitting this watchpoint may be anywhere in the range %{LOWER_BOUND} to %{UPPER_BOUND}. Fields:

LOWER_BOUND unsigned int

the lowest address accessed by the instruction that triggered the watchpoint.

UPPER_BOUND unsigned int

the highest watchpointed address accessed by that instruction.

ArchMsg.Warning.warning_AdvSIMDEExpandImmUnexpectedZero

TAGS **UNPREDICTABLE** DISPLAY AdvSIMDEExpandImm may treat this immediate value as **UNPREDICTABLE**.

ArchMsg.Warning.warning_ConditionalSMC

TAGS **UNPREDICTABLE** DISPLAY SMC instruction has **UNPREDICTABLE** effects when conditional and when combined with traps.

ArchMsg.Warning.warning_access_crosses_page_boundary

TAGS **UNPREDICTABLE** DISPLAY Access crosses page boundary.

ArchMsg.Warning.warning_access_crosses_page_boundary_has_fault

TAGS **UNPREDICTABLE** DISPLAY Access that crosses page boundary has fault(s).

ArchMsg.Warning.warning_access_crosses_page_boundary_has_fault_on_both_pages

TAGS **UNPREDICTABLE** DISPLAY Access that crosses page boundary has a fault on each page.

Fields:

ADDR1 unsigned int

FAR address on first page.

ADDR2 unsigned int

FAR address on second page.

FSTATUS1 unsigned int

Fault status on first page.

FSTATUS2 unsigned int

Fault status on second page.

ArchMsg.Warning.warning_access_crosses_page_boundary_spanning_different_memory

TAGS **UNPREDICTABLE** DISPLAY Access at address %{ADDR} crosses page from MEM_TYPE %{MEMTYPE_PAGE1} to %{MEMTYPE_PAGE2}. Fields:

ADDR unsigned int

address of access crossing page.

MEMTYPE_PAGE1 signed int

First Page's memory type (0-Normal, 1-Non-Normal).

MEMTYPE_PAGE2 signed int

Second Page's memory type (0-Normal, 1-Non-Normal).

ArchMsg.Warning.warning_access_wraps_around_memory

TAGS **UNPREDICTABLE** DISPLAY Access wraps around memory in %{WIDTH:d}bit mode.

Fields:

WIDTH unsigned int

Address width.

ArchMsg.Warning.warning_bcr_linking_status

TAGS **UNPREDICTABLE** DISPLAY DBGBCR.{BT2,BT} combination gives unpredictable behaviour.

ArchMsg.Warning.warning_bcr_mask_reserved

TAGS **UNPREDICTABLE** DISPLAY DBGBCR_ELn.MASK is set to a reserved value.

ArchMsg.Warning.warning_bvr_masked_bit_not_zero

TAGS **UNPREDICTABLE** DISPLAY DBGBVRn_EL1.VA contains a bit masked by DBGBCRn_EL1.MASK that is not zero.

ArchMsg.Warning.warning_ccsidr_unimplemented_level

TAGS **UNPREDICTABLE** DISPLAY CCSIDR read while CSSELR %{CSSELR:x} points to an unimplemented cache level. Fields:

CSSELR unsigned int

current effective value of CSSELR.

ArchMsg.Warning.warning_change_to_ns_when_tge_set

TAGS **UNPREDICTABLE** DISPLAY Attempting to change NS to 1 when HCR.TGE = 1.

ArchMsg.Warning.warning_contiguous_bit_check_abort

TAGS **UNPREDICTABLE** DISPLAY An abort occurred whilst attempting to check TLB entry is contiguous at %{ENTRY_ADDR}. Fields:

ENTRY_ADDR unsigned int

address of conflicting TLB entry.

ArchMsg.Warning.warning_contiguous_bit_error

TAGS **UNPREDICTABLE** DISPLAY TLB entry at %{ENTRY_ADDR} was not contiguous with entry at %{CONTIG_ADDR} Contents are %{ENTRY_DATA} but expected %{CONTIG_DATA}. Fields:

CONTIG_ADDR unsigned int

address of the TLB first read, with which this entry is expected to be contiguous.

CONTIG_DATA unsigned int

expected contents based on the entry at CONTIG_ADDR.

ENTRY_ADDR unsigned int

address of conflicting TLB entry.

ENTRY_DATA unsigned int

contents of conflicting TLB entry.

ArchMsg.Warning.warning_contiguous_bit_error_D128

TAGS **UNPREDICTABLE** DISPLAY TLB entry at %{ENTRY_ADDR} was not contiguous with entry at %{CONTIG_ADDR} Contents are %{ENTRY_DATA_HI} %{ENTRY_DATA_LO} but expected %{CONTIG_DATA}. Fields:

CONTIG_ADDR unsigned int

address of the TLB first read, with which this entry is expected to be contiguous.

CONTIG_DATA unsigned int

expected contents based on the entry at CONTIG_ADDR.

ENTRY_ADDR unsigned int

address of conflicting TLB entry.

ENTRY_DATA_HI unsigned int

Higher bit contents of conflicting TLB entry.

ENTRY_DATA_LO unsigned int

Lower bit contents of conflicting TLB entry.

ArchMsg.Warning.warning_cp10_cp11_mismatch

TAGS **UNPREDICTABLE** DISPLAY Unpredictable configuration of CP10 and CP11 controls at EL
%{EL:d}. Fields:

EL unsigned int

EL owning the control.

ArchMsg.Warning.warning_cp10_cp11_reserved_value

TAGS **UNPREDICTABLE** DISPLAY Unpredictable configuration of CPACR CP10 (%{CP10}) or
CP11 (%{CP11}). Fields:

CP10 unsigned int

CP10 access.

CP11 unsigned int

CP11 access.

ArchMsg.Warning.warning_csselr_level_out_of_range

TAGS **UNPREDICTABLE** DISPLAY CSSELR.Level out of range, written value %{LEVEL} with only
%{IMPLEMENTED:d} levels implemented. Fields:

IMPLEMENTED unsigned int

number of cache levels implemented.

LEVEL unsigned int

written value of CSSELR.Level.

TYPE enum

The type of cache selected by CSSELR.InD and TnD.

ArchMsg.Warning.warning_debug_flow_control_bits_not_obeyed

TAGS **UNPREDICTABLE** DISPLAY Flow control bits for %{REGISTER} were not obeyed, causing
%{ERROR}. Fields:

ERROR enum

type of error resulting.

REGISTER enum

register under consideration.

ArchMsg.Warning.warning_debug_register_access_during_reset

TAGS **UNPREDICTABLE** DISPLAY Unpredictable %{IS_WRITE:(read|write)} access to debug
register offset %{OFFSET} during reset. Fields:

IS_WRITE unsigned int

Write Not Read.

OFFSET unsigned int

Register Offset.

ArchMsg.Warning.warning_decode_cps_inconsistent_fields

TAGS **UNPREDICTABLE** DISPLAY The mode setting fields in the CPS instruction are inconsistent.

ArchMsg.Warning.warning_decode_invalid_state

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable in current exception-level/security state.

ArchMsg.Warning.warning_default_cacheable_mmu_on

TAGS **UNPREDICTABLE** DISPLAY Default cacheable is enabled (HCR.DC = 1) while MMU is enabled.

ArchMsg.Warning.warning_default_cacheable_vmmu_off

TAGS **UNPREDICTABLE** DISPLAY Default cacheable is enabled (HCR.DC = 1) while HCR.VM = 0.

ArchMsg.Warning.warning_deprecated_wvr_bit_2_set

TAGS **UNPREDICTABLE** DISPLAY DBGWVR%{N:d}_EL1=%{WVR:h} has bit 2 set, deprecated in ARMv8. Fields:

N unsigned int
index.

WVR unsigned int
DBGWVR.

ArchMsg.Warning.warning_eret_while_software_step_active_pending

Missing ISB between setting MDSCR_EL1.SS and ERET TAGS **UNPREDICTABLE**.

ArchMsg.Warning.warning_exclusive_to_non_normal

TAGS **UNPREDICTABLE** DISPLAY Exclusive Access to Non-Normal Memory %{ADDR}. Fields:

ADDR unsigned int
Address.

ArchMsg.Warning.warning_exclusive_to_non_writeback

TAGS **UNPREDICTABLE** DISPLAY Exclusive Access to Non-Writeback-Cacheable Memory %{ADDR}. Fields:

ADDR unsigned int
Address.

ArchMsg.Warning.warning_execute_from_device_memory

TAGS **UNPREDICTABLE** DISPLAY An attempt was made to execute from Device memory.

ArchMsg.Warning.warning_illegal_cpsr_mode

TAGS **UNPREDICTABLE** DISPLAY Writing an illegal or unimplement mode (%{NEW_MODE:h}) to CPSR was unpredictable. Fields:

NEW_MODE unsigned int
New value of CPSR.M.

ArchMsg.Warning.warning_illegal_srs_mode

TAGS **UNPREDICTABLE** DISPLAY Illegal or **UNPREDICTABLE** mode (%{MODE:h}) used for SRS instruction. Fields:

MODE unsigned int

mode for Banked SP.

ArchMsg.Warning.warning_implementation_defined_read_debug_register_in_SCS

Read Access to Debug registers from the processor is IMPLEMENTATION DEFINED TAGS IMP_DEF. Fields:

PPB_OFFSET unsigned int

debug register offset.

ArchMsg.Warning.warning_implementation_defined_sequential_security_transitions_supported

The behavior of sequential instruction fetches that cross from non-secure to secure memory and contain an SG instruction is **CONSTRAINED UNPREDICTABLE** TAGS **UNPREDICTABLE**.

ArchMsg.Warning.warning_implementation_defined_stack_limit_check_supported

It is **IMPLEMENTATION DEFINED** whether stack pointer limit checking is performed for this instruction TAGS IMP_DEF. Fields:

ADDRESS unsigned int

Address of instruction.

OPCODE unsigned int

instruction opcode.

ArchMsg.Warning.warning_implementation_defined_taggednonsharedmemorytype

TAGS IMP_DEF DISPLAY Memory attribute has Tagged, non shared memory type; it is **IMPLEMENTATION DEFINED** whether the memory location is treated as Tagged or non-Tagged.

ArchMsg.Warning.warning_implementation_defined_write_debug_register_in_SCS

Write Access to Debug registers from the processor is IMPLEMENTATION DEFINED TAGS IMP_DEF. Fields:

DATA unsigned int

data attempted to be written.

PPB_OFFSET unsigned int

debug register offset.

ArchMsg.Warning.warning_invalid_tcr_granule

TAGS **UNPREDICTABLE** DISPLAY TCR.TG{%TG_ID:(0|1)} bits have been set to a granule size {%REQUEST:d}K not implemented - using {%SUBSTITUTE:d}K. Fields:

REQUEST signed int

page size requested (or 0 for reserved).

SUBSTITUTE signed int

best guess available page size.

TG_ID bool

bits TG0 or TG1.

ArchMsg.Warning.warning_load_multiple_user_registers_from_user_mode

TAGS UNPREDICTABLE DISPLAY An LDM(user registers) instruction executed from user mode.

ArchMsg.Warning.warning_mte_in_mem_attr_not_supported

TAGS UNPREDICTABLE DISPLAY Memory attribute %{ATTR} has Tagged memory type but Armv8.5-MTE is not implemented. Fields:

ATTR unsigned int

Memory attribute.

ArchMsg.Warning.warning_reserved_breakpoint_state_match

TAGS UNPREDICTABLE DISPLAY %{IS_BREAKPOINT:(Watchpoint|Breakpoint)} programmed with a reserved combination of HMC, SSC and %{IS_BREAKPOINT:(PAC|PMC)}. Fields:

IS_BREAKPOINT unsigned int

Is a breakpoint, not a watchpoint.

ArchMsg.Warning.warning_shareability

TAGS UNPREDICTABLE DISPLAY Unpredictable: combination of the 1st and 2nd stages of translation is Normal Inner Non-Cacheable, Outer Non-Cacheable.

ArchMsg.Warning.warning_software_step_set_while_enabled

TAGS UNPREDICTABLE DISPLAY MDSCR_EL1.SS set to 1 while software step debug exceptions are enabled.

ArchMsg.Warning.warning_thumb_instruction_wraps_around_memory

TAGS UNPREDICTABLE DISPLAY Thumb instruction wraps around memory.

ArchMsg.Warning.warning_ttbr_sbz_bits_are_not_zero

TAGS UNPREDICTABLE DISPLAY %{TTBR}[%{MSB:d}:0] should be zero, but are not. Fields:

MSB unsigned int

x-1.

TTBR enum

which TTBR register.

ArchMsg.Warning.warning_unaligned_address_dbgdtrrx_write

TAGS UNPREDICTABLE DISPLAY Unaligned address %{ADDR} in X0/R0 when DBGDTRRX written in memory access mode. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unaligned_address_dbgdtrtx_read

TAGS UNPREDICTABLE DISPLAY Unaligned address %{ADDR} in X0/R0 when DBGDTRTX read in memory access mode. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unaligned_to_device

TAGS **UNPREDICTABLE** DISPLAY Unaligned Access to Device Memory %{ADDR}. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unaligned_to_strongly_ordered

TAGS **UNPREDICTABLE** DISPLAY Unaligned Access to Strongly Ordered Memory %{ADDR}. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unknown_sau_rnr

TAGS **UNPREDICTABLE** DISPLAY SAU_RNR was set to an unsupported value.

ArchMsg.Warning.warning_unpredictable_AIRCR_PRIS_and_BFHFNMINs

TAGS **UNPREDICTABLE** DISPLAY The effect of setting both AIRCR.BFHFNMINs and AIRCR.PRIS to 1 is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_AIRCR_VECTCLRACTIVE_when_not_in_debug

TAGS **UNPREDICTABLE** DISPLAY The effect of writing a 1 to AIRCR.VECTCLRACTIVE if the processor is not halted in Debug state is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_AIRCR_VECTRESET_and_SYSRESETREQ

TAGS **UNPREDICTABLE** DISPLAY When the processor is halted in Debug state, if a write to the register writes a 1 to both VECTRESET and SYSRESETREQ, the behavior is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_AIRCR_VECTRESET_when_not_in_debug

TAGS **UNPREDICTABLE** DISPLAY The effect of writing a 1 to AIRCR.VECTRESET if the processor is not halted in Debug state is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_AIRCR_incorrect_VKEY

TAGS **UNPREDICTABLE** DISPLAY The value 0x05FA must be written to AIRCR.VKEY, otherwise the register write is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_EXC_RETURN_Reserved_Bit

EXC_RETURN[23:7] are reserved with the special condition that all bits should be written as one. Values other than all 1s are **UNPREDICTABLE**. TAGS **UNPREDICTABLE**. Fields:

EXC_RETURN unsigned int

Special PC value.

SBO_MASK unsigned int

Which bits Should Be One to not UNPRED.

ArchMsg.Warning.warning_unpredictable_PMEVTYPEPn_ELO_VS_reserved_value

TAGS **UNPREDICTABLE** DISPLAY Unpredictable reserved value of PMEVTYPEn<n>_ELO.VS.

ArchMsg.Warning.warning_unpredictable_change_to_DEMCR_MON_STEP_at_insufficient_priority

TAGS UNPREDICTABLE DISPLAY The effect of changing DEMCR.MON_STEP at an execution priority that is lower than the priority of the DebugMonitor exception is **CONSTRAINED UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_change_to_priority_of_active_exception

TAGS UNPREDICTABLE DISPLAY Changing the priority of an active exception is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_clear_lspact

TAGS UNPREDICTABLE DISPLAY **UNPREDICTABLE** state on context switch. Fields:

DRVEXC_TAKEN unsigned int

Derived exception taken, not pended.

REACHED_LAST_STORE unsigned int

Only the last faulted.

ArchMsg.Warning.warning_unpredictable_dbg_exit_unaligned_dlr

TAGS UNPREDICTABLE DISPLAY Debug exit to AArch32 with DLR[0] set to 1. PC[0] can be set to 0 or to DLR[0].

ArchMsg.Warning.warning_unpredictable_exception_catch

TAGS UNPREDICTABLE DISPLAY Generation of an exception catch event is unpredictable.

ArchMsg.Warning.warning_unpredictable_exception_return_inconsistent_state

TAGS UNPREDICTABLE DISPLAY state on exception return is unpredictable.

ArchMsg.Warning.warning_unpredictable_exception_return_instruction

TAGS UNPREDICTABLE DISPLAY exception return instruction is unpredictable.

ArchMsg.Warning.warning_unpredictable_in_debug_state

TAGS UNPREDICTABLE DISPLAY Instruction is **UNPREDICTABLE** when executed in debug state.

ArchMsg.Warning.warning_unpredictable_mair_encoding

TAGS UNPREDICTABLE DISPLAY Memory attribute %{ATTR} has invalid MAIR encoding; this may lead to unpredictable results. Fields:

ATTR unsigned int

Memory attribute.

ArchMsg.Warning.warning_unpredictable_pmu_counter_access

TAGS UNPREDICTABLE DISPLAY Access to PMU counters from non-secure EL0 or EL1 is **UNPREDICTABLE** with MDCR_EL2.HPMN set to 0.

ArchMsg.Warning.warning_unpredictable_prioritization_breakpoint_match_vector_catch

TAGS UNPREDICTABLE DISPLAY Two events with the same priority occurred at the same instruction: Address Matching Vector Catch debug event, and Breakpoint debug event. It is constrained unpredictable which is taken.

ArchMsg.Warning.warning_unpredictable_stack_selection

TAGS UNPREDICTABLE DISPLAY **UNPREDICTABLE** state on context switch.

ArchMsg.Warning.warning_unpredictable_tsize_out_of_range

TAGS **UNPREDICTABLE** DISPLAY **%{TCR}.T%{TTBR:d}SZ (%{TSIZE:d} bits)** is out of range. Must be between 25 and **%{TSIZE_MAX:d}**. Fields:

TCR enum

Which TCR.

TSIZE signed int

Value of TSize.

TSIZE_MAX signed int

The maximum allowed value.

TTBR unsigned int

Which TTBR region.

ArchMsg.Warning.warning_unpredictable_unaligned_pc_as_base_register

TAGS **UNPREDICTABLE** DISPLAY The PC value must be word-aligned, otherwise the behavior of the instruction is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_unaligned_pop_stack

TAGS **UNPREDICTABLE** DISPLAY **CONSTRAINED UNPREDICTABLE** unaligned stack frame **%{FRAME_ADDR:x}** in exception-return. Fields:

FRAME_ADDR unsigned int

unaligned stack frame address.

ArchMsg.Warning.warning_unpredictable_unaligned_to_non_writeback

TAGS **UNPREDICTABLE** DISPLAY Unaligned Access to Non-Writeback-Cacheable Memory.

ArchMsg.Warning.warning_unpredictable_vmsa_memattrib

TAGS **UNPREDICTABLE** DISPLAY Unpredictable vmsa memory attribute with texcb value programmed to **%{TEXCB}** when tex remap is **%{TEX_REMAP}**. Fields:

TEXCB unsigned int

texcb value.

TEX_REMAP bool

use tex remap.

ArchMsg.Warning.warning_unpredictable_vtcr_t0sz_sl0

TAGS **UNPREDICTABLE** DISPLAY The combination of VTCR.TOSZ=**%{TOSZ:d}** and VTCR.SL0=**%{SL0:d}** is unpredictable. Fields:

SL0 signed int

VTCR.SL0, starting level for stage 2 translation table walks.

TOSZ signed int

VTCR.TOSZ, required input address range.

ArchMsg.Warning.warning_unpredictable_write_DHCSR

TAGS **UNPREDICTABLE** DISPLAY **UNPREDICTABLE** set of bit-changes in DHCSR. Fields:

NEW unsigned int

data attempted to be written.

OLD unsigned int

previous value.

ArchMsg.Warning.warning_unpredictable_write_SHCSR

TAGS UNPREDICTABLE DISPLAY UNPREDICTABLE setting a pending bit to 1 for an exception with priority \geq execution priority.

ArchMsg.Warning.warning_unsupported_access_to_memory_mapped_register

TAGS UNPREDICTABLE DISPLAY Access to memory mapped register is unsupported.

ArchMsg.Warning.warning_user_jmcr_access

TAGS UNPREDICTABLE DISPLAY User %{WRITE:(write|read)} access to JMCR is unpredictable. Fields:

WRITE unsigned int

access is write.

ArchMsg.Warning.warning_wcr_mask_and_bas

TAGS UNPREDICTABLE DISPLAY DBGWCR_EL1.MASK is non-zero and BAS is not 0xff.

ArchMsg.Warning.warning_wcr_mask_reserved

TAGS UNPREDICTABLE DISPLAY DBGWCR_EL1.MASK is set to a reserved value.

ArchMsg.Warning.warning_wcr_non_configuous_bas

TAGS UNPREDICTABLE DISPLAY DBGWCR_EL1.BAS has non-contiguous set of ones.

ArchMsg.Warning.warning_wvr_masked_bit_not_zero

TAGS UNPREDICTABLE DISPLAY DBGWVRn_EL1.VA contains a bit masked by DBGWCRn_EL1.MASK that is not zero.

ArchMsg.warning_unpred_system_register_access

DISPLAY Unpredictable %{IS_WRITE:(read|write)} access to system register offset %{OFFSET}. Fields:

IS_WRITE unsigned int

Write Not Read.

OFFSET unsigned int

Register Offset.

BRANCH_MISPREDICT

Simulating branch mispredict. Fields:

PC unsigned int

Origin address (or 0 if unavailable).

BRA_DIR

Direct branches, to immediate address. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISET enum

The instruction set of the branch instruction.

IS_COND bool

Indicates if this is a conditional branch.

IS_HINTED bool

Whether the conditional instruction is hinted.

IS_ISB bool

Whether the instruction is ISB.

IS_LINK bool

Indicates if this is a branch with link.

PC unsigned int

The address of the branch instruction.

TARGET_ISET enum

The instruction set after the branch.

TARGET_PC unsigned int

The address the instruction branches to.

BRA_INDIR

Indirect branches, perhaps to a register. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISET enum

The instruction set of the branch instruction.

IS_COND bool

Indicates if this is a conditional branch.

IS_HINTED bool

Whether the conditional instruction is hinted.

IS_ISB bool

Whether the instruction is ISB.

IS_LINK bool

Indicates if this is a branch with link.

PC unsigned int

The address of the branch instruction.

TARGET_ISET enum

The instruction set after the branch.

TARGET_PC unsigned int

The address the instruction branches to.

BRA_INDIR_CCFAIL

Indirect branches fail, perhaps to a register.

BRA_RET_CCFAIL

Branch return fail, perhaps to a register.

CACHE_MAINTENANCE_OP

Cache Maintenance Operation. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

DATA unsigned int

Specified MVA or Set/way.

FUNCTION enum

Clean / Invalidate.

SCOPE enum

Affected region.

SIDE enum

Inst / Data.

CCFAIL

Conditional instruction condition check fail. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CCFAIL_UNC

Unconditional instruction condition check fail. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CCPASS

Conditional instruction condition check pass. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CCPASS_UNC

Unconditional instruction condition check pass. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CODE_CACHE_MAINT

Code cache maintenance. Fields:

END_INCL unsigned int

Inclusive end address for VA or PA requests.

NS bool

Non-secure world for PA.

START unsigned int

Start address for VA or PA requests.

TYPE enum

Request type.

COMPILE_BLOCK_END

Last instruction of basic block translated. Fields:

VADDR unsigned int

Address of next instruction after this basic block.

COMPILE_BLOCK_START

First instruction of basic block translated. Fields:

VADDR unsigned int

Address of first instruction in basic block.

COMPILE_INST

ARM instruction compiled. Fields:

DISASS string

Disassembly of the instruction.

ISET enum

The instruction set of this instruction.

ITSTATE unsigned int

The ITSTATE current for the instruction.

OPCODE unsigned int

The opcode of the instruction.

PC unsigned int

The address of the instruction.

SIZE unsigned int

The size of the instruction in bytes.

CONTEXTIDR

Context ID Register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

NS bool

Secure or nonsecure banked register is accessed.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The new value written.

CONTEXT_SYNC

Called for every context synchronization event. Fields:

ARCH_CSE bool

Architectural Context Synchronization Event.

PC unsigned int

The address of the synchronization instruction.

SYSREG_SYNC bool

System register synchronization event.

CORE_ENDIAN

Core BE8 Big-Endian state changed. Fields:

BE8 bool

Core BE8 Big-Endian state.

CORE_INFO

Static processor attributes. Only triggered by a call to DumpState(). Fields:

ARCH_PROFILE enum

The architecture profile of the core.

CLUSTER_ID unsigned int

The cluster ID of this processor.

CORE_NUM unsigned int

The number of this core in an MP processor.

DETAILED_CORE_FEATURES enum

Detailed core features as a bitfield. Enum values are bitmasks to extract the info from the field.

FPU_VERSION enum

The VFP version implemented by the core.

MEM_ARCH enum

The memory architecture of the core.

MIDR_EL1 unsigned int

MIDR_EL1 value.

NUM_CORES unsigned int

The number of cores in this MP processor.

QUANTUM_SIZE unsigned int

The default quantum size of the core (in ticks).

SECURITY_FEATURES bool

Does the core have security features?.

CORE_LOADS

Processor load accesses. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ACQREL enum

Is this an acquire/release.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate

(Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DATA unsigned int

The data read or written.

ELEMENT_SIZE unsigned int

Width of each element.

ENDIANNESS enum

Byte ordering of this access.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

LOCK enum

Normal, exclusive or locked access.

MECID unsigned int

Memory Encryption Context ID, if applicable.

MECID2 unsigned int

Memory Encryption Context ID of the second page, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PAGE_SIZE unsigned int

Page size as $\log_2(\text{size in bytes})$.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK, 2=Exclusive Failed, 3=Unsupported.

SIZE unsigned int

Width of the access in bytes. Only required if DATA is not traced.

TRANS bool

Is this a translated access.

VADDR unsigned int

The virtual address of the access.

CORE_REGS

Changes of the core registers R0 to R14. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number, 0 to 14.

OLD_VALUE unsigned int

The old value overwritten.

PHYS_ID enum

The physical register accessed.

VALUE unsigned int

The new value written to the register.

CORE_STALLED_TRANSITION

Detect when a core is prevented from progressing PChannel states. Fields:

CURRENT_STATE enum

Current power mode of core.

REASON string

The reason a transition was stalled, and what must be done to unblock it.

TARGET_STATE enum

Desired power mode of core after transition.

CORE_STORES

Processor store accesses. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ACQREL enum

Is this an acquire/release.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes

([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DATA unsigned int

The data read or written.

ELEMENT_SIZE unsigned int

Width of each element.

ENDIANNESS enum

Byte ordering of this access.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

LOCK enum

Normal, exclusive or locked access.

MECID unsigned int

Memory Encryption Context ID, if applicable.

MECID2 unsigned int

Memory Encryption Context ID of the second page, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PAGE_SIZE unsigned int

Page size as $\log_2(\text{size in bytes})$.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK, 2=Exclusive Failed, 3=Unsupported.

SIZE unsigned int

Width of the access in bytes. Only required if DATA is not traced.

TRANS bool

Is this a translated access.

VADDR unsigned int

The virtual address of the access.

CRYPTO_SPEC

Every crypto instruction speculatively executed.

DATA_CACHE_SET_ALLOC_TAGS

Set allocation tags from DC GVA instruction. Fields:

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

NSDESC unsigned int

The security state of the access.

PADDR unsigned int

The physical address of the access.

PAS enum

The physical address space of the page.

RESPONSE enum

0=Aborted, 1=OK.

SIZE unsigned int

The size of memory in bytes.

TAG unsigned int

The value of the tag being set.

TAGGING_TYPE enum

Type of Memory tagging (Physical or Virtual).

TAG_VADDR unsigned int

The virtual address of the Virtual Tag if Virtual tagging is used; otherwise 0.

VADDR unsigned int

Virtual address used for setting allocation tags.

DATA_CACHE_ZERO

Zero and invalidate cache line from DC ZVA instruction. Fields:

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

NSDESC unsigned int

The security state of the access.

PADDR unsigned int

The physical address of the access.

PADDR_TAG unsigned int

If tag checked, the physical address tag for the access.

PAS enum

The physical address space of the page.

RESPONSE enum

0=Aborted, 1=OK.

SIZE unsigned int

The size of zero'd memory in bytes.

TAG_CHECKED bool

Is this access tag checked.

VADDR unsigned int

Virtual address of the zero'd cache line.

DEBUG_EVENT

Hardware debug support event. Fields:

EVENT enum

Description of event.

VALUE unsigned int

data value.

DMI_ALLOCATE

DMI allocated. Fields:

ACCESS_DIRECTION enum

READ / WRITE.

PADDR unsigned int

The physical address of the DMI.

PTR unsigned int

DMI range pointer.

SIDE enum

DSIDE / ISIDE.

SIZE unsigned int

The size of DMI.

VADDR unsigned int

The virtual address of the DMI..

DMI_HIT

DMI hit, access used DMI (only fire for small DMI regions i.e. less than a full page). Fields:

ACCESS_DIRECTION enum

READ / WRITE.

PADDR unsigned int

The physical address of the DMI.

SIDE enum

DSIDE / ISIDE.

SIZE unsigned int

The size of the DMI.

VADDR unsigned int

The virtual address of the DMI..

DMI_REVOKE

DMI revoked. Fields:

ACCESS_DIRECTION enum

READ / WRITE.

PADDR unsigned int

The physical address of the DMI.

SIDE enum

DSIDE / ISIDE.

SIZE unsigned int

The size of DMI.

VADDR unsigned int

The virtual address of the DMI..

DTCM_ACCESS

Trace TCM accesses.

DTCM_ACCESS_RAZ_WI

Trace TCM accesses. Fields:

ADDR **unsigned int**

Address of the **RAZ/WI** access.

DWT_MATCH

DWT comparator matches. Fields:

NUM **unsigned int**

DWT comparator number.

TYPE **enum**

DWT comparator configuration.

END_COMPILE

Compilation end. Fields:

END_OF_PAGE_COUNT **unsigned int**

Number of basic blocks exited due to page boundary since **START_COMPILE**.

FETCHFAIL_COUNT **unsigned int**

Number of basic blocks exited due to ifetch failure **START_COMPILE**.

INST_COUNT **unsigned int**

Number of instructions compiled since **START_COMPILE**.

NONSEQ_COUNT **unsigned int**

Number of basic blocks exited due to non-sequential instructions since **START_COMPILE**.

PAGE_STRADDLE_COUNT **unsigned int**

Number of basic blocks exited due to unaligned instructions crossing page since **START_COMPILE**.

EXCEPTION

Exceptions that are taken. Fields:

CORE_NUM **unsigned int**

Core number in a multi processor.

ISSET **enum**

The instruction set of the processor when the exception occurred.

LR **unsigned int**

The value assigned to the link register.

NS **enum**

The core's non-secure bit.

PC unsigned int

The location where the exception occurred.

TARGET_ISET enum

The instruction set of the exception handler code.

TARGET_PC unsigned int

The address the exception branches to.

VECTOR enum

The exception vector.

EXCEPTION_END

Every exception completed. Fields:

NS enum

The core's non-secure bit.

EXCEPTION_ENTRY

Event marking the entry of an exception. All pushes of registers to the stack will follow this event. Fields:

PC unsigned int

The location where the exception occurred.

VECTOR enum

The exception that occurred.

EXCEPTION_RAISE

Every exception raised.

EXCEPTION_RETURN

Marks the end of an exception, but on an M core does not indicate a program flow change (branch). In most situations it will be followed by an indirect branch. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISET enum

The instruction set of the branch instruction.

PC unsigned int

The address of the branch instruction.

EXCEPTION_RETURN_PREBRANCH

Exception return event that must occur before PC and processor state have updated.

EXCEPTION_START

Every exception started. Fields:

NS enum

The core's non-secure bit.

EXCEPTION_VECTOR_FETCH

I-side vector fetch for M-class exception. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

DATA unsigned int

Result of fetch if OK.

RESPONSE enum

0=Aborted, 1=OK.

VADDR unsigned int

Where the vector is fetched from (including VTOR).

FP_STATE

Floating point state. Fields:

FPSR unsigned int

FPSR register value.

MASK unsigned int

The mask indicating updated cumulative bits.

FREQ_CHANGED

Core frequency changed. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The instruction count of this core.

NEW_FREQ signed int

The new frequency of this core (expressed in Hz).

OLD_FREQ signed int

The old frequency of this core (expressed in Hz).

HLT

HLT instruction occurred. Fields:

IMM16 unsigned int

The 16-bit immediate value encoded in HLT instruction.

INFO_EXCEPTION_REASON

Provide information of real cause of exception. Fields:

FaultCause enum

Which bit of which V7M/V8M-mainline FSR/HFSR would have been set.

PC unsigned int

The location where the exception occurred.

PHASE enum

What we are doing with the exception.

REASONS enum

Why we are doing it.

SecurityState enum

Privacy State of exception.

VECTOR enum

The exception that occurred.

INFO_FP_CONTEXT

Inform when ExecuteFPCheck writes CONTROL.FPCA and FPSCR etc. Fields:

WHAT enum

What we are doing.

INFO_STACKING

Provide information of real cause of exception. Fields:

PHASE enum

Are we starting or ending a sequence of stack writes.

REASONS enum

Why we are doing it.

INST

Every instruction executed. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DEBUG_STATE bool

The instruction is executed in debug state.

DISASS string

Disassembly of instruction.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISET enum

The current instruction set.

ITSTATE unsigned int

The current ITSTATE.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

MODE enum

The mode the core is in.

NS unsigned int

The core's non-secure bit.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

OPCODE unsigned int

The opcode of the instruction.

PADDR unsigned int

The physical address of the instruction.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the instruction.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the page.

PC unsigned int

The address of the instruction.

SECURITY_STATE enum

The core's security state.

SIZE unsigned int

The size of the instruction in bytes.

INST_END

Every instruction completed.

INST_START

Every instruction started. Fields:

DEBUG_STATE bool

The instruction is executed in debug state.

ISET enum

The current instruction set.

MODE enum

The mode the core is in.

NS enum

The current Secure State.

PC unsigned int

The address of the conditional instruction.

SECURITY_STATE enum

The current Security State.

INST_STRADDLE

Instruction straddles boundary. Fields:

DEBUG_STATE bool

The instruction is executed in debug state.

ISSET enum

The current instruction set.

MODE enum

The mode the core is in.

NS enum

The current Secure State.

PC unsigned int

The address of the conditional instruction.

SECURITY_STATE enum

The current Security State.

ISB

ISB instruction occurred. Fields:

PC unsigned int

The address of the ISB instruction.

ITCM_ACCESS

Trace TCM accesses.

ITCM_ACCESS_RAZ_WI

Trace TCM accesses. Fields:

ADDR unsigned int

Address of the **RAZ/WI** access.

LOAD_MULTIPLE_REGS_LIST

List of destination registers (not source) of a multiple registers load instruction. Fields:

REG string

Register to add into the list.

LOCAL_MONITOR

Local monitor activity. Fields:

PADDR unsigned int

Local Monitor Address.

State enum

State of the monitor (Open/Exclusive).

LOCKUP_CYCLE

This event is triggered if the core enters or stays in the lock-up state. Fields:

CAUSE enum

What causes the lockup?.

LOCKUP_ENTRY

This event is triggered if the core enters the lock-up state. Fields:

PC unsigned int

The PC when the lockup state is entered.

LOCKUP_EXIT

This event is triggered if the core leaves the lock-up state.

MODE_CHANGE

Mode change. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

MODE enum

The new mode.

OLD_MODE enum

The old mode.

MPU_TRANS

Address translation information. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

EXEC_PERM enum

Execution Permission.

IDAU_REG_NUM enum

IDAU region-number or code.

INNERCACHE_RA bool

Is the inner cache allocate on read.

INNERCACHE_TYPE enum

Inner Caching scheme (NC/MB/WA).

INNERCACHE_WA bool

Is the inner cache allocate on write.

MEMTYPE enum

Memory type.

NSDESC enum

Security-attributes of the transaction.

OUTERCACHE_RA bool

Is the outer cache allocate on read.

OUTERCACHE_TYPE enum

Outer Caching scheme (NC/MB/WA).

OUTERCACHE_WA bool

Is the outer cache allocate on write.

PADDR unsigned int

Address of the access.

PAGESIZE unsigned int

Page size as $\log_2(\text{size})$.

PAS enum

PAS accessed is secure or nonsecure memory.

READ_PERM enum

Read Permission.

REG_NUM unsigned int

MPU register based region-number 0-15 (else 0xFF=none, 0xBB/0xDD=background/default).

SAU_REG_NUM enum

SAU region-number or code.

SAU_RGN_TYPE enum

Privacy Region Type.

SH enum

Shareability.

SIDE enum

Inst / Data.

WRITE_PERM enum

Write Permission.

PERIODIC

Called for every quantum. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The instruction count of this CPU.

PC unsigned int

The address of the next instruction to be executed on this CPU.

POTINDEX_UPDATE

Value of POT index for current pc changed. Fields:

NEW_POTINDEX unsigned int

New POTIndex.

OLD_POTINDEX unsigned int

Old POTIndex.

PPB_READ_ACCESS

Trace reads from the core to memory mapped registers in the PPB (private peripheral bus) address range. Fields:

ACCESS_SIZE unsigned int

Logarithm of the access size: 0=byte, 1=halfword, 2=word, 3=doubleword, ...

BURST_LEN unsigned int

The number of data transfers (beats) in this burst.

DATA unsigned int

The data transferred.

INNER_ATTRIBUTES unsigned int

The inner attributes: bit0=write allocate, bit1=read allocate, bit2=cachable, bit3=bufferable, bit4=shareable.

INST bool

Is an instruction fetch.

LOCK enum

Normal / Locked / Exclusive.

MANAGER_ID unsigned int

The AXI manager ID.

NS bool

Is a nonsecure access.

OUTER_ATTRIBUTES unsigned int

The outer attributes: bit0=write allocate, bit1=read allocate, bit2=cachable, bit3=bufferable, bit4=shareable.

PADDR unsigned int

Physical address of access.

PRIV bool

Is privileged access.

RESPONSE enum

Whether the transaction was successful, or an error occurred.

USER_FLAGS unsigned int

Core specific additional signals.

PPB_WRITE_ACCESS

Trace writes from the core to memory mapped registers in the PPB (private peripheral bus) address range. Fields:

ACCESS_SIZE unsigned int

Logarithm of the access size: 0=byte, 1=halfword, 2=word, 3=doubleword, ...

BURST_LEN unsigned int

The number of data transfers (beats) in this burst.

DATA unsigned int

The data transferred.

INNER_ATTRIBUTES unsigned int

The inner attributes: bit0=write allocate, bit1=read allocate, bit2=cachable, bit3=bufferable, bit4=shareable.

INST bool

Is an instruction fetch.

LOCK enum

Normal / Locked / Exclusive.

MANAGER_ID unsigned int

The AXI manager ID.

NS bool

Is a nonsecure access.

OUTER_ATTRIBUTES unsigned int

The outer attributes: bit0=write allocate, bit1=read allocate, bit2=cachable, bit3=bufferable, bit4=shareable.

PADDR unsigned int

Physical address of access.

PRIV bool

Is privileged access.

RESPONSE enum

Whether the transaction was successful, or an error occurred.

USER_FLAGS unsigned int

Core specific additional signals.

PRELOAD_DATA

Data preload from PLD instruction. Fields:

VADDR unsigned int

Virtual address of the data that should be preloaded.

PRELOAD_INST

Instruction preload from PLI instruction. Fields:

VADDR unsigned int

Virtual address of the instruction that should be preloaded.

PRE_CORE_LOAD

Trace just before a core load. Fields:

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

PRE_CORE_STORE

Trace just before a core store. Fields:

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

PSTATE_PACTIVE_OPMODE_TRANSITION

Trace pactive and Opmode mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Operating mode after the transition.

OLD_STATE enum

Operating mode before the transition.

PSTATE_PACTIVE_OPMODE_TRANSITION_DENIED

Trace the denied pactive and Opmode mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Operating mode after the transition.

OLD_STATE enum

Operating mode before the transition.

REASON string

Reason for the transition being denied.

PSTATE_PACTIVE_POWER_TRANSITION

Trace pactive and power mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Power mode after the transition.

OLD_STATE enum

Power mode before the transition.

PSTATE_PACTIVE_POWER_TRANSITION_DENIED

Trace the denied pactive and power mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Power mode after the transition.

OLD_STATE enum

Power mode before the transition.

REASON string

Reason for the transition being denied.

PSTATE_PACTIVE_THREAD_OPMODE_TRANSITION

Trace pactive and Opmode mode transitions for threads on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Operating mode after the transition.

OLD_STATE enum

Operating mode before the transition.

QUANTUM_START

Called at the start of every quantum.

RUN_STATE

Run state transition. Fields:

INST_COUNT unsigned int

Ticks count at point of transition.

NEW enum

New run state.

OLD enum

Old run state.

SEMIHOSTING_CALL

Call of a semihost function occurred. Fields:

PC unsigned int

The program counter after the semihosting call.

SEMIHOSTING_PRECALL

About to call semihost. Fields:

EL2 bool

Use EL2 translation regime.

NS bool

Is Non-Secure.

REG_WIDTH unsigned int

The current register width in bytes.

SIGNAL

External signal state change. Fields:

SIGNAL enum

Signal that changed.

STATE bool

Signal asserted state.

START_COMPILE

Compilation started. Fields:

VADDR unsigned int

Instruction where compilation begins.

STG_MULTIPLE_MICRO_OPS_ADDRESS_RANGE

Address range for MTE stores with multiple uops. Fields:

ADDR_END unsigned int

Final address.

ADDR_START unsigned int

Starting address.

STORE_MULTIPLE_REGS_ADDRESS_RANGE

Address range for stores with multiple registers. Fields:

ADDR_END unsigned int

Final address.

ADDR_START unsigned int

Starting address.

SUBOPTIMAL_LDST_RETIRED

suboptimal load/store. Fields:

ACCESS_SIZE unsigned int

Log2 of the access width in bytes used for alignment checking (1: half-word, 2: word ...).

OPTIMAL_ALIGN_SELECTOR unsigned int

Selector to decide the optimal alignment size (1: access width, 2: 4-bytes, 3: 8-bytes, ..., 12: 4-kbytes).

VADDR unsigned int

The virtual address of the access.

SYNC

Called for every synchronization, that is every quantum break. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The instruction count of this CPU.

LOCAL_QUANTUM unsigned int

The local quantum of this CPU.

LOCAL_TIME unsigned int

The local time of this CPU.

SYNC_REASON enum

Reason for sync.

SYSCALL

System call instruction executed. Fields:

IMM unsigned int

Immediate value of the system call instruction.

TYPE enum

System call type.

VADDR unsigned int

Instruction that caused the system call.

SYSREG_READ

Trace reads from system registers. Fields:

REG_NAME enum

Register number.

VALUE unsigned int

The value read.

SYSREG_UPDATE128

Triggers when the system updates a register. Fields:

REG enum

Register number.

UNKNOWN unsigned int

Bits of the register which became unknown.

VALUE unsigned int

Value written to the register.

SYSREG_UPDATE32

Triggers when the system updates a register. Fields:

REG enum

Register number.

UNKNOWN unsigned int

Bits of the register which became unknown.

VALUE unsigned int

Value written to the register.

SYSREG_UPDATE64

Triggers when the system updates a register. Fields:

REG enum

Register number.

UNKNOWN unsigned int

Bits of the register which became unknown.

VALUE unsigned int

Value written to the register.

SYSREG_WRITE

Trace write to system registers. Fields:

REG_ACCESS enum

Register access status.

REG_NAME enum

Register number.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

Value written to the register.

UNALIGNED_LDST_RETIRED

Processor unaligned load/store. Fields:

ACCESS_SIZE unsigned int

Log2 of the access width in bytes used for alignment checking (1=>half-word, 2->word ...).

NUMBER_OF_BEATS unsigned int

Number of accesses (beats) in this burst (Total size in bytes is ACCESS_SIZE * NUMBER_OF_BEATS).

VADDR unsigned int

The virtual address of the access.

VFP_D_REGS

VFP/NEON D 64 bit register write. Fields:

ALIASING enum

Type of register aliasing used.

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a write to the corresponding bit in the register. For example, 0x1 would indicate a write to VALUE[0].

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

VFP_D_REGS_READ

VFP/NEON D 64 bit register read. Fields:

ID unsigned int

The register number.

VALUE unsigned int

The current value of the register.

VFP_Q_REGS

VFP/NEON Q 128 bit register write. Fields:

ALIASING enum

Type of register aliasing used.

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a write to the corresponding byte in the register. For example, 0x1 would indicate a write to VALUE[7:0].

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

VFP_Q_REGS_READ

VFP/NEON Q 128 bit register read. Fields:

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a read to the corresponding byte in the register. For example, 0x1 would indicate a read to VALUE[7:0].

VALUE unsigned int

The current value of the register.

VFP_SYS_REGS

Writes to the VFP/NEON units system registers. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ID enum

Which VFP system register is written.

MASK unsigned int

The mask indicating updated cumulative bits.

OLD_VALUE unsigned int

The register's old value overwritten.

VALUE unsigned int

The new value written to the VFP system register.

VFP_S_REGS

VFP/NEON S 32 bit register write. Fields:

ALIASING enum

Type of register aliasing used.

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a write to the corresponding bit in the register. For example, 0x1 would indicate a write to VALUE[0].

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

VFP_S_REGS_READ

VFP/NEON S 32 bit register read. Fields:

ID unsigned int

The register number.

VALUE unsigned int

The current value of the register.

WAYPOINT

Signals end of basic block execution. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ISET enum

Origin instruction set.

IS_COND bool

Indicates if this is a conditional waypoint.

PC unsigned int

Origin address (or 0 if unavailable).

TAKEN bool

Indicates if this waypoint was taken.

TARGET unsigned int

Destination address.

TARGET_ISET enum

Destination instruction set.

WFE_END

WFE ended. Fields:

INST_COUNT unsigned int

Ticks count when leaving WFE.

WFE_EVENT_REGISTER

WFE event register status: set/clear, reason. Fields:

INST_COUNT unsigned int

Ticks count.

REASON enum

Reason for set/clear. Only REASON==0 (Cleared by WFE) clears the bit, all other reasons set it.

WFE_IGNORED

WFE ignored. Fields:

EVENT bool

This WFE was ignored because the event register was set.

INST_COUNT unsigned int

Ticks count when ignoring WFE.

TRAPPED bool

This WFE was trapped.

WFE_START

WFE entered. Fields:

INST_COUNT unsigned int

Ticks count when entering WFE.

WFI_END

WFI ended. Fields:

INST_COUNT unsigned int

Ticks count when leaving WFI.

WFI_IGNORED

WFI ignored. Fields:

DISABLED bool

This WFI was ignored because WFI is disabled.

INST_COUNT unsigned int

Ticks count when ignoring WFI.

REASON enum

specifies reason why WFI trace was ignored.

TRAPPED bool

This WFI was trapped.

WFI_START

WFI entered. Fields:

INST_COUNT unsigned int

Ticks count when entering WFI.

WFI_WAKEUP

WFI wakeup. Fields:

INST_COUNT **unsigned int**

Ticks count when WFI wakeup occurred.

REASON **enum**

Reason for wakeup.

XPSR

Changes to the xPSR register. Fields:

CORE_NUM **unsigned int**

Core number in a multi processor.

OLD_VALUE **unsigned int**

The old xPSR value.

UNKNOWN **unsigned int**

Bits within the register that have unknown value.

VALUE **unsigned int**

The new xPSR value.

2.8 ARM_C1-Nano

This section describes the trace sources.

AA64_ASE_SVE_REGS

Modification of AArch64 Advanced SIMD or SVE Vector Registers (via SIMD or Scalar operations), SVE Predicate Registers and ZTO. Fields:

ID **enum**

The register identifier.

MASK **unsigned int**

Bitmask of the register to signal the modified bits in the VALUE field.

SM **bool**

Whether the PE is in Streaming Mode.

VALUE **unsigned int**

Value of AArch64 vector register.

ASYNC_MEMORY_FAULT

Context ID Register write. Fields:

FAULT **unsigned int**

Fault status present in ESR.ISS encoding for Serror in ARCH64 and present in DFSR encoding for external abort format in ARCH32.

PADDR unsigned int

Physical Address (or 0 if unavailable).

VADDR signed int

Virtual Address (or 0 if unavailable).

ATOMIC_END_ACCESS

Bus trace access for atomics. Fields:

ACCESS_FAIL bool

Memory access failed.

ADDR unsigned int

The virtual address of the access.

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

COMPARE_VALUE unsigned int

Compare value for CAS.

LOAD_VALUE unsigned int

Loaded values.

NSDESC unsigned int

The security state of the access.

OPERAND_VALUE unsigned int

Operation's operand.

OPERATION enum

Operation type.

PADDR unsigned int

The physical address of the access.

PRIV bool

Is this a privileged access?.

ATOMIC_START_ACCESS

Bus trace access for atomics. Fields:

ADDR unsigned int

The virtual address of the access.

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

COMPARE_VALUE unsigned int

Compare value for CAS.

NSDESC unsigned int

The security state of the access.

OPERAND_VALUE unsigned int

Operation's operand.

OPERATION enum

Operation type.

PADDR unsigned int

The physical address of the access.

PRIV bool

Is this a privileged access?.

ArchMsg.Error.exit_code

DISPLAY exit code: %{EXITCODE} component kind: %{KIND} reason: %{REASON}. Fields:

EXITCODE signed int

Exit code to be returned.

KIND string

Component kind that invoked the exit code trace.

REASON string

Reason why the exit code trace was invoked (optional).

ArchMsg.Warning.IgnoredWarmResetRequest

Event fires when software attempts to request warm reset through write to deprecated DBGPRCR.CWRR.

ArchMsg.Warning.RAS_ErrselrUnpredictableState

DISPLAY ERRSELR is in unpredictable state due to value greater than number_of_error_records written to it.

ArchMsg.Warning.RAS_UnpredictableStateWrite

DISPLAY %{REG_NAME} write ignored due to unpredictable ERRSELR state. Fields:

REG_NAME string

Name of register being written.

ArchMsg.Warning.ReportedRevisionRegisterRead

DISPLAY Reported revision and patch do not match that of the cpu. Fields:

ORIGINAL_VALUE unsigned int

The value the register would have based on the modelled revision.

REGISTER string

The name of the register that has been read.

REPORTED_VALUE unsigned int

The value the register has been overridden to.

ArchMsg.Warning.Unpredictably Indexed PM Event Register

TAGS **UNPREDICTABLE** DISPLAY Event %{IsDirect:(Selector value|Register)} %{SEL} is >= %{N}.
Fields:

IsDirect unsigned int

Direct Access rather than indirect.

N unsigned int

Number of Accessible Registers.

SEL unsigned int

Selector.

ArchMsg.Warning.cache_contents_unknown

A cache has been activated without being invalidated and may contain **UNKNOWN** content.
DISPLAY %{SIDE:(D|I)}-Cache was enabled in %{REGIME:(SECURE|NON_SECURE|HYP)} regime but was not invalidated since power-on: cache lines could contain **UNKNOWN** content
%{SIDE:(Note: D-side caches must invalidate SECURE lines even if the cache is only used by NON_SECURE code in order to ensure that unexpected dirty lines tagged SECURE are not naturally evicted)}. Fields:

REGIME unsigned int

Current translation regime.

SIDE unsigned int

0 d-side cache or unified cache, 1 i-side.

ArchMsg.Warning.dcimvac_matches_watchpoint

TAGS **UNPREDICTABLE** DISPLAY Watchpoints matching an AArch32 DCIMVA are implementation defined.

ArchMsg.Warning.decode_fieldmismatch

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with different fields %{FIELD1} and %{FIELD2}. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_frachitsnegative

TAGS **UNPREDICTABLE** DISPLAY VCVT Instruction is unpredictable with a negative number of fraction bits.

ArchMsg.Warning.decode_initblock

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable in an IT block.

ArchMsg.Warning.decode_initblocknotlast

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable in an IT block when not the last.

ArchMsg.Warning.decode_invalid_condition

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable when the condition is not AL.

ArchMsg.Warning.decode_invalidfieldcombination

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with the values in %{FIELD1} and %{FIELD2}. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_invalidvalue

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with the value in %{FIELD}. Fields:

FIELD enum

Field name.

ArchMsg.Warning.decode_mem_hint_unallocated

TAGS **UNPREDICTABLE** DISPLAY There is no memory hint allocated to this bit pattern.

ArchMsg.Warning.decode_registermatch

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with %{FIELD1} and %{FIELD2} as the same register. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_registermismatch

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with %{FIELD1} and %{FIELD2} as different registers. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_registeroutofrange

TAGS **UNPREDICTABLE** DISPLAY Floating-point instruction is unpredictable indexing beyond the end of the register bank.

ArchMsg.Warning.decode_sbzsbo

TAGS UNPREDICTABLE DISPLAY Reserved bits in the instruction are not canonical.

ArchMsg.Warning.decode_transactiontoobig

TAGS UNPREDICTABLE DISPLAY Attempt to perform a memory transaction of more than 128 bytes.

ArchMsg.Warning.decode_unpred_other

TAGS UNPREDICTABLE DISPLAY Instruction is unpredictable.

ArchMsg.Warning.decode_unpreduseofpc

TAGS UNPREDICTABLE DISPLAY Instruction is unpredictable when %{FIELD} is PC. Fields:

FIELD enum

Field name.

ArchMsg.Warning.decode_unpreduseofr13

TAGS UNPREDICTABLE DISPLAY Instruction is unpredictable when %{FIELD} is R13. Fields:

FIELD enum

Field name.

ArchMsg.Warning.decode_unpreduseofr13inreglist

TAGS UNPREDICTABLE DISPLAY Instruction is unpredictable with R13 in register-list.

ArchMsg.Warning.decode_useofd16orhigher

TAGS UNPREDICTABLE DISPLAY Attempt to use register D16 or higher where it is not allowed.

ArchMsg.Warning.decode_writebackandbaseinlist

TAGS UNPREDICTABLE DISPLAY Instruction is unpredictable with write back when the base register is in the transfer list.

ArchMsg.Warning.decode_zeroregistersinlist

TAGS UNPREDICTABLE DISPLAY Instruction is unpredictable when no registers are to be transferred.

ArchMsg.Warning.misaligned_ptrs

DISPLAY TRBPTR_EL1(%{TRBPTR_EL1}) or TRBTRG_EL1(%{TRBTRG_EL1}) are not aligned as per TRBIDR_EL1.Align(%{TRBIDR_EL1}). Fields:

TRBIDR_EL1 unsigned int

TRBIDR_EL1.Align value.

TRBPTR_EL1 unsigned int

Current write pointer.

TRBTRG_EL1 unsigned int

Trigger pointer.

ArchMsg.Warning.out_of_range_ptrs

DISPLAY TRBPTR_EL1(%{TRBPTR_EL1}) is out of range as per TRBBASER_EL1(%{TRBBASER_EL1}) or TRBLIMITR_EL1.LIMIT(%{TRBLIMITR_EL1}). Fields:

TRBBASER_EL1 unsigned int

Base pointer.

TRBLIMITR_EL1 unsigned int

Limit pointer.

TRBPTR_EL1 unsigned int

Current write pointer.

ArchMsg.Warning.recursive_branch

An instruction is performing a branch that targets the same instruction. If this was intended then a WFI might be more effective. DISPLAY recursive branch detected at PC=%{PC}.

PRIMARY KEY PC. Fields:

PC unsigned int

Address of instruction causing the branch.

ArchMsg.Warning.recursive_exception

An instruction has generated an exception that targets the same instruction. DISPLAY recursive %{TYPE} exception detected at PC=%{PC} (CPSR=%{CPSR}). PRIMARY KEY PC. Fields:

CPSR unsigned int

Processor state.

PC unsigned int

Address of instruction causing the exception.

TYPE enum

Type of exception.

ArchMsg.Warning.reentrant_vector_catch

TAGS **UNPREDICTABLE** DISPLAY Vector catch debug event on Prefetch or Data abort vector %{ADDR}. Fields:

ADDR unsigned int

vector address.

ArchMsg.Warning.reserved_it_state

TAGS **UNPREDICTABLE** DISPLAY Execution with reserved IT state %{ITSTATE}. Fields:

ITSTATE unsigned int

reserved IT state.

ArchMsg.Warning.tlb_contents_unknown

Translation has been enabled without invalidating the TLB. DISPLAY Translation was enabled in the %{PAS:(Secure|Non-secure|Root|Realm)} EL%{EL:d} translation regime (vmid=%{VMID:x}) without invalidating %{INVALIDITY:(|d-side |i-side |)}entries since power-on: %{INVALIDITY:(|D||I|)}TLB could contain **UNKNOWN** content. Fields:

EL unsigned int

Exception level of translation regime.

INVALIDITY unsigned int

Which TLB may be invalid (if TLBs are separate).

NON_SECURE bool

Whether the translation regime is non-secure.

PAS unsigned int

Physical address space of translation regime.

VMID unsigned int

Current VMID, if applicable.

ArchMsg.Warning.unknown_DLR_DSPSR

Exiting debug state with unknown DLR or DSPSR.

ArchMsg.Warning.unknown_ELR_SPSR

Returning from debug exception with unknown ELR or SPSR.

ArchMsg.Warning.unpred_FAR_on_LRCPC3_non_atomic_mem_access

TAGS **UNPREDICTABLE** Unpredictable FAR/FSTATUS on LRCPC3 pair instructions resulting in two separate memory accesses
 DISPLAY Unpredictable FAR on LRCPC3 non atomic memory access at %{ADDR1} with FSTATUS %{FSTATUS1} or %{ADDR2} with FSTATUS %{FSTATUS2}. Fields:

ADDR1 unsigned int

Possible FAR address 1.

ADDR2 unsigned int

Possible FAR address 2.

FSTATUS1 unsigned int

Possible fault status value 1.

FSTATUS2 unsigned int

Possible fault status value 2.

ArchMsg.Warning.unpred_async_tag_check_fail_on_sve_speculative_load_do_update_ffr

Unpredictable set to FALSE of the FFR element associated to SVE speculative load with an async tag check fail TAGS **UNPREDICTABLE**.

ArchMsg.Warning.unpredictable_a32_breakpoint

TAGS **UNPREDICTABLE** DISPLAY Unpredictable breakpoint on a A32 instruction at %{ADDR1} Offset %{OFFSET} BAS %{BAS} for a %{IS_ADDRESS_MISMATCH_BREAKPOINT:(Mismatch|Match)} Breakpoint number %{N:d} Breakpoint address %{ADDR2}. Fields:

ADDR1 unsigned int

Address.

ADDR2 unsigned int

Breakpoint address.

BAS unsigned int

BAS field.

IS_ADDRESS_MISMATCH_BREAKPOINT bool

Is a Mismatch breakpoint, not a Match breakpoint.

N unsigned int

Breakpoint number.

OFFSET unsigned int

Always has a value of zero.

ArchMsg.Warning.unpredictable_banked_register_access

TAGS **UNPREDICTABLE** DISPLAY Access to banked register (R=%{R:d} SYSm=%{SYSM:2x}) is unpredictable. Fields:

R unsigned int

R bit.

SYSM unsigned int

SYSm.

ArchMsg.Warning.unpredictable_ccfail_undef

TAGS **UNPREDICTABLE** DISPLAY An instruction %{OPCODE} at VA=%{VADDR} failed its condition codes check but would otherwise have caused a non-data-dependent trap or undefined exception. Behaviour is implementation-defined choice of exception or nop. Fields:

OPCODE unsigned int

opcode of the instruction.

VADDR unsigned int

Virtual address of the instruction being executed.

ArchMsg.Warning.unpredictable_ici_bits

TAGS **UNPREDICTABLE** DISPLAY ICI bits do not apply or are architecturally unpredictable for instruction.

ArchMsg.Warning.unpredictable_t32_breakpoint

TAGS **UNPREDICTABLE** DISPLAY Unpredictable breakpoint on a T32 instruction at %{ADDR1} Offset %{OFFSET} BAS %{BAS} for a %{IS_ADDRESS_MISMATCH_BREAKPOINT:(Mismatch|Match)} Breakpoint number %{N:d} Breakpoint address %{ADDR2}. Fields:

ADDR1 unsigned int

Address.

ADDR2 unsigned int

Breakpoint address.

BAS unsigned int

BAS field.

IS_ADDRESS_MISMATCH_BREAKPOINT bool

Is a Mismatch breakpoint, not a Match breakpoint.

N unsigned int

Breakpoint number.

OFFSET unsigned int

Indicates which halfword of a 32-bit Thumb instruction the breakpoint is set on. 0 for the first halfword or if the instruction is 16 bits wide, or 1 for the second halfword.

ArchMsg.Warning.unpredictable_t32_vector_catch

TAGS **UNPREDICTABLE** DISPLAY Unpredictable Vector catch exception on a T32 instruction at %{ADDR} Offset %{OFFSET}. Fields:

ADDR unsigned int

Address.

OFFSET unsigned int

Indicates which halfword of a 32-bit Thumb instruction the vector catch exception is set on. 0 for the first halfword or if the instruction is 16 bits wide, or 1 for the second halfword.

ArchMsg.Warning.unpredictable_watchpoint_far

TAGS **UNPREDICTABLE** DISPLAY The value for FAR/EDWAR for hitting this watchpoint may be anywhere in the range %{LOWER_BOUND} to %{UPPER_BOUND}. Fields:

LOWER_BOUND unsigned int

the lowest address accessed by the instruction that triggered the watchpoint.

UPPER_BOUND unsigned int

the highest watchpointed address accessed by that instruction.

ArchMsg.Warning.unpredictable_watchpoint_far_isscmp

DISPLAY The value for FAR/EDWAR for hitting this watchpoint may be anywhere in the range %{LOWER_BOUND} to %{UPPER_BOUND}. Fields:

LOWER_BOUND unsigned int

the lowest address accessed by the instruction that triggered the watchpoint.

UPPER_BOUND unsigned int

the highest watchpointed address accessed by that instruction.

ArchMsg.Warning.warning_AdvSIMDExpandImmUnexpectedZero

TAGS **UNPREDICTABLE** DISPLAY AdvSIMDExpandImm may treat this immediate value as **UNPREDICTABLE**.

ArchMsg.Warning.warning_ConditionalSMC

TAGS **UNPREDICTABLE** DISPLAY SMC instruction has **UNPREDICTABLE** effects when conditional and when combined with traps.

ArchMsg.Warning.warning_TLBID_out_of_range

TLBID out of range in TLBI according to the configuration of NIS, NVIS, NOS and NVOS. Fields:

NIS unsigned int

NIS.

NOS unsigned int

NOS.

NVIS unsigned int

NVIS.

NVOS unsigned int

NVOS.

TLBID unsigned int

The TLBID value used in the TLBI instruction.

ArchMsg.Warning.warning_TTL_or_TG_mismatches_current_translation_granule

TTL or TG mismatches current translation granule in TLBI VA/RVA/IPA/RIPA. Fields:

CURRENT_TG unsigned int

Current TG as indicated by TCR_ELx or VTCR_EL2.

TLBI_TG unsigned int

TG (or TTL[3:2]) used in TLBI instruction.

ArchMsg.Warning.warning_access_crosses_page_boundary

TAGS **UNPREDICTABLE** DISPLAY Access crosses page boundary.

ArchMsg.Warning.warning_access_crosses_page_boundary_has_fault

TAGS **UNPREDICTABLE** DISPLAY Access that crosses page boundary has fault(s).

ArchMsg.Warning.warning_access_crosses_page_boundary_has_fault_on_both_pages

TAGS **UNPREDICTABLE** DISPLAY Access that crosses page boundary has a fault on each page.
Fields:

ADDR1 unsigned int

FAR address on first page.

ADDR2 unsigned int

FAR address on second page.

FSTATUS1 unsigned int

Fault status on first page.

FSTATUS2 unsigned int

Fault status on second page.

ArchMsg.Warning.warning_access_crosses_page_boundary_spanning_different_memory

TAGS **UNPREDICTABLE** DISPLAY Access at address %{ADDR} crosses page from MEM_TYPE %{MEMTYPE_PAGE1} to %{MEMTYPE_PAGE2}. Fields:

ADDR unsigned int

address of access crossing page.

MEMTYPE_PAGE1 signed int

First Page's memory type (0-Normal, 1-Non-Normal).

MEMTYPE_PAGE2 signed int

Second Page's memory type (0-Normal, 1-Non-Normal).

ArchMsg.Warning.warning_access_wraps_around_memory

TAGS UNPREDICTABLE DISPLAY Access wraps around memory in %{WIDTH:d}bit mode.
Fields:

WIDTH unsigned int

Address width.

ArchMsg.Warning.warning_bcr_linking_status

TAGS UNPREDICTABLE DISPLAY DBGBCR.{BT2,BT} combination gives unpredictable behaviour.

ArchMsg.Warning.warning_bcr_mask_reserved

TAGS UNPREDICTABLE DISPLAY DBGBCR_ELn.MASK is set to a reserved value.

ArchMsg.Warning.warning_bvr_masked_bit_not_zero

TAGS UNPREDICTABLE DISPLAY DBGBVRn_EL1.VA contains a bit masked by DBGBCRn_EL1.MASK that is not zero.

ArchMsg.Warning.warning_ccsidr_unimplemented_level

TAGS UNPREDICTABLE DISPLAY CCSIDR read while CSSELR %{CSSELR:x} points to an unimplemented cache level. Fields:

CSSELR unsigned int

current effective value of CSSELR.

ArchMsg.Warning.warning_change_to_ns_when_tge_set

TAGS UNPREDICTABLE DISPLAY Attempting to change NS to 1 when HCR.TGE = 1.

ArchMsg.Warning.warning_contiguous_bit_check_abort

TAGS UNPREDICTABLE DISPLAY An abort occurred whilst attempting to check TLB entry is contiguous at %{ENTRY_ADDR}. Fields:

ENTRY_ADDR unsigned int

address of conflicting TLB entry.

ArchMsg.Warning.warning_contiguous_bit_error

TAGS UNPREDICTABLE DISPLAY TLB entry at %{ENTRY_ADDR} was not contiguous with entry at %{CONTIG_ADDR} Contents are %{ENTRY_DATA} but expected %{CONTIG_DATA}.
Fields:

CONTIG_ADDR unsigned int

address of the TLB first read, with which this entry is expected to be contiguous.

CONTIG_DATA unsigned int

expected contents based on the entry at CONTIG_ADDR.

ENTRY_ADDR unsigned int

address of conflicting TLB entry.

ENTRY_DATA unsigned int

contents of conflicting TLB entry.

ArchMsg.Warning.warning_contiguous_bit_error_D128

TAGS **UNPREDICTABLE** DISPLAY TLB entry at %{ENTRY_ADDR} was not contiguous with entry at %{CONTIG_ADDR} Contents are %{ENTRY_DATA_HI} %{ENTRY_DATA_LO} but expected %{CONTIG_DATA}. Fields:

CONTIG_ADDR unsigned int

address of the TLB first read, with which this entry is expected to be contiguous.

CONTIG_DATA unsigned int

expected contents based on the entry at CONTIG_ADDR.

ENTRY_ADDR unsigned int

address of conflicting TLB entry.

ENTRY_DATA_HI unsigned int

Higher bit contents of conflicting TLB entry.

ENTRY_DATA_LO unsigned int

Lower bit contents of conflicting TLB entry.

ArchMsg.Warning.warning_cp10_cp11_mismatch

TAGS **UNPREDICTABLE** DISPLAY Unpredictable configuration of CP10 and CP11 controls at EL %{EL:d}. Fields:

EL unsigned int

EL owning the control.

ArchMsg.Warning.warning_cp10_cp11_reserved_value

TAGS **UNPREDICTABLE** DISPLAY Unpredictable configuration of CPACR CP10 (%{CP10}) or CP11 (%{CP11}). Fields:

CP10 unsigned int

CP10 access.

CP11 unsigned int

CP11 access.

ArchMsg.Warning.warning_csselr_level_out_of_range

TAGS **UNPREDICTABLE** DISPLAY CSSELR.Level out of range, written value %{LEVEL} with only %{IMPLEMENTED:d} levels implemented. Fields:

IMPLEMENTED unsigned int

number of cache levels implemented.

LEVEL unsigned int

written value of CSSELR.Level.

TYPE enum

The type of cache selected by CSSELR.InD and TnD.

ArchMsg.Warning.warning_debug_flow_control_bits_not_obeyed

TAGS **UNPREDICTABLE** DISPLAY Flow control bits for %{REGISTER} were not obeyed, causing %{ERROR}. Fields:

ERROR enum

type of error resulting.

REGISTER enum

register under consideration.

ArchMsg.Warning.warning_debug_register_access_during_reset

TAGS **UNPREDICTABLE** DISPLAY Unpredictable %{IS_WRITE:(read|write)} access to debug register offset %{OFFSET} during reset. Fields:

IS_WRITE unsigned int

Write Not Read.

OFFSET unsigned int

Register Offset.

ArchMsg.Warning.warning_decode_cps_inconsistent_fields

TAGS **UNPREDICTABLE** DISPLAY The mode setting fields in the CPS instruction are inconsistent.

ArchMsg.Warning.warning_decode_invalid_state

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable in current exception-level/security state.

ArchMsg.Warning.warning_default_cacheable_mmu_on

TAGS **UNPREDICTABLE** DISPLAY Default cacheable is enabled (HCR.DC = 1) while MMU is enabled.

ArchMsg.Warning.warning_default_cacheable_vmmu_off

TAGS **UNPREDICTABLE** DISPLAY Default cacheable is enabled (HCR.DC = 1) while HCR.VM = 0.

ArchMsg.Warning.warning_deprecated_wvr_bit_2_set

TAGS **UNPREDICTABLE** DISPLAY DBGWVR%{N:d}_EL1=%{WVR:h} has bit 2 set, deprecated in ARMv8. Fields:

N unsigned int

index.

WVR unsigned int

DBGWVR.

ArchMsg.Warning.warning_eret_while_software_step_active_pending

Missing ISB between setting MDSCR_EL1.SS and ERET TAGS **UNPREDICTABLE**.

ArchMsg.Warning.warning_exclusive_to_non_normal

TAGS **UNPREDICTABLE** DISPLAY Exclusive Access to Non-Normal Memory %{ADDR}. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_exclusive_to_non_writeback

TAGS **UNPREDICTABLE** DISPLAY Exclusive Access to Non-Writeback-Cacheable Memory %{ADDR}. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_execute_from_device_memory

TAGS **UNPREDICTABLE** DISPLAY An attempt was made to execute from Device memory.

ArchMsg.Warning.warning_illegal_cpsr_mode

TAGS **UNPREDICTABLE** DISPLAY Writing an illegal or unimplement mode (%{NEW_MODE:h}) to CPSR was unpredictable. Fields:

NEW_MODE unsigned int

New value of CPSR.M.

ArchMsg.Warning.warning_illegal_srs_mode

TAGS **UNPREDICTABLE** DISPLAY Illegal or **UNPREDICTABLE** mode (%{MODE:h}) used for SRS instruction. Fields:

MODE unsigned int

mode for Banked SP.

ArchMsg.Warning.warning_implementation_defined_read_debug_register_in_SCS

Read Access to Debug registers from the processor is **IMPLEMENTATION DEFINED** TAGS **IMP_DEF**. Fields:

PPB_OFFSET unsigned int

debug register offset.

ArchMsg.Warning.warning_implementation_defined_sequential_security_transitions_supported

The behavior of sequential instruction fetches that cross from non-secure to secure memory and contain an SG instruction is **CONSTRAINED UNPREDICTABLE** TAGS **UNPREDICTABLE**.

ArchMsg.Warning.warning_implementation_defined_stack_limit_check_supported

It is **IMPLEMENTATION DEFINED** whether stack pointer limit checking is performed for this instruction TAGS **IMP_DEF**. Fields:

ADDRESS unsigned int

Address of instruction.

OPCODE unsigned int

instruction opcode.

ArchMsg.Warning.warning_implementation_defined_taggednonsharedmemorytype

TAGS IMP_DEF DISPLAY Memory attribute has Tagged, non shared memory type; it is **IMPLEMENTATION DEFINED** whether the memory location is treated as Tagged or non-Tagged.

ArchMsg.Warning.warning_implementation_defined_write_debug_register_in_SCS

Write Access to Debug registers from the processor is **IMPLEMENTATION DEFINED** TAGS IMP_DEF. Fields:

DATA unsigned int

data attempted to be written.

PPB_OFFSET unsigned int

debug register offset.

ArchMsg.Warning.warning_invalid_tcr_granule

TAGS **UNPREDICTABLE** DISPLAY TCR.TG%{TG_ID:(0|1)} bits have been set to a granule size %{REQUEST:d}K not implemented - using %{SUBSTITUTE:d}K. Fields:

REQUEST signed int

page size requested (or 0 for reserved).

SUBSTITUTE signed int

best guess available page size.

TG_ID bool

bits TG0 or TG1.

ArchMsg.Warning.warning_load_multiple_user_registers_from_user_mode

TAGS **UNPREDICTABLE** DISPLAY An LDM(user registers) instruction executed from user mode.

ArchMsg.Warning.warning_mte_in_mem_attr_not_supported

TAGS **UNPREDICTABLE** DISPLAY Memory attribute %{ATTR} has Tagged memory type but Armv8.5-MTE is not implemented. Fields:

ATTR unsigned int

Memory attribute.

ArchMsg.Warning.warning_reserved_breakpoint_state_match

TAGS **UNPREDICTABLE** DISPLAY %{IS_BREAKPOINT:(Watchpoint|Breakpoint)} programmed with a reserved combination of HMC, SSC and %{IS_BREAKPOINT:(PAC|PMC)}. Fields:

IS_BREAKPOINT unsigned int

Is a breakpoint, not a watchpoint.

ArchMsg.Warning.warning_shareability

TAGS **UNPREDICTABLE** DISPLAY Unpredictable: combination of the 1st and 2nd stages of translation is Normal Inner Non-Cacheable, Outer Non-Cacheable.

ArchMsg.Warning.warning_software_step_set_while_enabled

TAGS **UNPREDICTABLE** DISPLAY MDSCR_EL1.SS set to 1 while software step debug exceptions are enabled.

ArchMsg.Warning.warning_thumb_instruction_wraps_around_memory

TAGS UNPREDICTABLE DISPLAY Thumb instruction wraps around memory.

ArchMsg.Warning.warning_ttbr_sbz_bits_are_not_zero

TAGS UNPREDICTABLE DISPLAY %{TTBR}[%{MSB:d}:0] should be zero, but are not. Fields:

MSB unsigned int

x-1.

TTBR enum

which TTBR register.

ArchMsg.Warning.warning_unaligned_address_dbgdttrrx_write

TAGS UNPREDICTABLE DISPLAY Unaligned address %{ADDR} in X0/R0 when DBGDTTRRX written in memory access mode. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unaligned_address_dbgdttrtx_read

TAGS UNPREDICTABLE DISPLAY Unaligned address %{ADDR} in X0/R0 when DBGDTTRTX read in memory access mode. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unaligned_to_device

TAGS UNPREDICTABLE DISPLAY Unaligned Access to Device Memory %{ADDR}. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unaligned_to_strongly_ordered

TAGS UNPREDICTABLE DISPLAY Unaligned Access to Strongly Ordered Memory %{ADDR}. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unknown_sau_rnr

TAGS UNPREDICTABLE DISPLAY SAU_RNR was set to an unsupported value.

ArchMsg.Warning.warning_unpredictable_AIRCR_PRIS_and_BFHFNMINs

TAGS UNPREDICTABLE DISPLAY The effect of setting both AIRCR.BFHFNMINs and AIRCR.PRIS to 1 is UNPREDICTABLE.

ArchMsg.Warning.warning_unpredictable_AIRCR_VECTCLRACTIVE_when_not_in_debug

TAGS UNPREDICTABLE DISPLAY The effect of writing a 1 to AIRCR.VECTCLRACTIVE if the processor is not halted in Debug state is UNPREDICTABLE.

ArchMsg.Warning.warning_unpredictable_AIRCR_VECTRESET_and_SYSRESETREQ

TAGS **UNPREDICTABLE** DISPLAY When the processor is halted in Debug state, if a write to the register writes a 1 to both VECTRESET and SYSRESETREQ, the behavior is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_AIRCR_VECTRESET_when_not_in_debug

TAGS **UNPREDICTABLE** DISPLAY The effect of writing a 1 to AIRCR.VECTRESET if the processor is not halted in Debug state is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_AIRCR_incorrect_VKEY

TAGS **UNPREDICTABLE** DISPLAY The value 0x05FA must be written to AIRCR.VKEY, otherwise the register write is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_EXC_RETURN_Reserved_Bit

EXC_RETURN[23:7] are reserved with the special condition that all bits should be written as one. Values other than all 1s are **UNPREDICTABLE**. TAGS **UNPREDICTABLE**. Fields:

EXC_RETURN unsigned int

Special PC value.

SBO_MASK unsigned int

Which bits Should Be One to not UNPRED.

ArchMsg.Warning.warning_unpredictable_PMEVTYPEPn_ELO_VS_reserved_value

TAGS **UNPREDICTABLE** DISPLAY Unpredictable reserved value of PMEVTYPEN<n>_ELO.VS.

ArchMsg.Warning.warning_unpredictable_change_to_DEMCR_MON_STEP_at_insufficient_priority

TAGS **UNPREDICTABLE** DISPLAY The effect of changing DEMCR.MON_STEP at an execution priority that is lower than the priority of the DebugMonitor exception is **CONSTRAINED UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_change_to_priority_of_active_exception

TAGS **UNPREDICTABLE** DISPLAY Changing the priority of an active exception is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_clear_lspact

TAGS **UNPREDICTABLE** DISPLAY **UNPREDICTABLE** state on context switch. Fields:

DRVEXC_TAKEN unsigned int

Derived exception taken, not pended.

REACHED_LAST_STORE unsigned int

Only the last faulted.

ArchMsg.Warning.warning_unpredictable_dbg_exit_unaligned_dlr

TAGS **UNPREDICTABLE** DISPLAY Debug exit to AArch32 with DLR[0] set to 1. PC[0] can be set to 0 or to DLR[0].

ArchMsg.Warning.warning_unpredictable_exception_catch

TAGS **UNPREDICTABLE** DISPLAY Generation of an exception catch event is unpredictable.

ArchMsg.Warning.warning_unpredictable_exception_return_inconsistent_state

TAGS **UNPREDICTABLE** DISPLAY state on exception return is unpredictable.

ArchMsg.Warning.warning_unpredictable_exception_return_instruction

TAGS **UNPREDICTABLE** DISPLAY exception return instruction is unpredictable.

ArchMsg.Warning.warning_unpredictable_in_debug_state

TAGS **UNPREDICTABLE** DISPLAY Instruction is **UNPREDICTABLE** when executed in debug state.

ArchMsg.Warning.warning_unpredictable_mair_encoding

TAGS **UNPREDICTABLE** DISPLAY Memory attribute %{ATTR} has invalid MAIR encoding; this may lead to unpredictable results. Fields:

ATTR unsigned int

Memory attribute.

ArchMsg.Warning.warning_unpredictable_pmu_counter_access

TAGS **UNPREDICTABLE** DISPLAY Access to PMU counters from non-secure EL0 or EL1 is **UNPREDICTABLE** with MDCR_EL2.HPMN set to 0.

ArchMsg.Warning.warning_unpredictable_prioritization_breakpoint_match_vector_catch

TAGS **UNPREDICTABLE** DISPLAY Two events with the same priority occurred at the same instruction: Address Matching Vector Catch debug event, and Breakpoint debug event. It is constrained unpredictable which is taken.

ArchMsg.Warning.warning_unpredictable_stack_selection

TAGS **UNPREDICTABLE** DISPLAY **UNPREDICTABLE** state on context switch.

ArchMsg.Warning.warning_unpredictable_tsize_out_of_range

TAGS **UNPREDICTABLE** DISPLAY %{TCR}.T%{TTBR:d}SZ (%{TSIZE:d} bits) is out of range. Must be between 25 and %{TSIZE_MAX:d}. Fields:

TCR enum

Which TCR.

TSIZE signed int

Value of TSize.

TSIZE_MAX signed int

The maximum allowed value.

TTBR unsigned int

Which TTBR region.

ArchMsg.Warning.warning_unpredictable_unaligned_pc_as_base_register

TAGS **UNPREDICTABLE** DISPLAY The PC value must be word-aligned, otherwise the behavior of the instruction is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_unaligned_pop_stack

TAGS **UNPREDICTABLE** DISPLAY **CONSTRAINED UNPREDICTABLE** unaligned stack frame %{FRAME_ADDR:x} in exception-return. Fields:

FRAME_ADDR unsigned int

unaligned stack frame address.

ArchMsg.Warning.warning_unpredictable_unaligned_to_non_writeback

TAGS **UNPREDICTABLE** DISPLAY Unaligned Access to Non-Writeback-Cacheable Memory.

ArchMsg.Warning.warning_unpredictable_vmsa_memattrib

TAGS **UNPREDICTABLE** DISPLAY Unpredictable vmsa memory attribute with texcb value programmed to %{TEXCB} when tex remap is %{TEX_REMAP}. Fields:

TEXCB unsigned int

texcb value.

TEX_REMAP bool

use tex remap.

ArchMsg.Warning.warning_unpredictable_vtcr_t0sz_sl0

TAGS **UNPREDICTABLE** DISPLAY The combination of VTCR.TOSZ=%{TOSZ:d} and VTCR.SL0=%{SL0:d} is unpredictable. Fields:

SL0 signed int

VTCR.SL0, starting level for stage 2 translation table walks.

TOSZ signed int

VTCR.TOSZ, required input address range.

ArchMsg.Warning.warning_unpredictable_write_DHCSR

TAGS **UNPREDICTABLE** DISPLAY **UNPREDICTABLE** set of bit-changes in DHCSR. Fields:

NEW unsigned int

data attempted to be written.

OLD unsigned int

previous value.

ArchMsg.Warning.warning_unpredictable_write_SHCSR

TAGS **UNPREDICTABLE** DISPLAY **UNPREDICTABLE** setting a pending bit to 1 for an exception with priority >= execution priority.

ArchMsg.Warning.warning_unsupported_access_to_memory_mapped_register

TAGS **UNPREDICTABLE** DISPLAY Access to memory mapped register is unsupported.

ArchMsg.Warning.warning_user_jmcr_access

TAGS **UNPREDICTABLE** DISPLAY User %{WRITE:(write|read)} access to JMCR is unpredictable. Fields:

WRITE unsigned int

access is write.

ArchMsg.Warning.warning_wcr_mask_and_bas

TAGS **UNPREDICTABLE** DISPLAY DBGWCR_EL1.MASK is non-zero and BAS is not 0xff.

ArchMsg.Warning.warning_wcr_mask_reserved

TAGS **UNPREDICTABLE** DISPLAY DBGWCR_ELn.MASK is set to a reserved value.

ArchMsg.Warning.warning_wcr_non_configuous_bas

TAGS **UNPREDICTABLE** DISPLAY DBGWCRn_EL1.BAS has non-contiguous set of ones.

ArchMsg.Warning.warning_wvr_masked_bit_not_zero

TAGS **UNPREDICTABLE** DISPLAY DBGWVRn_EL1.VA contains a bit masked by DBGWCRn_EL1.MASK that is not zero.

ArchMsg.Why.why_branch_target_exception

A branch target exception has occurred. DISPLAY branch target exception: target pc=%{PC}, target opcode=%{OPCODE}. PRIMARY KEY PC. Fields:

BTYPE unsigned int

pstate.BTYPE at time branch taken.

OPCODE unsigned int

opcode at branch target.

PC unsigned int

Virtual Address of branch target.

ArchMsg.Why.why_illegal_state

An illegal mode change has occurred. DISPLAY illegal mode change: %{MESSAGE}. Fields:

MESSAGE string

The message.

BRANCH_MISPREDICT

Simulating branch mispredict. Fields:

PC unsigned int

Origin address (or 0 if unavailable).

BRA_DIR

Direct branches, to immediate address. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISSET enum

The instruction set of the branch instruction.

IS_COND bool

Indicates if this is a conditional branch.

IS_HINTED bool

Whether the conditional instruction is hinted.

IS_ISB bool

Whether the instruction is ISB.

IS_LINK bool

Indicates if this is a branch with link.

PC unsigned int

The address of the branch instruction.

TARGET_ISET enum

The instruction set after the branch.

TARGET_PC unsigned int

The address the instruction branches to.

BRA_INDIR

Indirect branches, perhaps to a register. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISET enum

The instruction set of the branch instruction.

IS_COND bool

Indicates if this is a conditional branch.

IS_HINTED bool

Whether the conditional instruction is hinted.

IS_ISB bool

Whether the instruction is ISB.

IS_LINK bool

Indicates if this is a branch with link.

PC unsigned int

The address of the branch instruction.

TARGET_ISET enum

The instruction set after the branch.

TARGET_PC unsigned int

The address the instruction branches to.

BRA_INDIR_CCFAIL

Indirect branches fail, perhaps to a register.

BRA_RET_CCFAIL

Branch return fail, perhaps to a register.

CACHE_MAINTENANCE_OP

Cache Maintenance Operation. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

DATA unsigned int

Specified MVA or Set/way.

FUNCTION enum

Clean / Invalidate.

SCOPE enum

Affected region.

SIDE enum

Inst / Data.

CCFAIL

Conditional instruction condition check fail. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CCFAIL_UNC

Unconditional instruction condition check fail. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CCPASS

Conditional instruction condition check pass. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CCPASS_UNC

Unconditional instruction condition check pass. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CODE_CACHE_MAINT

Code cache maintenance. Fields:

END_INCL unsigned int

Inclusive end address for VA or PA requests.

NS bool

Non-secure world for PA.

START unsigned int

Start address for VA or PA requests.

TYPE enum

Request type.

COMPILE_BLOCK_END

Last instruction of basic block translated. Fields:

VADDR unsigned int

Address of next instruction after this basic block.

COMPILE_BLOCK_START

First instruction of basic block translated. Fields:

VADDR unsigned int

Address of first instruction in basic block.

COMPILE_INST

ARM instruction compiled. Fields:

DISASS string

Disassembly of the instruction.

ISSET enum

The instruction set of this instruction.

ITSTATE unsigned int

The ITSTATE current for the instruction.

OPCODE unsigned int

The opcode of the instruction.

PC unsigned int

The address of the instruction.

SIZE unsigned int

The size of the instruction in bytes.

CONTEXTIDR

Context ID Register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

NS bool

Secure or nonsecure banked register is accessed.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The new value written.

CONTEXT_SYNC

Called for every context synchronization event. Fields:

ARCH_CSE bool

Architectural Context Synchronization Event.

PC unsigned int

The address of the synchronization instruction.

SYSREG_SYNC bool

System register synchronization event.

CORE_ENDIAN

Core BE8 Big-Endian state changed. Fields:

BE8 bool

Core BE8 Big-Endian state.

CORE_INFO

Static processor attributes. Only triggered by a call to DumpState(). Fields:

ARCH_PROFILE enum

The architecture profile of the core.

CLUSTER_ID unsigned int

The cluster ID of this processor.

CORE_NUM unsigned int

The number of this core in an MP processor.

DETAILED_CORE_FEATURES enum

Detailed core features as a bitfield. Enum values are bitmasks to extract the info from the field.

FPU_VERSION enum

The VFP version implemented by the core.

MEM_ARCH enum

The memory architecture of the core.

MIDR_EL1 unsigned int

MIDR_EL1 value.

NUM_CORES unsigned int

The number of cores in this MP processor.

QUANTUM_SIZE unsigned int

The default quantum size of the core (in ticks).

SECURITY_FEATURES bool

Does the core have security features?.

CORE_LOADS

Processor load accesses. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ACQREL enum

Is this an acquire/release.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DATA unsigned int

The data read or written.

ELEMENT_SIZE unsigned int

Width of each element.

ENDIANNESS enum

Byte ordering of this access.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

LOCK enum

Normal, exclusive or locked access.

MECID unsigned int

Memory Encryption Context ID, if applicable.

MECID2 unsigned int

Memory Encryption Context ID of the second page, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PADDR_TAG unsigned int

If tag checked, the physical address tag for the access.

PAGE_SIZE unsigned int

Page size as $\log_2(\text{size in bytes})$.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK, 2=Exclusive Failed, 3=Unsupported.

SIZE unsigned int

Width of the access in bytes. Only required if DATA is not traced.

TAG_CHECKED bool

Is this access tag checked.

TAG_CHECKED2 bool

Is the second page access tag checked.

TRANS bool

Is this a translated access.

VADDR unsigned int

The virtual address of the access.

CORE_REGS

Changes of the core registers R0 to R14. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number, 0 to 14.

MODE enum

Bank of the register accessed.

OLD_VALUE unsigned int

The old value overwritten.

PHYS_ID enum

The physical register accessed.

VALUE unsigned int

The new value written to the register.

CORE_REGS64

Changes of the core registers X0..X30, SP_ELn. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ID enum

The register number, 0-30 for X0-X30, >=32 for SP_ELn.

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

CORE_REGS64_READ

General purpose AArch64 register has been read X0..X30, SP_ELn. Fields:

ID enum

The register number, 0-30 for X0-X30, >=32 for SP_ELn.

VALUE unsigned int

The value read from the register.

CORE_STALLED_TRANSITION

Detect when a core is prevented from progressing PChannel states. Fields:

CURRENT_STATE enum

Current power mode of core.

REASON string

The reason a transition was stalled, and what must be done to unblock it.

TARGET_STATE enum

Desired power mode of core after transition.

CORE_STORES

Processor store accesses. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ACQREL enum

Is this an acquire/release.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DATA unsigned int

The data read or written.

ELEMENT_SIZE unsigned int

Width of each element.

ENDIANNESS enum

Byte ordering of this access.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

LOCK enum

Normal, exclusive or locked access.

MECID unsigned int

Memory Encryption Context ID, if applicable.

MECID2 unsigned int

Memory Encryption Context ID of the second page, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PADDR_TAG unsigned int

If tag checked, the physical address tag for the access.

PAGE_SIZE unsigned int

Page size as $\log_2(\text{size in bytes})$.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK, 2=Exclusive Failed, 3=Unsupported.

SIZE unsigned int

Width of the access in bytes. Only required if DATA is not traced.

TAG_CHECKED bool

Is this access tag checked.

TAG_CHECKED2 bool

Is the second page access tag checked.

TRANS bool

Is this a translated access.

VADDR unsigned int

The virtual address of the access.

CP14_READ

System Coprocessor register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP14 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc1 unsigned int

opcode 1.

opc2 unsigned int

opcode 2.

CP14_WRITE

System Coprocessor register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP14 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc1 unsigned int

opcode 1.

opc2 unsigned int

opcode 2.

CP15_READ

System Coprocessor register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP15 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc1 unsigned int

opcode 1.

opc2 unsigned int

opcode 2.

CP15_READ64

System Coprocessor register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP15 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc unsigned int

opcode 1.

CP15_WRITE

System Control Coprocessor register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP15 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc1 unsigned int

opcode 1.

opc2 unsigned int

opcode 2.

CP15_WRITE64

System Coprocessor register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP15 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc unsigned int

opcode 1.

CPSR

CPSR change. (In AArch64 PSTATE represented in SPSR format). Fields:

CORE_NUM unsigned int

Core number in a multi processor.

OLD_VALUE unsigned int

The old CPSR value.

UNKNOWN unsigned int

Bits within the register that have unknown value.

VALUE unsigned int

The new CPSR value.

CRYPTO_SPEC

Every crypto instruction speculatively executed.

CorePowerStateChange

Core power-up state has changed. DISPLAY Core powered %{PU:(up|down)}. Fields:

PU bool

Whether the core powered up.

DATA_CACHE_SET_ALLOC_TAGS

Set allocation tags from DC GVA instruction. Fields:

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

NSDESC unsigned int

The security state of the access.

PADDR unsigned int

The physical address of the access.

PAS enum

The physical address space of the page.

RESPONSE enum

0=Aborted, 1=OK.

SIZE unsigned int

The size of memory in bytes.

TAG unsigned int

The value of the tag being set.

TAGGING_TYPE enum

Type of Memory tagging (Physical or Virtual).

TAG_VADDR unsigned int

The virtual address of the Virtual Tag if Virtual tagging is used; otherwise 0.

VADDR unsigned int

Virtual address used for setting allocation tags.

DATA_CACHE_ZERO

Zero and invalidate cache line from DC ZVA instruction. Fields:

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

NSDESC unsigned int

The security state of the access.

PADDR unsigned int

The physical address of the access.

PADDR_TAG unsigned int

If tag checked, the physical address tag for the access.

PAS enum

The physical address space of the page.

RESPONSE enum

0=Aborted, 1=OK.

SIZE unsigned int

The size of zero'd memory in bytes.

TAG_CHECKED bool

Is this access tag checked.

VADDR unsigned int

Virtual address of the zero'd cache line.

DEBUG_EVENT

Hardware debug support event. Fields:

EVENT enum

Description of event.

VALUE unsigned int

data value.

DEBUG_TARGET_ISET

Hardware debug target ISA. Fields:

TARGET_ISET enum

The target instruction set of the debug state.

DMI_ALLOCATE

DMI allocated. Fields:

ACCESS_DIRECTION enum

READ / WRITE.

PADDR unsigned int

The physical address of the DMI.

PTR unsigned int

DMI range pointer.

SIDE enum

DSIDE / ISIDE.

SIZE unsigned int

The size of DMI.

VADDR unsigned int

The virtual address of the DMI..

DMI_HIT

DMI hit, access used DMI (only fire for small DMI regions i.e. less than a full page). Fields:

ACCESS_DIRECTION enum

READ / WRITE.

PADDR unsigned int

The physical address of the DMI.

SIDE enum

DSIDE / ISIDE.

SIZE unsigned int

The size of the DMI.

VADDR unsigned int

The virtual address of the DMI..

DMI_REVOKE

DMI revoked. Fields:

ACCESS_DIRECTION enum

READ / WRITE.

PADDR unsigned int

The physical address of the DMI.

SIDE enum

DSIDE / ISIDE.

SIZE unsigned int

The size of DMI.

VADDR unsigned int

The virtual address of the DMI..

END_COMPILE

Compilation end. Fields:

END_OF_PAGE_COUNT unsigned int

Number of basic blocks exited due to page boundary since START_COMPILE.

FETCHFAIL_COUNT unsigned int

Number of basic blocks exited due to ifetch failure START_COMPILE.

INST_COUNT unsigned int

Number of instructions compiled since START_COMPILE.

NONSEQ_COUNT unsigned int

Number of basic blocks exited due to non-sequential instructions since START_COMPILE.

PAGE_STRADDLE_COUNT unsigned int

Number of basic blocks exited due to unaligned instructions crossing page since START_COMPILE.

ESB

Error Synchronization Barrier hit. Fields:

ASYNC bool

ESB is async.

IMPLICIT bool

ESB is implicit.

EXCEPTION

Exception taken. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ESR unsigned int

The value assigned to ESR when exception is taken.

ESR_EC unsigned int

The Value assigned to ESR.EC when exception is taken.

IS_PHYSICAL bool

Physical or Virtual exception.

IS_TRAP bool

Exception coming from a trap.

LR unsigned int

The value assigned to the link register.

PC unsigned int

The location where the exception occurred.

PREFERRED_RETURN unsigned int

The preferred return address for the exception.

TAKEN_LOCALLY bool

Whether exception is taken locally(same exception level).

TARGET_ISET enum

The instruction set of the exception handler code.

TARGET_PC unsigned int

The address the exception branches to.

VECTOR enum

The exception vector.

VECTOR_OFFSET unsigned int

The offset of an exception vector from the base address.

EXCEPTION_END

Every exception completed.

EXCEPTION_INFO

Trace human-readable reason of the current exception for data/instruction aborts, floating point exceptions and unknown exceptions. Fields:

DESCRIPTION string

Reason for current exception.

ESR_EC unsigned int

The value of ESR.EC when exception is taken.

EXCEPTION_RAISE

Every exception raised.

EXCEPTION_RETURN

Branches on leaving exception. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISSET enum

The instruction set of the branch instruction.

IS_COND bool

Indicates if this is a conditional branch.

IS_HINTED bool

Whether the conditional instruction is hinted.

IS_ISB bool

Whether the instruction is ISB.

IS_LINK bool

Indicates if this is a branch with link.

PC unsigned int

The address of the branch instruction.

TARGET_ISET enum

The instruction set after the branch.

TARGET_PC unsigned int

The address the instruction branches to.

EXCEPTION_RETURN_PREBRANCH

Exception return event that must occur before PC and processor state have updated.

EXCEPTION_START

Every exception started.

FIQ_TAKEN

FIQ taken exception.

FP_EXCEPTIONS_SIGNALLED

Trace which FP exceptions were signalled as a part of the FP operations performed by an instruction. Fields:

FP_EXCEPTIONS_SIGNALLED_MASK unsigned int

Indicates which FP exceptions were signalled by an instruction. This uses the same encoding as the cumulative floating-point exception bitfields in FPSR.

FP_STATE

Floating point state. Fields:

FPSR unsigned int

FPSR register value.

MASK unsigned int

The mask indicating updated cumulative bits.

FREQ_CHANGED

Core frequency changed. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The instruction count of this core.

NEW_FREQ signed int

The new frequency of this core (expressed in Hz).

OLD_FREQ signed int

The old frequency of this core (expressed in Hz).

HLT

HLT instruction occurred. Fields:

IMM16 unsigned int

The 16-bit immediate value encoded in HLT instruction.

INST

Every instruction executed. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DEBUG_STATE bool

The instruction is executed in debug state.

DISASS string

Disassembly of instruction.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISET enum

The current instruction set.

ITSTATE unsigned int

The current ITSTATE.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

MODE enum

The mode the core is in.

NS unsigned int

The core's non-secure bit.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

OPCODE unsigned int

The opcode of the instruction.

PADDR unsigned int

The physical address of the instruction.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the instruction.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the page.

PC unsigned int

The address of the instruction.

SECURITY_STATE enum

The core's security state.

SIZE unsigned int

The size of the instruction in bytes.

INST_END

Every instruction completed.

INST_START

Every instruction started. Fields:

DEBUG_STATE bool

The instruction is executed in debug state.

ISET enum

The current instruction set.

MODE enum

The mode the core is in.

NS enum

The current Secure State.

PC unsigned int

The address of the conditional instruction.

SECURITY_STATE enum

The current Security State.

INST_STRADDLE

Instruction straddles boundary. Fields:

DEBUG_STATE bool

The instruction is executed in debug state.

ISET enum

The current instruction set.

MODE enum

The mode the core is in.

NS enum

The current Secure State.

PC unsigned int

The address of the conditional instruction.

SECURITY_STATE enum

The current Security State.

IRQ_TAKEN

IRQ taken exception.

ISB

ISB instruction occurred. Fields:

PC unsigned int

The address of the ISB instruction.

LOAD_MULTIPLE_REGS_LIST

List of destination registers (not source) of a multiple registers load instruction. Fields:

REG string

Register to add into the list.

LOCAL_MONITOR

Local monitor activity. Fields:

PADDR unsigned int

Local Monitor Address.

State enum

State of the monitor (Open/Exclusive).

MAPPING_CREATE

A memory mapping has been created. This is an implementation detail of the model and so may change without notice. Fields:

ASID unsigned int

The ASID used during translation, or 0 if no ASID was used.

IS_VALID bool

Whether the mapping is valid or not.

PADDR unsigned int

Base physical address of region, or 0 if not known.

SIDE enum

I-side or D-side.

SIZE unsigned int

Size of the page, or 0 if not known.

VADDR unsigned int

Base virtual address of region.

VMID unsigned int

The VMID used during translation, or 0 if no VMID was used.

MAPPING_DESTROY

A memory mapping has been destroyed. This is an implementation detail of the model and so may change without notice. Fields:

ASID unsigned int

The ASID used during translation.

CAUSE enum

The cause of invalidation.

PADDR unsigned int

Base physical address of page.

SIDE enum

I-side or D-side.

SIZE unsigned int

Size of the page.

VADDR unsigned int

Base virtual address of page.

VMID unsigned int

The VMID used during translation.

MEMMAP_DEBUG_READ

Memory mapped reads to the debug registers. Fields:

ADDR unsigned int

address.

CORE_NUM unsigned int

Core number in a multi processor.

EXT bool

Whether access is from an external device (such as the DAP).

NS enum

Secure state of the access.

REG_NAME string

Name of the debug register accessed.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

MEMMAP_DEBUG_WRITE

Memory mapped writes to the debug registers. Fields:

ADDR unsigned int

address.

CORE_NUM unsigned int

Core number in a multi processor.

EXT bool

Whether access is from an external device (such as the DAP).

NS enum

Secure state of the access.

REG_NAME string

Name of the debug register accessed.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The new value written.

MEMTAG_LOADS

Allocation tag memory read; when attributes and system configuration allow tag accesses.

Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

NUM_GRANULES unsigned int

Number of tags to be Loaded/stored.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK.

TAGGING_TYPE enum

Type of Memory tagging (Physical or Virtual).

TAGS unsigned int

The tags read or written. Tags for subsequent granules would be found at subsequent bytes in this field (One byte for each tag).

TAG_VADDR unsigned int

The virtual address of the Virtual Tag if Virtual tagging is used; otherwise 0.

VADDR unsigned int

The virtual address of the access.

MEMTAG_LOAD_INST

Allocation tag load instruction; regardless of tags being actually accessed. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

NUM_GRANULES unsigned int

Number of tags to be Loaded/stored.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK.

TAGGING_TYPE enum

Type of Memory tagging (Physical or Virtual).

TAGS unsigned int

The tags read or written. Tags for subsequent granules would be found at subsequent bytes in this field (One byte for each tag).

TAG_VADDR unsigned int

The virtual address of the Virtual Tag if Virtual tagging is used; otherwise 0.

VADDR unsigned int

The virtual address of the access.

MEMTAG_STORES

Allocation tag memory write; when attributes and system configuration allow tag accesses.
Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

NUM_GRANULES unsigned int

Number of tags to be Loaded/stored.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK.

TAGGING_TYPE enum

Type of Memory tagging (Physical or Virtual).

TAGS unsigned int

The tags read or written. Tags for subsequent granules would be found at subsequent bytes in this field (One byte for each tag).

TAG_VADDR unsigned int

The virtual address of the Virtual Tag if Virtual tagging is used; otherwise 0.

VADDR unsigned int

The virtual address of the access.

MENTAG_STORE_INST

Allocation tag store instruction; regardless of tags being actually accessed. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

NUM_GRANULES unsigned int

Number of tags to be Loaded/stored.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK.

TAGGING_TYPE enum

Type of Memory tagging (Physical or Virtual).

TAGS unsigned int

The tags read or written. Tags for subsequent granules would be found at subsequent bytes in this field (One byte for each tag).

TAG_VADDR unsigned int

The virtual address of the Virtual Tag if Virtual tagging is used; otherwise 0.

VADDR unsigned int

The virtual address of the access.

MMU_D128_TTB_READ

This event is triggered by reads caused by a D128 translation table walk. Fields:

ABORTED bool

Was the walk successful.

DATA_HIGH unsigned int

The data upper bits read.

DATA_LOW unsigned int

The data lower bits read.

IPA unsigned int

For stage 1, the IPA of the read.

LEVEL unsigned int

Translation table level.

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

OUTADDR unsigned int

The output physical address.

PADDR unsigned int

The physical address of the read.

PASmem enum

Physical Address Space of the accessed memory.

PASreq enum

Physical Address Space of the requested lookup.

REGIME_EL enum

Entry matches in this translation regime.

REQUESTER enum

The requester of this table walk.

SIDE enum

Inst / Data.

STAGE unsigned int

Translation stage.

MMU_D128_TTB_WRITE

This event is triggered by writes caused by a D128 translation table walk. Fields:

ABORTED bool

Was the walk successful.

DATA_HIGH unsigned int

The data upper bits written.

DATA_LOW unsigned int

The data lower bits written.

IPA unsigned int

For stage 1, the IPA of the write.

IS_HACDBS_ACCESS bool

If the access was due to HACDBS_CLEAN.

LEVEL unsigned int

Translation table level.

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

OUTADDR unsigned int

The output physical address.

PADDR unsigned int

The physical address of the write.

PASmem enum

Physical Address Space of the accessed memory.

PASreq enum

Physical Address Space of the requested lookup.

REGIME_EL enum

Entry matches in this translation regime.

SIDE enum

Inst / Data.

STAGE unsigned int

Translation stage.

MMU_TRANS

Address translation information. Fields:

ASID unsigned int

Address space identifier.

CORE_NUM unsigned int

Core number in a multi processor.

Hyp bool

Entry matches in Hyp state only.

INNERCACHE_RA bool

Is the inner cache allocate on read.

INNERCACHE_TRANSIENT bool

Is the inner write-through transient.

INNERCACHE_TYPE enum

Inner Caching scheme (NC/MB/WA).

INNERCACHE_WA bool

Is the inner cache allocate on write.

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MEMTYPE enum

Memory type.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC enum

Is secure side supposed to access secure or nonsecure memory.

OUTERCACHE_RA bool

Is the outer cache allocate on read.

OUTERCACHE_TRANSIENT bool

Is the outer write-through transient.

OUTERCACHE_TYPE enum

Outer Caching scheme (NC/MB/WA).

OUTERCACHE_WA bool

Is the outer cache allocate on write.

PADDR unsigned int

Physical address of the access.

PAGESIZE unsigned int

Page size as $\log_2(\text{size})$.

PAS enum

PAS supposed to be accessed is secure or nonsecure memory or Root/Realm if RME is implemented.

SH enum

Shareability.

SIDE enum

Inst / Data.

STAGE1_PERM unsigned int

Stage 1 permissions, [5:3] Privileged access (XWR) [2:0] User access (XWR).

STAGE2_PERM unsigned int

Stage 2 permission mask [5:3] Privileged access (XWR) [2:0] User access (XWR).

VADDR unsigned int

Virtual address of the access.

VMID unsigned int

Virtual machine identifier.

XS bool

XS Attribute.

nG enum

Flag indicating whether ASID will be matched.

MMU_TTB_READ

This event is triggered by reads caused by a translation table walk. Fields:

ABORTED bool

Was the walk successful.

DATA unsigned int

The data read.

DESC_KIND enum

The encoding scheme of the descriptors used for this walk.

IPA unsigned int

For stage 1, the IPA of the read.

LEVEL unsigned int

Translation table level.

LPAE bool

Is this for an LPAE translation (*DEPRECATED*, use DESC_KIND).

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSmem bool

Non secure memory access.

NSreq bool

Non secure request.

OUTADDR unsigned int

The output physical address.

PADDR unsigned int

The physical address of the read.

PASmem enum

Physical Address Space of the accessed memory.

PASreq enum

Physical Address Space of the requested lookup.

REGIME_EL enum

Entry matches in this translation regime.

REQUESTER enum

The requester of this table walk.

SIDE enum

Inst / Data.

STAGE unsigned int

Translation stage.

MMU_TTB_WRITE

This event is triggered by writes caused by a translation table walk. Fields:

ABORTED bool

Was the walk successful.

DATA unsigned int

The data written.

DESC_KIND enum

The encoding scheme of the descriptors used for this walk.

IPA unsigned int

For stage 1, the IPA of the write.

IS_HACDBS_ACCESS bool

If the access was due to HACDBS_CLEAN.

LEVEL unsigned int

Translation table level.

LPAE bool

Is this for an LPAE translation (*DEPRECATED*, use DESC_KIND).

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSmem bool

Non secure memory access.

NSreq bool

Non secure request.

OUTADDR unsigned int

The output physical address.

PADDR unsigned int

The physical address of the write.

PASmem enum

Physical Address Space of the accessed memory.

PASreq enum

Physical Address Space of the requested lookup.

REGIME_EL enum

Entry matches in this translation regime.

SIDE enum

Inst / Data.

STAGE unsigned int

Translation stage.

MODE_CHANGE

Mode change. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

MODE enum

The new mode.

NON_SECURE enum

The core's new non-secure bit.

OLD_MODE enum

The old mode.

SECURITY_STATE enum

The new security state.

MOPS_LD_RETIRED

Load instructions architecturally executed for MOPS.

MOPS_ST_RETIRED

Store instructions architecturally executed for MOPS. Fields:

IS_MEMCPY bool

Whether the memory operation is MEMCPY.

NOT_TAG_CHECKED_INST

MTE Tag not checked for current instruction.

PERIODIC

Called for every quantum. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The instruction count of this CPU.

PC unsigned int

The address of the next instruction to be executed on this CPU.

PMU_COUNTER_OVERFLOW

PMU counter overflow. Fields:

COUNTER_GROUP enum

Counter group.

COUNTER_RANGE enum

Counter range.

EVENT_ID unsigned int

Event Identifier, field is valid only for Event counter group.

INDEX unsigned int

Counter index (as selected by PMSELR), field is valid only for Event/Cycle counter groups.

INTERRUPT bool

Is interrupt enabled on overflow for this counter.

POINTER_AUTH

Called when execution hits a pointer authentication check. Fields:

AUTH_STATE bool

Authentication result.

KEY_USED enum

Pointer authentication key.

POTINDEX_UPDATE

Value of POT index for current pc changed. Fields:

NEW_POTINDEX unsigned int

New POTIndex.

OLD_POTINDEX unsigned int

Old POTIndex.

PREFETCH_MEMORY64

Prefetch from PRFM or PRFUM instructions. Fields:

PRFOP enum

Prefetch hint.

VADDR unsigned int

Virtual address of the location that should be prefetched.

PRELOAD64_REQUEST

Preload requests from PRFM or PRFUM instructions. Fields:

PRFOP enum

Prefetch hint.

VADDR unsigned int

Virtual address of the location that should be prefetched.

PRELOAD_DATA

Data preload from PLD instruction. Fields:

VADDR unsigned int

Virtual address of the data that should be preloaded.

PRELOAD_INST

Instruction preload from PLI instruction. Fields:

VADDR unsigned int

Virtual address of the instruction that should be preloaded.

PRE_CORE_LOAD

Trace just before a core load. Fields:

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

PRE_CORE_STORE

Trace just before a core store. Fields:

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

PRE_TTB_READ

This event is triggered before reads caused by a translation table walk. Fields:

ADDR unsigned int

The physical address of the read.

PSTATE

PSTATE fields update. Fields:

ALLINT unsigned int

All interrupt mask bit.

BTYPE unsigned int

Branch target identification bit.

CORE_NUM unsigned int

Core number in a multi processor.

DAIF unsigned int

Exception mask bits.

DIT unsigned int

Data Independent Timing bit.

EL unsigned int

Current Exception level field.

EXLOCK unsigned int

Exception return state lock bit.

IL unsigned int

Illegal Execution state bit.

NZCV unsigned int

Condition flags.

PACM unsigned int

PACM bit.

PAN unsigned int

Privileged Access Never state bit.

PM unsigned int

PMU exception mask bit.

PPEND unsigned int

PMU exception pending bit.

SM unsigned int

StreamingMode status bit.

SP unsigned int

Stack pointer register selection bit.

SS unsigned int

Software Step bit.

SSBS unsigned int

Speculative Store Bypass Safe bit.

TCO unsigned int

Tag Check Override bit.

TOUCHED unsigned int

Mask of the fields that have been touched since this trace has last been emitted. Note that touched does not necessarily mean the value changed. The index of the field is the bit index in the mask plus the index of this field plus one.

UAO unsigned int

User Access Override bit.

UINJ unsigned int

Undefined Instruction exceptions bit.

ZA unsigned int

ZA storage and ZT0 register enable bit.

nRW unsigned int

Current Execution state bit.

PSTATE_PACTIVE_OPMODE_TRANSITION

Trace pactive and Opmode mode transitions on PChannel. Fields:

Copyright © 2023–2025 Arm Limited (or its affiliates). All rights reserved.

Non-Confidential

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Operating mode after the transition.

OLD_STATE enum

Operating mode before the transition.

PSTATE_PACTIVE_OPMODE_TRANSITION_DENIED

Trace the denied pactive and Opmode mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Operating mode after the transition.

OLD_STATE enum

Operating mode before the transition.

REASON string

Reason for the transition being denied.

PSTATE_PACTIVE_POWER_TRANSITION

Trace pactive and power mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Power mode after the transition.

OLD_STATE enum

Power mode before the transition.

PSTATE_PACTIVE_POWER_TRANSITION_DENIED

Trace the denied pactive and power mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Power mode after the transition.

OLD_STATE enum

Power mode before the transition.

REASON string

Reason for the transition being denied.

PSTATE_PACTIVE_THREAD_OPMODE_TRANSITION

Trace pactive and Opmode mode transitions for threads on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Operating mode after the transition.

OLD_STATE enum

Operating mode before the transition.

QUANTUM_START

Called at the start of every quantum.

RANGE_PRELOAD_REQUEST

Preload requests from RPRFM. Fields:

COUNT unsigned int

Number of blocks of data to access.

LENGTH unsigned int

Number of contiguous bytes to be accessed in each block.

OP unsigned int

Range prefetch operation .

REUSE unsigned int

Maximum number of bytes to be accessed before next RPRFM .

STRIDE signed int

Stride to advance after accessing LENGTH bytes.

VADDR unsigned int

Base virtual address of range prefetch request.

RUN_STATE

Run state transition. Fields:

INST_COUNT unsigned int

Ticks count at point of transition.

NEW enum

New run state.

OLD enum

Old run state.

SEMIHOSTING_CALL

Call of a semihost function occurred. Fields:

PC unsigned int

The program counter after the semihosting call.

SEMIHOSTING_PRECALL

About to call semihost. Fields:

EL2 bool

Use EL2 translation regime.

NS bool

Is Non-Secure.

REG_WIDTH unsigned int

The current register width in bytes.

SIGNAL

External signal state change. Fields:

SIGNAL enum

Signal that changed.

STATE bool

Signal asserted state.

SOFTWARE_STEP

Return the Debug Software Step state. Fields:

STATE enum

Software Step state.

SPSR

SPSR change. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

MODE enum

Which of the banked SPSR registers is written.

OLD_VALUE unsigned int

The old SPSR value.

VALUE unsigned int

The new SPSR value.

START_COMPILE

Compilation started. Fields:

VADDR unsigned int

Instruction where compilation begins.

STG_MULTIPLE_MICRO_OPS_ADDRESS_RANGE

Address range for MTE stores with multiple uops. Fields:

ADDR_END unsigned int

Final address.

ADDR_START unsigned int

Starting address.

STORE_MULTIPLE_REGS_ADDRESS_RANGE

Address range for stores with multiple registers. Fields:

ADDR_END unsigned int

Final address.

ADDR_START unsigned int

Starting address.

SUBOPTIMAL_LDST_RETIRED

suboptimal load/store. Fields:

ACCESS_SIZE unsigned int

Log2 of the access width in bytes used for alignment checking (1: half-word, 2: word ...).

OPTIMAL_ALIGN_SELECTOR unsigned int

Selector to decide the optimal alignment size (1: access width, 2: 4-bytes, 3: 8-bytes, ..., 12: 4-kbytes).

VADDR unsigned int

The virtual address of the access.

SVE_INST_SPEC

SVE operations speculatively executed.

SVE_LD_RETIRED

SVE load instructions architecturally executed.

SVE_LOADS

SVE load accesses. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ACQREL enum

Is this an acquire/release.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DATA unsigned int

The data read or written.

ELEMENT_SIZE unsigned int

Width of each element.

ENDIANNESS enum

Byte ordering of this access.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

LOCK enum

Normal, exclusive or locked access.

MECID unsigned int

Memory Encryption Context ID, if applicable.

MECID2 unsigned int

Memory Encryption Context ID of the second page, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PADDR_TAG unsigned int

If tag checked, the physical address tag for the access.

PAGE_SIZE unsigned int

Page size as $\log_2(\text{size in bytes})$.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK, 2=Exclusive Failed, 3=Unsupported.

SIZE unsigned int

Width of the access in bytes. Only required if DATA is not traced.

TAG_CHECKED bool

Is this access tag checked.

TAG_CHECKED2 bool

Is the second page access tag checked.

TRANS bool

Is this a translated access.

VADDR unsigned int

The virtual address of the access.

SVE_MEM_PREFETCH

Scalable Vector Extension memory prefetch. Fields:

CACHE_LEVEL enum

The preferred cache level.

HINT enum

What the memory is expected to be used for.

SIZE unsigned int

The prefetch size in bytes.

TEMPORAL bool

Whether the data is temporal.

VADDR unsigned int

The virtual address of the prefetch.

SVE_REG_READ

Scalable Vector Extension register read. Fields:

ID enum

The register identifier.

SM bool

Whether the PE is in Streaming Mode.

VALUE unsigned int

The read data.

SVE_REG_UPDATE

Scalable Vector Extension register update. Fields:

ID enum

The register identifier.

SM bool

Whether the PE is in Streaming Mode.

VALUE unsigned int

The written data.

SVE_REG_WRITE

Scalable Vector Extension register write. Fields:

ID enum

The register identifier.

SM bool

Whether the PE is in Streaming Mode.

VALUE unsigned int

The written data.

SVE_STORES

SVE store accesses. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ACQREL enum

Is this an acquire/release.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DATA unsigned int

The data read or written.

ELEMENT_SIZE unsigned int

Width of each element.

ENDIANNESS enum

Byte ordering of this access.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

LOCK enum

Normal, exclusive or locked access.

MECID unsigned int

Memory Encryption Context ID, if applicable.

MECID2 unsigned int

Memory Encryption Context ID of the second page, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PADDR_TAG unsigned int

If tag checked, the physical address tag for the access.

PAGE_SIZE unsigned int

Page size as $\log_2(\text{size in bytes})$.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK, 2=Exclusive Failed, 3=Unsupported.

SIZE unsigned int

Width of the access in bytes. Only required if DATA is not traced.

TAG_CHECKED bool

Is this access tag checked.

TAG_CHECKED2 bool

Is the second page access tag checked.

TRANS bool

Is this a translated access.

VADDR unsigned int

The virtual address of the access.

SVE_ST_RETIRED

SVE store instructions architecturally executed.

SVE_UNALIGNED_LDST_RETIRED

SVE load or store instructions architecturally executed that would cause alignment fault if A bit is set.

SVE_Z_PRED_MEM_ACCESS_INFO

Information about SVE/SME predicated accesses to Z registers (non Gather/Scatter). Fields:

MEM_ELEM_SIZE unsigned int

Width of elements accessed in memory (in bytes).

RnW bool

Whether this is a Read not a Write.

VA unsigned int

Virtual base address computed for accesses.

SYNC

Called for every synchronization, that is every quantum break. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The instruction count of this CPU.

LOCAL_QUANTUM unsigned int

The local quantum of this CPU.

LOCAL_TIME unsigned int

The local time of this CPU.

SYNC_REASON enum

Reason for sync.

SYSCALL

System call instruction executed. Fields:

IMM unsigned int

Immediate value of the system call instruction.

TYPE enum

System call type.

VADDR unsigned int

Instruction that caused the system call.

SYSREG_READ128

System Coprocessor 128 bit register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

REGNUM enum

Internal register number.

REGNUM_PHYS enum

Internal register number physically accessed (e.g. in presense of VHE redirection).

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_READ64

System Coprocessor register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REGNUM enum

Internal register number.

REGNUM_OPERAND enum

Register number corresponding to the instruction operand.

REGNUM_PHYS enum

Internal register number physically accessed (e.g. in presense of VHE redirection).

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_UPDATE128

Triggers when the system updates a register. Fields:

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG enum

Register number.

UNKNOWN unsigned int

Bits of the register which became unknown.

VALUE unsigned int

Value written to the register.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_UPDATE32

Triggers when the system updates a register. Fields:

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG enum

Register number.

UNKNOWN unsigned int

Bits of the register which became unknown.

VALUE unsigned int

Value written to the register.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_UPDATE64

Triggers when the system updates a register. Fields:

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG enum

Register number.

UNKNOWN unsigned int

Bits of the register which became unknown.

VALUE unsigned int

Value written to the register.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_WRITE128

System Coprocessor 128 bit register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

REGNUM enum

Internal register number.

REGNUM_PHYS enum

Internal register number physically accessed (e.g. in presense of VHE redirection).

UNDEF bool

The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_WRITE64

System Coprocessor register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REGNUM enum

Internal register number.

REGNUM_OPERAND enum

Register number corresponding to the instruction operand.

REGNUM_PHYS enum

Internal register number physically accessed (e.g. in presense of VHE redirection).

UNDEF bool

The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

TRBU_BUFFER_WRAP

DISPLAY Trace buffer current write pointer wrapped.

TRBU_TRIGGER_EVENT

DISPLAY Trace buffer Trigger Event.

TRBU_WRITE

DISPLAY PETU trace data has been written to the TRBU memory buffer at %{BUFFER_ADDRESS} with size %{SIZE} and data %{DATA}. Fields:

BUFFER_ADDRESS unsigned int

Virtual Address of start of buffer write.

DATA unsigned int

Data written to the buffer.

SIZE unsigned int

Size of sample written to buffer, in bytes.

UNALIGNED_LDST_RETIRE

Processor unaligned load/store. Fields:

ACCESS_SIZE unsigned int

Log2 of the access width in bytes used for alignment checking (1=>half-word, 2->word ...).

NUMBER_OF_BEATS unsigned int

Number of accesses (beats) in this burst (Total size in bytes is ACCESS_SIZE * NUMBER_OF_BEATS).

VADDR unsigned int

The virtual address of the access.

VFP_D_REGS

VFP/NEON D 64 bit register write. Fields:

ALIASING enum

Type of register aliasing used.

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a write to the corresponding bit in the register. For example, 0x1 would indicate a write to VALUE[0].

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

VFP_D_REGS_READ

VFP/NEON D 64 bit register read. Fields:

ID unsigned int

The register number.

VALUE unsigned int

The current value of the register.

VFP_Q_REGS

VFP/NEON Q 128 bit register write. Fields:

ALIASING enum

Type of register aliasing used.

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a write to the corresponding byte in the register. For example, 0x1 would indicate a write to VALUE[7:0].

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

VFP_Q_REGS_READ

VFP/NEON Q 128 bit register read. Fields:

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a read to the corresponding byte in the register. For example, 0x1 would indicate a read to VALUE[7:0].

VALUE unsigned int

The current value of the register.

VFP_SYS_REGS

Writes to the VFP/NEON units system registers. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ID enum

Which VFP system register is written.

MASK unsigned int

The mask indicating updated cumulative bits.

OLD_VALUE unsigned int

The register's old value overwritten.

VALUE unsigned int

The new value written to the VFP system register.

VFP_S_REGS

VFP/NEON S 32 bit register write. Fields:

ALIASING enum

Type of register aliasing used.

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a write to the corresponding bit in the register. For example, 0x1 would indicate a write to VALUE[0].

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

VFP_S_REGS_READ

VFP/NEON S 32 bit register read. Fields:

ID unsigned int

The register number.

VALUE unsigned int

The current value of the register.

VIRTUAL_TAG_ALLOCATED

Virtual tag information when allocated. Fields:

TAG_VADDR unsigned int

Virtual tag virtual address.

VADDR unsigned int

Data Virtual address.

VIRTUAL_TAG unsigned int

Virtual tag value.

WAYPOINT

Signals end of basic block execution. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ISET enum

Origin instruction set.

IS_COND bool

Indicates if this is a conditional waypoint.

PC unsigned int

Origin address (or 0 if unavailable).

TAKEN bool

Indicates if this waypoint was taken.

TARGET unsigned int

Destination address.

TARGET_ISET enum

Destination instruction set.

WFE_END

WFE ended. Fields:

INST_COUNT unsigned int

Ticks count when leaving WFE.

WFE_EVENT_REGISTER

WFE event register status: set/clear, reason. Fields:

INST_COUNT unsigned int

Ticks count.

REASON enum

Reason for set/clear. Only REASON==0 (Cleared by WFE) clears the bit, all other reasons set it.

WFE_IGNORED

WFE ignored. Fields:

EVENT bool

This WFE was ignored because the event register was set.

INST_COUNT unsigned int

Ticks count when ignoring WFE.

TRAPPED bool

This WFE was trapped.

WFE_START

WFE entered. Fields:

INST_COUNT unsigned int

Ticks count when entering WFE.

WFI_END

WFI ended. Fields:

INST_COUNT unsigned int

Ticks count when leaving WFI.

WFI_IGNORED

WFI ignored. Fields:

DISABLED bool

This WFI was ignored because WFI is disabled.

INST_COUNT unsigned int

Ticks count when ignoring WFI.

REASON enum

specifies reason why WFI trace was ignored.

TRAPPED bool

This WFI was trapped.

WFI_START

WFI entered. Fields:

INST_COUNT unsigned int

Ticks count when entering WFI.

WFI_WAKEUP

WFI wakeup. Fields:

INST_COUNT unsigned int

Ticks count when WFI wakeup occurred.

REASON enum

Reason for wakeup.

2.9 ARM_C1-Premium

This section describes the trace sources.

AA64_ASE_SVE_REGS

Modification of AArch64 Advanced SIMD or SVE Vector Registers (via SIMD or Scalar operations), SVE Predicate Registers and ZT0. Fields:

ID enum

The register identifier.

MASK unsigned int

Bitmask of the register to signal the modified bits in the VALUE field.

SM bool

Whether the PE is in Streaming Mode.

VALUE unsigned int

Value of AArch64 vector register.

ASYNC_MEMORY_FAULT

Context ID Register write. Fields:

FAULT unsigned int

Fault status present in ESR.ISS encoding for Serror in ARCH64 and present in DFSR encoding for external abort format in ARCH32.

PADDR unsigned int

Physical Address (or 0 if unavailable).

VADDR signed int

Virtual Address (or 0 if unavailable).

ATOMIC_END_ACCESS

Bus trace access for atomics. Fields:

ACCESS_FAIL bool

Memory access failed.

ADDR unsigned int

The virtual address of the access.

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

COMPARE_VALUE unsigned int

Compare value for CAS.

LOAD_VALUE unsigned int

Loaded values.

NSDESC unsigned int

The security state of the access.

OPERAND_VALUE unsigned int

Operation's operand.

OPERATION enum

Operation type.

PADDR unsigned int

The physical address of the access.

PRIV bool

Is this a privileged access?.

ATOMIC_START_ACCESS

Bus trace access for atomics. Fields:

ADDR unsigned int

The virtual address of the access.

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

COMPARE_VALUE unsigned int

Compare value for CAS.

NSDESC unsigned int

The security state of the access.

OPERAND_VALUE unsigned int

Operation's operand.

OPERATION enum

Operation type.

PADDR unsigned int

The physical address of the access.

PRIV bool

Is this a privileged access?.

ArchMsg.Error.exit_code

DISPLAY exit code: %{EXITCODE} component kind: %{KIND} reason: %{REASON}. Fields:

EXITCODE signed int

Exit code to be returned.

KIND string

Component kind that invoked the exit code trace.

REASON string

Reason why the exit code trace was invoked (optional).

ArchMsg.Warning.IgnoredWarmResetRequest

Event fires when software attempts to request warm reset through write to deprecated DBGPRCR.CWRR.

ArchMsg.Warning.RAS_ErrselrUnpredictableState

DISPLAY ERRSELR is in unpredictable state due to value greater than number_of_error_records written to it.

ArchMsg.Warning.RAS_UnpredictableStateWrite

DISPLAY %{REG_NAME} write ignored due to unpredictable ERRSELR state. Fields:

REG_NAME string

Name of register being written.

ArchMsg.Warning.ReportedRevisionRegisterRead

DISPLAY Reported revision and patch do not match that of the cpu. Fields:

ORIGINAL_VALUE unsigned int

The value the register would have based on the modelled revision.

REGISTER string

The name of the register that has ben read.

REPORTED_VALUE unsigned int

The value the register has been overridden to.

ArchMsg.Warning.Unpredictably Indexed PM Event Register

TAGS **UNPREDICTABLE** DISPLAY Event %{IsDirect:(Selector value|Register)} %{SEL} is >= %{N}. Fields:

IsDirect unsigned int

Direct Access rather than indirect.

N unsigned int

Number of Accessible Registers.

SEL unsigned int

Selector.

ArchMsg.Warning.cache_contents_unknown

A cache has been activated without being invalidated and may contain **UNKNOWN** content. DISPLAY %{SIDE:(D|I)}-Cache was enabled in %{REGIME:(SECURE|NON_SECURE|HYP)} regime but was not invalidated since power-on: cache lines could contain **UNKNOWN** content %{SIDE:(Note: D-side caches must invalidate SECURE lines even if the cache is only used by NON_SECURE code in order to ensure that unexpected dirty lines tagged SECURE are not naturally evicted)}. Fields:

REGIME unsigned int

Current translation regime.

SIDE unsigned int

0 d-side cache or unified cache, 1 i-side.

ArchMsg.Warning.dcimvac_matches_watchpoint

TAGS **UNPREDICTABLE** DISPLAY Watchpoints matching an AArch32 DCIMVA are implementation defined.

ArchMsg.Warning.decode_fieldmismatch

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with different fields %{FIELD1} and %{FIELD2}. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_fracbitsnegative

TAGS **UNPREDICTABLE** DISPLAY VCVT Instruction is unpredictable with a negative number of fraction bits.

ArchMsg.Warning.decode_initblock

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable in an IT block.

ArchMsg.Warning.decode_initblocknotlast

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable in an IT block when not the last.

ArchMsg.Warning.decode_invalid_condition

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable when the condition is not AL.

ArchMsg.Warning.decode_invalidfieldcombination

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with the values in %{FIELD1} and %{FIELD2}. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_invalidvalue

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with the value in %{FIELD}. Fields:

FIELD enum

Field name.

ArchMsg.Warning.decode_mem_hint_unallocated

TAGS **UNPREDICTABLE** DISPLAY There is no memory hint allocated to this bit pattern.

ArchMsg.Warning.decode_registermatch

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with %{FIELD1} and %{FIELD2} as the same register. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_registermismatch

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with %{FIELD1} and %{FIELD2} as different registers. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_registeroutofrange

TAGS **UNPREDICTABLE** DISPLAY Floating-point instruction is unpredictable indexing beyond the end of the register bank.

ArchMsg.Warning.decode_sbzsbo

TAGS **UNPREDICTABLE** DISPLAY Reserved bits in the instruction are not canonical.

ArchMsg.Warning.decode_transactiontoobig

TAGS **UNPREDICTABLE** DISPLAY Attempt to perform a memory transaction of more than 128 bytes.

ArchMsg.Warning.decode_unpred_other

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable.

ArchMsg.Warning.decode_unpreduseofpc

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable when %{FIELD} is PC. Fields:

FIELD enum

Field name.

ArchMsg.Warning.decode_unpreduseofr13

TAGS UNPREDICTABLE DISPLAY Instruction is unpredictable when %{FIELD} is R13. Fields:

FIELD enum

Field name.

ArchMsg.Warning.decode_unpreduseofr13inreglist

TAGS UNPREDICTABLE DISPLAY Instruction is unpredictable with R13 in register-list.

ArchMsg.Warning.decode_useofd16orhigher

TAGS UNPREDICTABLE DISPLAY Attempt to use register D16 or higher where it is not allowed.

ArchMsg.Warning.decode_writebackandbaseinlist

TAGS UNPREDICTABLE DISPLAY Instruction is unpredictable with write back when the base register is in the transfer list.

ArchMsg.Warning.decode_zeroregistersinlist

TAGS UNPREDICTABLE DISPLAY Instruction is unpredictable when no registers are to be transferred.

ArchMsg.Warning.misaligned_ptrs

DISPLAY TRBPTR_EL1(%{TRBPTR_EL1}) or TRBTRG_EL1(%{TRBTRG_EL1}) are not aligned as per TRBIDR_EL1.Align(%{TRBIDR_EL1}). Fields:

TRBIDR_EL1 unsigned int

TRBIDR_EL1.Align value.

TRBPTR_EL1 unsigned int

Current write pointer.

TRBTRG_EL1 unsigned int

Trigger pointer.

ArchMsg.Warning.out_of_range_ptrs

DISPLAY TRBPTR_EL1(%{TRBPTR_EL1}) is out of range as per TRBBASER_EL1(%{TRBBASER_EL1}) or TRBLIMITR_EL1.LIMIT(%{TRBLIMITR_EL1}). Fields:

TRBBASER_EL1 unsigned int

Base pointer.

TRBLIMITR_EL1 unsigned int

Limit pointer.

TRBPTR_EL1 unsigned int

Current write pointer.

ArchMsg.Warning.recursive_branch

An instruction is performing a branch that targets the same instruction. If this was intended then a WFI might be more effective. DISPLAY recursive branch detected at PC=%{PC}.

PRIMARY KEY PC. Fields:

PC unsigned int

Address of instruction causing the branch.

ArchMsg.Warning.recursive_exception

An instruction has generated an exception that targets the same instruction. DISPLAY recursive %{TYPE} exception detected at PC=%{PC} (CPSR=%{CPSR}). PRIMARY KEY PC. Fields:

CPSR unsigned int

Processor state.

PC unsigned int

Address of instruction causing the exception.

TYPE enum

Type of exception.

ArchMsg.Warning.reentrant_vector_catch

TAGS **UNPREDICTABLE** DISPLAY Vector catch debug event on Prefetch or Data abort vector %{ADDR}. Fields:

ADDR unsigned int

vector address.

ArchMsg.Warning.reserved_it_state

TAGS **UNPREDICTABLE** DISPLAY Execution with reserved IT state %{ITSTATE}. Fields:

ITSTATE unsigned int

reserved IT state.

ArchMsg.Warning.tlb_contents_unknown

Translation has been enabled without invalidating the TLB. DISPLAY Translation was enabled in the %{PAS:(Secure|Non-secure|Root|Realm)} EL%{EL:d} translation regime (vmid=%{VMID:x}) without invalidating %{INVALIDITY:(|d-side |i-side |)}entries since power-on: %{INVALIDITY:(|D||I)}TLB could contain **UNKNOWN** content. Fields:

EL unsigned int

Exception level of translation regime.

INVALIDITY unsigned int

Which TLB may be invalid (if TLBs are separate).

NON_SECURE bool

Whether the translation regime is non-secure.

PAS unsigned int

Physical address space of translation regime.

VMID unsigned int

Current VMID, if applicable.

ArchMsg.Warning.unknown_DLR_DSPSR

Exiting debug state with unknown DLR or DSPSR.

ArchMsg.Warning.unknown_ELR_SPSR

Returning from debug exception with unknown ELR or SPSR.

ArchMsg.Warning.unpred_FAR_on_LRCPC3_non_atomic_mem_access

TAGS **UNPREDICTABLE** Unpredictable FAR/FSTATUS on LRCPC3 pair instructions resulting in two separate memory accesses
DISPLAY Unpredictable FAR on LRCPC3 non atomic memory access at %{ADDR1} with FSTATUS %{FSTATUS1} or %{ADDR2} with FSTATUS %{FSTATUS2}. Fields:

ADDR1 unsigned int

Possible FAR address 1.

ADDR2 unsigned int

Possible FAR address 2.

FSTATUS1 unsigned int

Possible fault status value 1.

FSTATUS2 unsigned int

Possible fault status value 2.

ArchMsg.Warning.unpred_async_tag_check_fail_on_sve_speculative_load_do_update_ffr

Unpredictable set to FALSE of the FFR element associated to SVE speculative load with an async tag check fail TAGS **UNPREDICTABLE**.

ArchMsg.Warning.unpredictable_a32_breakpoint

TAGS **UNPREDICTABLE** DISPLAY Unpredictable breakpoint on a A32 instruction at %{ADDR1} Offset %{OFFSET} BAS %{BAS} for a %{IS_ADDRESS_MISMATCH_BREAKPOINT:(Mismatch|Match)} Breakpoint number %{N:d} Breakpoint address %{ADDR2}. Fields:

ADDR1 unsigned int

Address.

ADDR2 unsigned int

Breakpoint address.

BAS unsigned int

BAS field.

IS_ADDRESS_MISMATCH_BREAKPOINT bool

Is a Mismatch breakpoint, not a Match breakpoint.

N unsigned int

Breakpoint number.

OFFSET unsigned int

Always has a value of zero.

ArchMsg.Warning.unpredictable_banked_register_access

TAGS **UNPREDICTABLE** DISPLAY Access to banked register (R=%{R:d} SYSm=%{SYSM:2x}) is unpredictable. Fields:

R unsigned int

R bit.

SYSM unsigned int

SYSm.

ArchMsg.Warning.unpredictable_ccfail_undef

TAGS **UNPREDICTABLE** DISPLAY An instruction %{OPCODE} at VA=%{VADDR} failed its condition codes check but would otherwise have caused a non-data-dependent trap or undefined exception. Behaviour is implementation-defined choice of exception or nop. Fields:

OPCODE unsigned int

opcode of the instruction.

VADDR unsigned int

Virtual address of the instruction being executed.

ArchMsg.Warning.unpredictable_ici_bits

TAGS **UNPREDICTABLE** DISPLAY ICI bits do not apply or are architecturally unpredictable for instruction.

ArchMsg.Warning.unpredictable_t32_breakpoint

TAGS **UNPREDICTABLE** DISPLAY Unpredictable breakpoint on a T32 instruction at %{ADDR1} Offset %{OFFSET} BAS %{BAS} for a %{IS_ADDRESS_MISMATCH_BREAKPOINT:(Mismatch|Match)} Breakpoint number %{N:d} Breakpoint address %{ADDR2}. Fields:

ADDR1 unsigned int

Address.

ADDR2 unsigned int

Breakpoint address.

BAS unsigned int

BAS field.

IS_ADDRESS_MISMATCH_BREAKPOINT bool

Is a Mismatch breakpoint, not a Match breakpoint.

N unsigned int

Breakpoint number.

OFFSET unsigned int

Indicates which halfword of a 32-bit Thumb instruction the breakpoint is set on. 0 for the first halfword or if the instruction is 16 bits wide, or 1 for the second halfword.

ArchMsg.Warning.unpredictable_t32_vector_catch

TAGS **UNPREDICTABLE** DISPLAY Unpredictable Vector catch exception on a T32 instruction at %{ADDR} Offset %{OFFSET}. Fields:

ADDR unsigned int

Address.

OFFSET unsigned int

Indicates which halfword of a 32-bit Thumb instruction the vector catch exception is set on. 0 for the first halfword or if the instruction is 16 bits wide, or 1 for the second halfword.

ArchMsg.Warning.unpredictable_watchpoint_far

TAGS **UNPREDICTABLE** DISPLAY The value for FAR/EDWAR for hitting this watchpoint may be anywhere in the range %{LOWER_BOUND} to %{UPPER_BOUND}. Fields:

LOWER_BOUND unsigned int

the lowest address accessed by the instruction that triggered the watchpoint.

UPPER_BOUND unsigned int

the highest watchpointed address accessed by that instruction.

ArchMsg.Warning.unpredictable_watchpoint_far_isscmp

DISPLAY The value for FAR/EDWAR for hitting this watchpoint may be anywhere in the range %{LOWER_BOUND} to %{UPPER_BOUND}. Fields:

LOWER_BOUND unsigned int

the lowest address accessed by the instruction that triggered the watchpoint.

UPPER_BOUND unsigned int

the highest watchpointed address accessed by that instruction.

ArchMsg.Warning.warning_AdvSIMDExpandImmUnexpectedZero

TAGS **UNPREDICTABLE** DISPLAY AdvSIMDExpandImm may treat this immediate value as **UNPREDICTABLE**.

ArchMsg.Warning.warning_ConditionalSMC

TAGS **UNPREDICTABLE** DISPLAY SMC instruction has **UNPREDICTABLE** effects when conditional and when combined with traps.

ArchMsg.Warning.warning_TLBID_out_of_range

TLBID out of range in TLBI according to the configuration of NIS, NVIS, NOS and NVOS. Fields:

NIS unsigned int

NIS.

NOS unsigned int

NOS.

NVIS unsigned int

NVIS.

NVOS unsigned int

NVOS.

TLBID unsigned int

The TLBID value used in the TLBI instruction.

ArchMsg.Warning.warning_TTL_or_TG_mismatches_current_translation_granule

TTL or TG mismatches current translation granule in TLBI VA/RVA/IPA/RIPA. Fields:

CURRENT_TG unsigned int

Current TG as indicated by TCR_ELx or VTCR_EL2.

TLBI_TG unsigned int

TG (or TTL[3:2]) used in TLBI instruction.

ArchMsg.Warning.warning_access_crosses_page_boundary

TAGS **UNPREDICTABLE** DISPLAY Access crosses page boundary.

ArchMsg.Warning.warning_access_crosses_page_boundary_has_fault

TAGS **UNPREDICTABLE** DISPLAY Access that crosses page boundary has fault(s).

ArchMsg.Warning.warning_access_crosses_page_boundary_has_fault_on_both_pages

TAGS **UNPREDICTABLE** DISPLAY Access that crosses page boundary has a fault on each page.
Fields:

ADDR1 unsigned int

FAR address on first page.

ADDR2 unsigned int

FAR address on second page.

FSTATUS1 unsigned int

Fault status on first page.

FSTATUS2 unsigned int

Fault status on second page.

ArchMsg.Warning.warning_access_crosses_page_boundary_spanning_different_memory

TAGS **UNPREDICTABLE** DISPLAY Access at address %{ADDR} crosses page from MEM_TYPE %{MEMTYPE_PAGE1} to %{MEMTYPE_PAGE2}. Fields:

ADDR unsigned int

address of access crossing page.

MEMTYPE_PAGE1 signed int

First Page's memory type (0-Normal, 1-Non-Normal).

MEMTYPE_PAGE2 signed int

Second Page's memory type (0-Normal, 1-Non-Normal).

ArchMsg.Warning.warning_access_wraps_around_memory

TAGS **UNPREDICTABLE** DISPLAY Access wraps around memory in %{WIDTH:d}bit mode.
Fields:

WIDTH unsigned int

Address width.

ArchMsg.Warning.warning_bcr_linking_status

TAGS **UNPREDICTABLE** DISPLAY DBGBCR.{BT2,BT} combination gives unpredictable behaviour.

ArchMsg.Warning.warning_bcr_mask_reserved

TAGS **UNPREDICTABLE** DISPLAY DBGBCR_ELn.MASK is set to a reserved value.

ArchMsg.Warning.warning_bvr_masked_bit_not_zero

TAGS **UNPREDICTABLE** DISPLAY DBGBVRn_EL1.VA contains a bit masked by DBGBCRn_EL1.MASK that is not zero.

ArchMsg.Warning.warning_ccsidr_unimplemented_level

TAGS **UNPREDICTABLE** DISPLAY CCSIDR read while CSSELR %{CSSELR:x} points to an unimplemented cache level. Fields:

CSSELR unsigned int

current effective value of CSSELR.

ArchMsg.Warning.warning_change_to_ns_when_tge_set

TAGS **UNPREDICTABLE** DISPLAY Attempting to change NS to 1 when HCR.TGE = 1.

ArchMsg.Warning.warning_contiguous_bit_check_abort

TAGS **UNPREDICTABLE** DISPLAY An abort occurred whilst attempting to check TLB entry is contiguous at %{ENTRY_ADDR}. Fields:

ENTRY_ADDR unsigned int

address of conflicting TLB entry.

ArchMsg.Warning.warning_contiguous_bit_error

TAGS **UNPREDICTABLE** DISPLAY TLB entry at %{ENTRY_ADDR} was not contiguous with entry at %{CONTIG_ADDR} Contents are %{ENTRY_DATA} but expected %{CONTIG_DATA}. Fields:

CONTIG_ADDR unsigned int

address of the TLB first read, with which this entry is expected to be contiguous.

CONTIG_DATA unsigned int

expected contents based on the entry at CONTIG_ADDR.

ENTRY_ADDR unsigned int

address of conflicting TLB entry.

ENTRY_DATA unsigned int

contents of conflicting TLB entry.

ArchMsg.Warning.warning_contiguous_bit_error_D128

TAGS **UNPREDICTABLE** DISPLAY TLB entry at %{ENTRY_ADDR} was not contiguous with entry at %{CONTIG_ADDR} Contents are %{ENTRY_DATA_HI} %{ENTRY_DATA_LO} but expected %{CONTIG_DATA}. Fields:

CONTIG_ADDR unsigned int

address of the TLB first read, with which this entry is expected to be contiguous.

CONTIG_DATA unsigned int

expected contents based on the entry at CONTIG_ADDR.

ENTRY_ADDR unsigned int

address of conflicting TLB entry.

ENTRY_DATA_HI unsigned int

Higher bit contents of conflicting TLB entry.

ENTRY_DATA_LO unsigned int

Lower bit contents of conflicting TLB entry.

ArchMsg.Warning.warning_cp10_cp11_mismatch

TAGS **UNPREDICTABLE** DISPLAY Unpredictable configuration of CP10 and CP11 controls at EL
%{EL:d}. Fields:

EL unsigned int

EL owning the control.

ArchMsg.Warning.warning_cp10_cp11_reserved_value

TAGS **UNPREDICTABLE** DISPLAY Unpredictable configuration of CPACR CP10 (%{CP10}) or
CP11 (%{CP11}). Fields:

CP10 unsigned int

CP10 access.

CP11 unsigned int

CP11 access.

ArchMsg.Warning.warning_csselr_level_out_of_range

TAGS **UNPREDICTABLE** DISPLAY CSSELR.Level out of range, written value %{LEVEL} with only
%{IMPLEMENTED:d} levels implemented. Fields:

IMPLEMENTED unsigned int

number of cache levels implemented.

LEVEL unsigned int

written value of CSSELR.Level.

TYPE enum

The type of cache selected by CSSELR.InD and TnD.

ArchMsg.Warning.warning_debug_flow_control_bits_not_obeyed

TAGS **UNPREDICTABLE** DISPLAY Flow control bits for %{REGISTER} were not obeyed, causing
%{ERROR}. Fields:

ERROR enum

type of error resulting.

REGISTER enum

register under consideration.

ArchMsg.Warning.warning_debug_register_access_during_reset

TAGS **UNPREDICTABLE** DISPLAY Unpredictable %{IS_WRITE:(read|write)} access to debug register offset %{OFFSET} during reset. Fields:

IS_WRITE unsigned int

Write Not Read.

OFFSET unsigned int

Register Offset.

ArchMsg.Warning.warning_decode_cps_inconsistent_fields

TAGS **UNPREDICTABLE** DISPLAY The mode setting fields in the CPS instruction are inconsistent.

ArchMsg.Warning.warning_decode_invalid_state

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable in current exception-level/security state.

ArchMsg.Warning.warning_default_cacheable_mmu_on

TAGS **UNPREDICTABLE** DISPLAY Default cacheable is enabled (HCR.DC = 1) while MMU is enabled.

ArchMsg.Warning.warning_default_cacheable_vmmu_off

TAGS **UNPREDICTABLE** DISPLAY Default cacheable is enabled (HCR.DC = 1) while HCR.VM = 0.

ArchMsg.Warning.warning_deprecated_wvr_bit_2_set

TAGS **UNPREDICTABLE** DISPLAY DBGWVR%{N:d}_EL1=%{WVR:h} has bit 2 set, deprecated in ARMv8. Fields:

N unsigned int

index.

WVR unsigned int

DBGWVR.

ArchMsg.Warning.warning_eret_while_software_step_active_pending

Missing ISB between setting MDSCR_EL1.SS and ERET TAGS **UNPREDICTABLE**.

ArchMsg.Warning.warning_exclusive_to_non_normal

TAGS **UNPREDICTABLE** DISPLAY Exclusive Access to Non-Normal Memory %{ADDR}. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_exclusive_to_non_writeback

TAGS **UNPREDICTABLE** DISPLAY Exclusive Access to Non-Writeback-Cacheable Memory %{ADDR}. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_execute_from_device_memory

TAGS **UNPREDICTABLE** DISPLAY An attempt was made to execute from Device memory.

ArchMsg.Warning.warning_illegal_cpsr_mode

TAGS **UNPREDICTABLE** DISPLAY Writing an illegal or unimplement mode (%{NEW_MODE:h}) to CPSR was unpredictable. Fields:

NEW_MODE unsigned int

New value of CPSR.M.

ArchMsg.Warning.warning_illegal_srs_mode

TAGS **UNPREDICTABLE** DISPLAY Illegal or **UNPREDICTABLE** mode (%{MODE:h}) used for SRS instruction. Fields:

MODE unsigned int

mode for Banked SP.

ArchMsg.Warning.warning_implementation_defined_read_debug_register_in_SCS

Read Access to Debug registers from the processor is **IMPLEMENTATION DEFINED** TAGS **IMP_DEF**. Fields:

PPB_OFFSET unsigned int

debug register offset.

ArchMsg.Warning.warning_implementation_defined_sequential_security_transitions_supported

The behavior of sequential instruction fetches that cross from non-secure to secure memory and contain an SG instruction is **CONSTRAINED UNPREDICTABLE** TAGS **UNPREDICTABLE**.

ArchMsg.Warning.warning_implementation_defined_stack_limit_check_supported

It is **IMPLEMENTATION DEFINED** whether stack pointer limit checking is performed for this instruction TAGS **IMP_DEF**. Fields:

ADDRESS unsigned int

Address of instruction.

OPCODE unsigned int

instruction opcode.

ArchMsg.Warning.warning_implementation_defined_taggednonsharedmemorytype

TAGS **IMP_DEF** DISPLAY Memory attribute has Tagged, non shared memory type; it is **IMPLEMENTATION DEFINED** whether the memory location is treated as Tagged or non-Tagged.

ArchMsg.Warning.warning_implementation_defined_write_debug_register_in_SCS

Write Access to Debug registers from the processor is **IMPLEMENTATION DEFINED** TAGS **IMP_DEF**. Fields:

DATA unsigned int

data attempted to be written.

PPB_OFFSET unsigned int

debug register offset.

ArchMsg.Warning.warning_invalid_tcr_granule

TAGS UNPREDICTABLE DISPLAY TCR.TG{%TG_ID:(0|1)} bits have been set to a granule size {%REQUEST:d}K not implemented - using {%SUBSTITUTE:d}K. Fields:

REQUEST signed int

page size requested (or 0 for reserved).

SUBSTITUTE signed int

best guess available page size.

TG_ID bool

bits TG0 or TG1.

ArchMsg.Warning.warning_load_multiple_user_registers_from_user_mode

TAGS UNPREDICTABLE DISPLAY An LDM(user registers) instruction executed from user mode.

ArchMsg.Warning.warning_mte_in_mem_attr_not_supported

TAGS UNPREDICTABLE DISPLAY Memory attribute {%ATTR} has Tagged memory type but Armv8.5-MTE is not implemented. Fields:

ATTR unsigned int

Memory attribute.

ArchMsg.Warning.warning_reserved_breakpoint_state_match

TAGS UNPREDICTABLE DISPLAY {%IS_BREAKPOINT:(Watchpoint|Breakpoint)} programmed with a reserved combination of HMC, SSC and {%IS_BREAKPOINT:(PAC|PMC)}. Fields:

IS_BREAKPOINT unsigned int

Is a breakpoint, not a watchpoint.

ArchMsg.Warning.warning_shareability

TAGS UNPREDICTABLE DISPLAY Unpredictable: combination of the 1st and 2nd stages of translation is Normal Inner Non-Cacheable, Outer Non-Cacheable.

ArchMsg.Warning.warning_software_step_set_while_enabled

TAGS UNPREDICTABLE DISPLAY MDSCR_EL1.SS set to 1 while software step debug exceptions are enabled.

ArchMsg.Warning.warning_thumb_instruction_wraps_around_memory

TAGS UNPREDICTABLE DISPLAY Thumb instruction wraps around memory.

ArchMsg.Warning.warning_ttbr_sbz_bits_are_not_zero

TAGS UNPREDICTABLE DISPLAY {%TTBR}[{%MSB:d}:0] should be zero, but are not. Fields:

MSB unsigned int

x-1.

TTBR enum

which TTBR register.

ArchMsg.Warning.warning_unaligned_address_dbgdtrrx_write

TAGS **UNPREDICTABLE** DISPLAY Unaligned address %{ADDR} in X0/R0 when DBGDTRRX written in memory access mode. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unaligned_address_dbgdtrtx_read

TAGS **UNPREDICTABLE** DISPLAY Unaligned address %{ADDR} in X0/R0 when DBGDTRTX read in memory access mode. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unaligned_to_device

TAGS **UNPREDICTABLE** DISPLAY Unaligned Access to Device Memory %{ADDR}. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unaligned_to_strongly_ordered

TAGS **UNPREDICTABLE** DISPLAY Unaligned Access to Strongly Ordered Memory %{ADDR}. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unknown_sau_rnr

TAGS **UNPREDICTABLE** DISPLAY SAU_RNR was set to an unsupported value.

ArchMsg.Warning.warning_unpredictable_AIRCR_PRIS_and_BFHFNMINs

TAGS **UNPREDICTABLE** DISPLAY The effect of setting both AIRCR.BFHFNMINs and AIRCR.PRIS to 1 is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_AIRCR_VECTCLRACTIVE_when_not_in_debug

TAGS **UNPREDICTABLE** DISPLAY The effect of writing a 1 to AIRCR.VECTCLRACTIVE if the processor is not halted in Debug state is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_AIRCR_VECTRESET_and_SYSRESETREQ

TAGS **UNPREDICTABLE** DISPLAY When the processor is halted in Debug state, if a write to the register writes a 1 to both VECTRESET and SYSRESETREQ, the behavior is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_AIRCR_VECTRESET_when_not_in_debug

TAGS **UNPREDICTABLE** DISPLAY The effect of writing a 1 to AIRCR.VECTRESET if the processor is not halted in Debug state is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_AIRCR_incorrect_VKEY

TAGS **UNPREDICTABLE** DISPLAY The value 0x05FA must be written to AIRCR.VKEY, otherwise the register write is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_EXC_RETURN_Reserved_Bit

EXC_RETURN[23:7] are reserved with the special condition that all bits should be written as one. Values other than all 1s are **UNPREDICTABLE**. TAGS **UNPREDICTABLE**. Fields:

EXC_RETURN unsigned int

Special PC value.

SBO_MASK unsigned int

Which bits Should Be One to not UNPRED.

ArchMsg.Warning.warning_unpredictable_PMEVTYPEPn_ELO_VS_reserved_value

TAGS **UNPREDICTABLE** DISPLAY Unpredictable reserved value of PMEVTYPEN<n>_ELO.VS.

ArchMsg.Warning.warning_unpredictable_change_to_DEMCR_MON_STEP_at_insufficient_priority

TAGS **UNPREDICTABLE** DISPLAY The effect of changing DEMCR.MON_STEP at an execution priority that is lower than the priority of the DebugMonitor exception is **CONSTRAINED UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_change_to_priority_of_active_exception

TAGS **UNPREDICTABLE** DISPLAY Changing the priority of an active exception is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_clear_lspact

TAGS **UNPREDICTABLE** DISPLAY **UNPREDICTABLE** state on context switch. Fields:

DRVEXC_TAKEN unsigned int

Derived exception taken, not pended.

REACHED_LAST_STORE unsigned int

Only the last faulted.

ArchMsg.Warning.warning_unpredictable_dbg_exit_unaligned_dlr

TAGS **UNPREDICTABLE** DISPLAY Debug exit to AArch32 with DLR[0] set to 1. PC[0] can be set to 0 or to DLR[0].

ArchMsg.Warning.warning_unpredictable_exception_catch

TAGS **UNPREDICTABLE** DISPLAY Generation of an exception catch event is unpredictable.

ArchMsg.Warning.warning_unpredictable_exception_return_inconsistent_state

TAGS **UNPREDICTABLE** DISPLAY state on exception return is unpredictable.

ArchMsg.Warning.warning_unpredictable_exception_return_instruction

TAGS **UNPREDICTABLE** DISPLAY exception return instruction is unpredictable.

ArchMsg.Warning.warning_unpredictable_in_debug_state

TAGS **UNPREDICTABLE** DISPLAY Instruction is **UNPREDICTABLE** when executed in debug state.

ArchMsg.Warning.warning_unpredictable_mair_encoding

TAGS **UNPREDICTABLE** DISPLAY Memory attribute %{ATTR} has invalid MAIR encoding; this may lead to unpredictable results. Fields:

ATTR unsigned int

Memory attribute.

ArchMsg.Warning.warning_unpredictable_pmu_counter_access

TAGS **UNPREDICTABLE** DISPLAY Access to PMU counters from non-secure EL0 or EL1 is **UNPREDICTABLE** with MDCR_EL2.HPMN set to 0.

ArchMsg.Warning.warning_unpredictable_prioritization_breakpoint_match_vector_catch

TAGS **UNPREDICTABLE** DISPLAY Two events with the same priority occurred at the same instruction: Address Matching Vector Catch debug event, and Breakpoint debug event. It is constrained unpredictable which is taken.

ArchMsg.Warning.warning_unpredictable_stack_selection

TAGS **UNPREDICTABLE** DISPLAY **UNPREDICTABLE** state on context switch.

ArchMsg.Warning.warning_unpredictable_tsize_out_of_range

TAGS **UNPREDICTABLE** DISPLAY %{TCR}.T%{TTBR:d}SZ (%{TSIZE:d} bits) is out of range. Must be between 25 and %{TSIZE_MAX:d}. Fields:

TCR enum

Which TCR.

TSIZE signed int

Value of TSize.

TSIZE_MAX signed int

The maximum allowed value.

TTBR unsigned int

Which TTBR region.

ArchMsg.Warning.warning_unpredictable_unaligned_pc_as_base_register

TAGS **UNPREDICTABLE** DISPLAY The PC value must be word-aligned, otherwise the behavior of the instruction is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_unaligned_pop_stack

TAGS **UNPREDICTABLE** DISPLAY **CONSTRAINED UNPREDICTABLE** unaligned stack frame %{FRAME_ADDR:x} in exception-return. Fields:

FRAME_ADDR unsigned int

unaligned stack frame address.

ArchMsg.Warning.warning_unpredictable_unaligned_to_non_writeback

TAGS **UNPREDICTABLE** DISPLAY Unaligned Access to Non-Writeback-Cacheable Memory.

ArchMsg.Warning.warning_unpredictable_vmsa_memattrib

TAGS **UNPREDICTABLE** DISPLAY Unpredictable vmsa memory attribute with texcb value programmed to %{TEXCB} when tex remap is %{TEX_REMAP}. Fields:

TEXCB unsigned int

texcb value.

TEX_REMAP bool

use tex remap.

ArchMsg.Warning.warning_unpredictable_vtcr_t0sz_sl0

TAGS **UNPREDICTABLE** DISPLAY The combination of VTCR.TOSZ=%{TOSZ:d} and VTCR.SLO=%{SLO:d} is unpredictable. Fields:

SLO signed int

VTCR.SLO, starting level for stage 2 translation table walks.

TOSZ signed int

VTCR.TOSZ, required input address range.

ArchMsg.Warning.warning_unpredictable_write_DHCSR

TAGS **UNPREDICTABLE** DISPLAY **UNPREDICTABLE** set of bit-changes in DHCSR. Fields:

NEW unsigned int

data attempted to be written.

OLD unsigned int

previous value.

ArchMsg.Warning.warning_unpredictable_write_SHCSR

TAGS **UNPREDICTABLE** DISPLAY **UNPREDICTABLE** setting a pending bit to 1 for an exception with priority >= execution priority.

ArchMsg.Warning.warning_unsupported_access_to_memory_mapped_register

TAGS **UNPREDICTABLE** DISPLAY Access to memory mapped register is unsupported.

ArchMsg.Warning.warning_user_jmcr_access

TAGS **UNPREDICTABLE** DISPLAY User %{WRITE:(write|read)} access to JMCR is unpredictable. Fields:

WRITE unsigned int

access is write.

ArchMsg.Warning.warning_wcr_mask_and_bas

TAGS **UNPREDICTABLE** DISPLAY DBGWCR_EL1.MASK is non-zero and BAS is not 0xff.

ArchMsg.Warning.warning_wcr_mask_reserved

TAGS **UNPREDICTABLE** DISPLAY DBGWCR_ELn.MASK is set to a reserved value.

ArchMsg.Warning.warning_wcr_non_configuous_bas

TAGS **UNPREDICTABLE** DISPLAY DBGWCR_EL1.BAS has non-contiguous set of ones.

ArchMsg.Warning.warning_wvr_masked_bit_not_zero

TAGS **UNPREDICTABLE** DISPLAY DBGWVRn_EL1.VA contains a bit masked by DBGWCRn_EL1.MASK that is not zero.

ArchMsg.Why.why_branch_target_exception

A branch target exception has occurred. DISPLAY branch target exception: target pc=%{PC}, target opcode=%{OPCODE}. PRIMARY KEY PC. Fields:

BTYPE unsigned int

pstate.BTYPE at time branch taken.

OPCODE unsigned int

opcode at branch target.

PC unsigned int

Virtual Address of branch target.

ArchMsg.Why.why_illegal_state

An illegal mode change has occurred. DISPLAY illegal mode change: %{MESSAGE}. Fields:

MESSAGE string

The message.

ArchMsg.warning_invalid_interval_counter_reload_value

Invalid interval counter reload value in PMSIRR_EL1.INTERVAL: software should set this to a value greater than or equal to the minimum indicated by PMSIDR_EL1.Interval.

ArchMsg.warning_invalid_spe_buffer_write_pointer

Invalid value programmed in PMBPTR_EL1.PTR or PMBLIMITR_EL1.LIMIT: Current write pointer must be at least one sample record below the write limit pointer and PMBPTR_EL1.PTR[63:56] must be equal to PMBLIMITR_EL1.LIMIT[63:56].

BRANCH_MISPREDICT

Simulating branch mispredict. Fields:

PC unsigned int

Origin address (or 0 if unavailable).

BRA_DIR

Direct branches, to immediate address. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISSET enum

The instruction set of the branch instruction.

IS_COND bool

Indicates if this is a conditional branch.

IS_HINTED bool

Whether the conditional instruction is hinted.

IS_ISB bool

Whether the instruction is ISB.

IS_LINK bool

Indicates if this is a branch with link.

PC unsigned int

The address of the branch instruction.

TARGET_ISET enum

The instruction set after the branch.

TARGET_PC unsigned int

The address the instruction branches to.

BRA_INDIR

Indirect branches, perhaps to a register. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISET enum

The instruction set of the branch instruction.

IS_COND bool

Indicates if this is a conditional branch.

IS_HINTED bool

Whether the conditional instruction is hinted.

IS_ISB bool

Whether the instruction is ISB.

IS_LINK bool

Indicates if this is a branch with link.

PC unsigned int

The address of the branch instruction.

TARGET_ISET enum

The instruction set after the branch.

TARGET_PC unsigned int

The address the instruction branches to.

BRA_INDIR_CCFAIL

Indirect branches fail, perhaps to a register.

BRA_RET_CCFAIL

Branch return fail, perhaps to a register.

CACHE_MAINTENANCE_OP

Cache Maintenance Operation. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

DATA unsigned int

Specified MVA or Set/way.

FUNCTION enum

Clean / Invalidate.

SCOPE enum

Affected region.

SIDE enum

Inst / Data.

CCFAIL

Conditional instruction condition check fail. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CCFAIL_UNC

Unconditional instruction condition check fail. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CCPASS

Conditional instruction condition check pass. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CCPASS_UNC

Unconditional instruction condition check pass. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CODE_CACHE_MAINT

Code cache maintenance. Fields:

END_INCL unsigned int

Inclusive end address for VA or PA requests.

NS bool

Non-secure world for PA.

START unsigned int

Start address for VA or PA requests.

TYPE enum

Request type.

COMPILE_BLOCK_END

Last instruction of basic block translated. Fields:

VADDR unsigned int

Address of next instruction after this basic block.

COMPILE_BLOCK_START

First instruction of basic block translated. Fields:

VADDR unsigned int

Address of first instruction in basic block.

COMPILE_INST

ARM instruction compiled. Fields:

DISASS string

Disassembly of the instruction.

ISSET enum

The instruction set of this instruction.

ITSTATE unsigned int

The ITSTATE current for the instruction.

OPCODE unsigned int

The opcode of the instruction.

PC unsigned int

The address of the instruction.

SIZE unsigned int

The size of the instruction in bytes.

CONTEXTIDR

Context ID Register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

NS bool

Secure or nonsecure banked register is accessed.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The new value written.

CONTEXT_SYNC

Called for every context synchronization event. Fields:

ARCH_CSE bool

Architectural Context Synchronization Event.

PC unsigned int

The address of the synchronization instruction.

SYSREG_SYNC bool

System register synchronization event.

CORE_ENDIAN

Core BE8 Big-Endian state changed. Fields:

BE8 bool

Core BE8 Big-Endian state.

CORE_INFO

Static processor attributes. Only triggered by a call to DumpState(). Fields:

ARCH_PROFILE enum

The architecture profile of the core.

CLUSTER_ID unsigned int

The cluster ID of this processor.

CORE_NUM unsigned int

The number of this core in an MP processor.

DETAILED_CORE_FEATURES enum

Detailed core features as a bitfield. Enum values are bitmasks to extract the info from the field.

FPU_VERSION enum

The VFP version implemented by the core.

MEM_ARCH enum

The memory architecture of the core.

MIDR_EL1 unsigned int

MIDR_EL1 value.

NUM_CORES unsigned int

The number of cores in this MP processor.

QUANTUM_SIZE unsigned int

The default quantum size of the core (in ticks).

SECURITY_FEATURES bool

Does the core have security features?.

CORE_LOADS

Processor load accesses. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ACQREL enum

Is this an acquire/release.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DATA unsigned int

The data read or written.

ELEMENT_SIZE unsigned int

Width of each element.

ENDIANNESS enum

Byte ordering of this access.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

LOCK enum

Normal, exclusive or locked access.

MECID unsigned int

Memory Encryption Context ID, if applicable.

MECID2 unsigned int

Memory Encryption Context ID of the second page, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PADDR_TAG unsigned int

If tag checked, the physical address tag for the access.

PAGE_SIZE unsigned int

Page size as log2(size in bytes).

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK, 2=Exclusive Failed, 3=Unsupported.

SIZE unsigned int

Width of the access in bytes. Only required if DATA is not traced.

TAG_CHECKED bool

Is this access tag checked.

TAG_CHECKED2 bool

Is the second page access tag checked.

TRANS bool

Is this a translated access.

VADDR unsigned int

The virtual address of the access.

CORE_REGS

Changes of the core registers R0 to R14. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number, 0 to 14.

MODE enum

Bank of the register accessed.

OLD_VALUE unsigned int

The old value overwritten.

PHYS_ID enum

The physical register accessed.

VALUE unsigned int

The new value written to the register.

CORE_REGS64

Changes of the core registers X0..X30, SP_ELn. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ID enum

The register number, 0-30 for X0-X30, >=32 for SP_ELn.

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

CORE_REGS64_READ

General purpose AArch64 register has been read X0..X30, SP_ELn. Fields:

ID enum

The register number, 0-30 for X0-X30, >=32 for SP_ELn.

VALUE unsigned int

The value read from the register.

CORE_STALLED_TRANSITION

Detect when a core is prevented from progressing PChannel states. Fields:

CURRENT_STATE enum

Current power mode of core.

REASON string

The reason a transition was stalled, and what must be done to unblock it.

TARGET_STATE enum

Desired power mode of core after transition.

CORE_STORES

Processor store accesses. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ACQREL enum

Is this an acquire/release.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DATA unsigned int

The data read or written.

ELEMENT_SIZE unsigned int

Width of each element.

ENDIANNESS enum

Byte ordering of this access.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

LOCK enum

Normal, exclusive or locked access.

MECID unsigned int

Memory Encryption Context ID, if applicable.

MECID2 unsigned int

Memory Encryption Context ID of the second page, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PADDR_TAG unsigned int

If tag checked, the physical address tag for the access.

PAGE_SIZE unsigned int

Page size as $\log_2(\text{size in bytes})$.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK, 2=Exclusive Failed, 3=Unsupported.

SIZE unsigned int

Width of the access in bytes. Only required if DATA is not traced.

TAG_CHECKED bool

Is this access tag checked.

TAG_CHECKED2 bool

Is the second page access tag checked.

TRANS bool

Is this a translated access.

VADDR unsigned int

The virtual address of the access.

CP14_READ

System Coprocessor register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP14 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc1 unsigned int

opcode 1.

opc2 unsigned int

opcode 2.

CP14_WRITE

System Coprocessor register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP14 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc1 unsigned int

opcode 1.

opc2 unsigned int
opcode 2.

CP15_READ

System Coprocessor register read. Fields:

CORE_NUM unsigned int
Core number in a multi processor.

CRm unsigned int
CRm.

CRn unsigned int
CRn.

NS bool
Secure or nonsecure banked register is accessed.

REG_NAME string
Name of the CP15 register accessed.

SECURITY_STATE enum
The current security state.

UNDEF bool
The register accessed is undefined.

VALUE unsigned int
The value read.

opc1 unsigned int
opcode 1.

opc2 unsigned int
opcode 2.

CP15_READ64

System Coprocessor register read. Fields:

CORE_NUM unsigned int
Core number in a multi processor.

CRm unsigned int
CRm.

NS bool
Secure or nonsecure banked register is accessed.

REG_NAME string
Name of the CP15 register accessed.

SECURITY_STATE enum
The current security state.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc unsigned int

opcode 1.

CP15_WRITE

System Control Coprocessor register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP15 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc1 unsigned int

opcode 1.

opc2 unsigned int

opcode 2.

CP15_WRITE64

System Coprocessor register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP15 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc unsigned int

opcode 1.

CPSR

CPSR change. (In AArch64 PSTATE represented in SPSR format). Fields:

CORE_NUM unsigned int

Core number in a multi processor.

OLD_VALUE unsigned int

The old CPSR value.

UNKNOWN unsigned int

Bits within the register that have unknown value.

VALUE unsigned int

The new CPSR value.

CRYPTO_SPEC

Every crypto instruction speculatively executed.

CorePowerStateChange

Core power-up state has changed. DISPLAY Core powered %{PU:(up|down)}. Fields:

PU bool

Whether the core powered up.

DATA_CACHE_SET_ALLOC_TAGS

Set allocation tags from DC GVA instruction. Fields:

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate

when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

NSDESC unsigned int

The security state of the access.

PADDR unsigned int

The physical address of the access.

PAS enum

The physical address space of the page.

RESPONSE enum

0=Aborted, 1=OK.

SIZE unsigned int

The size of memory in bytes.

TAG unsigned int

The value of the tag being set.

TAGGING_TYPE enum

Type of Memory tagging (Physical or Virtual).

TAG_VADDR unsigned int

The virtual address of the Virtual Tag if Virtual tagging is used; otherwise 0.

VADDR unsigned int

Virtual address used for setting allocation tags.

DATA_CACHE_ZERO

Zero and invalidate cache line from DC ZVA instruction. Fields:

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

NSDESC unsigned int

The security state of the access.

PADDR unsigned int

The physical address of the access.

PADDR_TAG unsigned int

If tag checked, the physical address tag for the access.

PAS enum

The physical address space of the page.

RESPONSE enum

0=Aborted, 1=OK.

SIZE unsigned int

The size of zero'd memory in bytes.

TAG_CHECKED bool

Is this access tag checked.

VADDR unsigned int

Virtual address of the zero'd cache line.

DEBUG_EVENT

Hardware debug support event. Fields:

EVENT enum

Description of event.

VALUE unsigned int

data value.

DEBUG_TARGET_ISET

Hardware debug target ISA. Fields:

TARGET_ISET enum

The target instruction set of the debug state.

DMI_ALLOCATE

DMI allocated. Fields:

ACCESS_DIRECTION enum

READ / WRITE.

PADDR unsigned int

The physical address of the DMI.

PTR unsigned int

DMI range pointer.

SIDE enum

DSIDE / ISIDE.

SIZE unsigned int

The size of DMI.

VADDR unsigned int

The virtual address of the DMI..

DMI_HIT

DMI hit, access used DMI (only fire for small DMI regions i.e. less than a full page). Fields:

ACCESS_DIRECTION enum

READ / WRITE.

PADDR unsigned int

The physical address of the DMI.

SIDE enum

DSIDE / ISIDE.

SIZE unsigned int

The size of the DMI.

VADDR unsigned int

The virtual address of the DMI..

DMI_REVOKE

DMI revoked. Fields:

ACCESS_DIRECTION enum

READ / WRITE.

PADDR unsigned int

The physical address of the DMI.

SIDE enum

DSIDE / ISIDE.

SIZE unsigned int

The size of DMI.

VADDR unsigned int

The virtual address of the DMI..

END_COMPILE

Compilation end. Fields:

END_OF_PAGE_COUNT unsigned int

Number of basic blocks exited due to page boundary since START_COMPILE.

FETCHFAIL_COUNT unsigned int

Number of basic blocks exited due to ifetch failure START_COMPILE.

INST_COUNT unsigned int

Number of instructions compiled since START_COMPILE.

NONSEQ_COUNT unsigned int

Number of basic blocks exited due to non-sequential instructions since START_COMPILE.

PAGE_STRADDLE_COUNT unsigned int

Number of basic blocks exited due to unaligned instructions crossing page since START_COMPILE.

ESB

Error Synchronization Barrier hit. Fields:

ASYNC bool

ESB is async.

IMPLICIT bool

ESB is implicit.

EXCEPTION

Exception taken. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ESR unsigned int

The value assigned to ESR when exception is taken.

ESR_EC unsigned int

The Value assigned to ESR.EC when exception is taken.

IS_PHYSICAL bool

Physical or Virtual exception.

IS_TRAP bool

Exception coming from a trap.

LR unsigned int

The value assigned to the link register.

PC unsigned int

The location where the exception occurred.

PREFERRED_RETURN unsigned int

The preferred return address for the exception.

TAKEN_LOCALLY bool

Whether exception is taken locally(same exception level).

TARGET_ISET enum

The instruction set of the exception handler code.

TARGET_PC unsigned int

The address the exception branches to.

VECTOR enum

The exception vector.

VECTOR_OFFSET unsigned int

The offset of an exception vector from the base address.

EXCEPTION_END

Every exception completed.

EXCEPTION_INFO

Trace human-readable reason of the current exception for data/instruction aborts, floating point exceptions and unknown exceptions. Fields:

DESCRIPTION string

Reason for current exception.

ESR_EC unsigned int

The value of ESR.EC when exception is taken.

EXCEPTION_RAISE

Every exception raised.

EXCEPTION_RETURN

Branches on leaving exception. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISET enum

The instruction set of the branch instruction.

IS_COND bool

Indicates if this is a conditional branch.

IS_HINTED bool

Whether the conditional instruction is hinted.

IS_ISB bool

Whether the instruction is ISB.

IS_LINK bool

Indicates if this is a branch with link.

PC unsigned int

The address of the branch instruction.

TARGET_ISET enum

The instruction set after the branch.

TARGET_PC unsigned int

The address the instruction branches to.

EXCEPTION_RETURN_PREBRANCH

Exception return event that must occur before PC and processor state have updated.

EXCEPTION_START

Every exception started.

FIQ_TAKEN

FIQ taken exception.

FP_EXCEPTIONS_SIGNALLED

Trace which FP exceptions were signalled as a part of the FP operations performed by an instruction. Fields:

FP_EXCEPTIONS_SIGNALLED_MASK unsigned int

Indicates which FP exceptions were signalled by an instruction. This uses the same encoding as the cumulative floating-point exception bitfields in FPSR.

FP_STATE

Floating point state. Fields:

FPSR unsigned int

FPSR register value.

MASK unsigned int

The mask indicating updated cumulative bits.

FREQ_CHANGED

Core frequency changed. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The instruction count of this core.

NEW_FREQ signed int

The new frequency of this core (expressed in Hz).

OLD_FREQ signed int

The old frequency of this core (expressed in Hz).

HLT

HLT instruction occurred. Fields:

IMM16 unsigned int

The 16-bit immediate value encoded in HLT instruction.

INST

Every instruction executed. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DEBUG_STATE bool

The instruction is executed in debug state.

DISASS string

Disassembly of instruction.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISSET enum

The current instruction set.

ITSTATE unsigned int

The current ITSTATE.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

MODE enum

The mode the core is in.

NS unsigned int

The core's non-secure bit.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

OPCODE unsigned int

The opcode of the instruction.

PADDR unsigned int

The physical address of the instruction.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the instruction.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the page.

PC unsigned int

The address of the instruction.

SECURITY_STATE enum

The core's security state.

SIZE unsigned int

The size of the instruction in bytes.

INST_END

Every instruction completed.

INST_START

Every instruction started. Fields:

DEBUG_STATE bool

The instruction is executed in debug state.

ISSET enum

The current instruction set.

MODE enum

The mode the core is in.

NS enum

The current Secure State.

PC unsigned int

The address of the conditional instruction.

SECURITY_STATE enum

The current Security State.

INST_STRADDLE

Instruction straddles boundary. Fields:

DEBUG_STATE bool

The instruction is executed in debug state.

ISSET enum

The current instruction set.

MODE enum

The mode the core is in.

NS enum

The current Secure State.

PC unsigned int

The address of the conditional instruction.

SECURITY_STATE enum

The current Security State.

IRQ_TAKEN

IRQ taken exception.

ISB

ISB instruction occurred. Fields:

PC unsigned int

The address of the ISB instruction.

LOAD_MULTIPLE_REGS_LIST

List of destination registers (not source) of a multiple registers load instruction. Fields:

REG string

Register to add into the list.

LOCAL_MONITOR

Local monitor activity. Fields:

PADDR unsigned int

Local Monitor Address.

State enum

State of the monitor (Open/Exclusive).

MAPPING_CREATE

A memory mapping has been created. This is an implementation detail of the model and so may change without notice. Fields:

ASID unsigned int

The ASID used during translation, or 0 if no ASID was used.

IS_VALID bool

Whether the mapping is valid or not.

PADDR unsigned int

Base physical address of region, or 0 if not known.

SIDE enum

I-side or D-side.

SIZE unsigned int

Size of the page, or 0 if not known.

VADDR unsigned int

Base virtual address of region.

VMID unsigned int

The VMID used during translation, or 0 if no VMID was used.

MAPPING_DESTROY

A memory mapping has been destroyed. This is an implementation detail of the model and so may change without notice. Fields:

ASID unsigned int

The ASID used during translation.

CAUSE enum

The cause of invalidation.

PADDR unsigned int

Base physical address of page.

SIDE enum

I-side or D-side.

SIZE unsigned int

Size of the page.

VADDR unsigned int

Base virtual address of page.

VMID unsigned int

The VMID used during translation.

MEMMAP_DEBUG_READ

Memory mapped reads to the debug registers. Fields:

ADDR unsigned int

address.

CORE_NUM unsigned int

Core number in a multi processor.

EXT bool

Whether access is from an external device (such as the DAP).

NS enum

Secure state of the access.

REG_NAME string

Name of the debug register accessed.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

MEMMAP_DEBUG_WRITE

Memory mapped writes to the debug registers. Fields:

ADDR unsigned int

address.

CORE_NUM unsigned int

Core number in a multi processor.

EXT bool

Whether access is from an external device (such as the DAP).

NS enum

Secure state of the access.

REG_NAME string

Name of the debug register accessed.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The new value written.

MEMTAG_LOADS

Allocation tag memory read; when attributes and system configuration allow tag accesses.

Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

NUM_GRANULES unsigned int

Number of tags to be Loaded/stored.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK.

TAGGING_TYPE enum

Type of Memory tagging (Physical or Virtual).

TAGS unsigned int

The tags read or written. Tags for subsequent granules would be found at subsequent bytes in this field (One byte for each tag).

TAG_VADDR unsigned int

The virtual address of the Virtual Tag if Virtual tagging is used; otherwise 0.

VADDR unsigned int

The virtual address of the access.

MEMTAG_LOAD_INST

Allocation tag load instruction; regardless of tags being actually accessed. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

NUM_GRANULES unsigned int

Number of tags to be Loaded/stored.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK.

TAGGING_TYPE enum

Type of Memory tagging (Physical or Virtual).

TAGS unsigned int

The tags read or written. Tags for subsequent granules would be found at subsequent bytes in this field (One byte for each tag).

TAG_VADDR unsigned int

The virtual address of the Virtual Tag if Virtual tagging is used; otherwise 0.

VADDR unsigned int

The virtual address of the access.

MEMTAG_STORES

Allocation tag memory write; when attributes and system configuration allow tag accesses.

Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

NUM_GRANULES unsigned int

Number of tags to be Loaded/stored.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK.

TAGGING_TYPE enum

Type of Memory tagging (Physical or Virtual).

TAGS unsigned int

The tags read or written. Tags for subsequent granules would be found at subsequent bytes in this field (One byte for each tag).

TAG_VADDR unsigned int

The virtual address of the Virtual Tag if Virtual tagging is used; otherwise 0.

VADDR unsigned int

The virtual address of the access.

MEMTAG_STORE_INST

Allocation tag store instruction; regardless of tags being actually accessed. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

NUM_GRANULES unsigned int

Number of tags to be Loaded/stored.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK.

TAGGING_TYPE enum

Type of Memory tagging (Physical or Virtual).

TAGS unsigned int

The tags read or written. Tags for subsequent granules would be found at subsequent bytes in this field (One byte for each tag).

TAG_VADDR unsigned int

The virtual address of the Virtual Tag if Virtual tagging is used; otherwise 0.

VADDR unsigned int

The virtual address of the access.

MMU_D128_TTB_READ

This event is triggered by reads caused by a D128 translation table walk. Fields:

ABORTED bool

Was the walk successful.

DATA_HIGH unsigned int

The data upper bits read.

DATA_LOW unsigned int

The data lower bits read.

IPA unsigned int

For stage 1, the IPA of the read.

LEVEL unsigned int

Translation table level.

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

OUTADDR unsigned int

The output physical address.

PADDR unsigned int

The physical address of the read.

PASmem enum

Physical Address Space of the accessed memory.

PASreq enum

Physical Address Space of the requested lookup.

REGIME_EL enum

Entry matches in this translation regime.

REQUESTER enum

The requester of this table walk.

SIDE enum

Inst / Data.

STAGE unsigned int

Translation stage.

MMU_D128_TTB_WRITE

This event is triggered by writes caused by a D128 translation table walk. Fields:

ABORTED bool

Was the walk successful.

DATA_HIGH unsigned int

The data upper bits written.

DATA_LOW unsigned int

The data lower bits written.

IPA unsigned int

For stage 1, the IPA of the write.

IS_HACDBS_ACCESS bool

If the access was due to HACDBS_CLEAN.

LEVEL unsigned int

Translation table level.

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

OUTADDR unsigned int

The output physical address.

PADDR unsigned int

The physical address of the write.

PASmem enum

Physical Address Space of the accessed memory.

PASreq enum

Physical Address Space of the requested lookup.

REGIME_EL enum

Entry matches in this translation regime.

SIDE enum

Inst / Data.

STAGE unsigned int

Translation stage.

MMU_TRANS

Address translation information. Fields:

ASID unsigned int

Address space identifier.

CORE_NUM unsigned int

Core number in a multi processor.

Hyp bool

Entry matches in Hyp state only.

INNERCACHE_RA bool

Is the inner cache allocate on read.

INNERCACHE_TRANSIENT bool

Is the inner write-through transient.

INNERCACHE_TYPE enum

Inner Caching scheme (NC/MB/WA).

INNERCACHE_WA bool

Is the inner cache allocate on write.

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MEMTYPE enum

Memory type.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC enum

Is secure side supposed to access secure or nonsecure memory.

OUTERCACHE_RA bool

Is the outer cache allocate on read.

OUTERCACHE_TRANSIENT bool

Is the outer write-through transient.

OUTERCACHE_TYPE enum

Outer Caching scheme (NC/MB/WA).

OUTERCACHE_WA bool

Is the outer cache allocate on write.

PADDR unsigned int

Physical address of the access.

PAGESIZE unsigned int

Page size as $\log_2(\text{size})$.

PAS enum

PAS supposed to be accessed is secure or nonsecure memory or Root/Realm if RME is implemented.

SH enum

Shareability.

SIDE enum

Inst / Data.

STAGE1_PERM unsigned int

Stage 1 permissions, [5:3] Privileged access (XWR) [2:0] User access (XWR).

STAGE2_PERM unsigned int

Stage 2 permission mask [5:3] Privileged access (XWR) [2:0] User access (XWR).

VADDR unsigned int

Virtual address of the access.

VMID unsigned int

Virtual machine identifier.

XS bool

XS Attribute.

nG enum

Flag indicating whether ASID will be matched.

MMU_TTB_READ

This event is triggered by reads caused by a translation table walk. Fields:

ABORTED bool

Was the walk successful.

DATA unsigned int

The data read.

DESC_KIND enum

The encoding scheme of the descriptors used for this walk.

IPA unsigned int

For stage 1, the IPA of the read.

LEVEL unsigned int

Translation table level.

LPAE bool

Is this for an LPAE translation (*DEPRECATED*, use DESC_KIND).

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSmem bool

Non secure memory access.

NSreq bool

Non secure request.

OUTADDR unsigned int

The output physical address.

PADDR unsigned int

The physical address of the read.

PASmem enum

Physical Address Space of the accessed memory.

PASreq enum

Physical Address Space of the requested lookup.

REGIME_EL enum

Entry matches in this translation regime.

REQUESTER enum

The requester of this table walk.

SIDE enum

Inst / Data.

STAGE unsigned int

Translation stage.

MMU_TTB_WRITE

This event is triggered by writes caused by a translation table walk. Fields:

ABORTED bool

Was the walk successful.

DATA unsigned int

The data written.

DESC_KIND enum

The encoding scheme of the descriptors used for this walk.

IPA unsigned int

For stage 1, the IPA of the write.

IS_HACDBS_ACCESS bool

If the access was due to HACDBS_CLEAN.

LEVEL unsigned int

Translation table level.

LPAE bool

Is this for an LPAE translation (*DEPRECATED*, use DESC_KIND).

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSmem bool

Non secure memory access.

NSreq bool

Non secure request.

OUTADDR unsigned int

The output physical address.

PADDR unsigned int

The physical address of the write.

PASmem enum

Physical Address Space of the accessed memory.

PASreq enum

Physical Address Space of the requested lookup.

REGIME_EL enum

Entry matches in this translation regime.

SIDE enum

Inst / Data.

STAGE unsigned int

Translation stage.

MODE_CHANGE

Mode change. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

MODE enum

The new mode.

NON_SECURE enum

The core's new non-secure bit.

OLD_MODE enum

The old mode.

SECURITY_STATE enum

The new security state.

MOPS_LD_RETIRED

Load instructions architecturally executed for MOPS.

MOPS_ST_RETIRED

Store instructions architecturally executed for MOPS. Fields:

IS_MEMCPY bool

Whether the memory operation is MEMCPY.

NOT_TAG_CHECKED_INST

MTE Tag not checked for current instruction.

PERIODIC

Called for every quantum. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The instruction count of this CPU.

PC unsigned int

The address of the next instruction to be executed on this CPU.

PMU_COUNTER_OVERFLOW

PMU counter overflow. Fields:

COUNTER_GROUP enum

Counter group.

COUNTER_RANGE enum

Counter range.

EVENT_ID unsigned int

Event Identifier, field is valid only for Event counter group.

INDEX unsigned int

Counter index (as selected by PMSELR), field is valid only for Event/Cycle counter groups.

INTERRUPT bool

Is interrupt enabled on overflow for this counter.

POINTER_AUTH

Called when execution hits a pointer authentication check. Fields:

AUTH_STATE bool

Authentication result.

KEY_USED enum

Pointer authentication key.

POTINDEX_UPDATE

Value of POT index for current pc changed. Fields:

NEW_POTINDEX unsigned int

New POTIndex.

OLD_POTINDEX unsigned int

Old POTIndex.

PREFETCH_MEMORY64

Prefetch from PRFM or PRFUM instructions. Fields:

PRFOP enum

Prefetch hint.

VADDR unsigned int

Virtual address of the location that should be prefetched.

PRELOAD64_REQUEST

Preload requests from PRFM or PRFUM instructions. Fields:

PRFOP enum

Prefetch hint.

VADDR unsigned int

Virtual address of the location that should be prefetched.

PRELOAD_DATA

Data preload from PLD instruction. Fields:

VADDR unsigned int

Virtual address of the data that should be preloaded.

PRELOAD_INST

Instruction preload from PLI instruction. Fields:

VADDR unsigned int

Virtual address of the instruction that should be preloaded.

PRE_CORE_LOAD

Trace just before a core load. Fields:

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

PRE_CORE_STORE

Trace just before a core store. Fields:

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

PRE_TTB_READ

This event is triggered before reads caused by a translation table walk. Fields:

ADDR unsigned int

The physical address of the read.

PSTATE

PSTATE fields update. Fields:

ALLINT unsigned int

All interrupt mask bit.

BTYPE unsigned int

Branch target identification bit.

CORE_NUM unsigned int

Core number in a multi processor.

DAIF unsigned int

Exception mask bits.

DIT unsigned int

Data Independent Timing bit.

EL unsigned int

Current Exception level field.

EXLOCK unsigned int

Exception return state lock bit.

IL unsigned int

Illegal Execution state bit.

NZCV unsigned int

Condition flags.

PACM unsigned int

PACM bit.

PAN unsigned int

Privileged Access Never state bit.

PM unsigned int

PMU exception mask bit.

PPEND unsigned int

PMU exception pending bit.

SM unsigned int

StreamingMode status bit.

SP unsigned int

Stack pointer register selection bit.

SS unsigned int

Software Step bit.

SSBS unsigned int

Speculative Store Bypass Safe bit.

TCO unsigned int

Tag Check Override bit.

TOUCHED unsigned int

Mask of the fields that have been touched since this trace has last been emitted. Note that touched does not necessarily mean the value changed. The index of the field is the bit index in the mask plus the index of this field plus one.

UAO unsigned int

User Access Override bit.

UINJ unsigned int

Undefined Instruction exceptions bit.

ZA unsigned int

ZA storage and ZT0 register enable bit.

nRW unsigned int

Current Execution state bit.

PSTATE_PACTIVE_OPMODE_TRANSITION

Trace pactive and Opmode mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Operating mode after the transition.

OLD_STATE enum

Operating mode before the transition.

PSTATE_PACTIVE_OPMODE_TRANSITION_DENIED

Trace the denied pactive and Opmode mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Operating mode after the transition.

OLD_STATE enum

Operating mode before the transition.

REASON string

Reason for the transition being denied.

PSTATE_PACTIVE_POWER_TRANSITION

Trace pactive and power mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Power mode after the transition.

OLD_STATE enum

Power mode before the transition.

PSTATE_PACTIVE_POWER_TRANSITION_DENIED

Trace the denied pactive and power mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Power mode after the transition.

OLD_STATE enum

Power mode before the transition.

REASON string

Reason for the transition being denied.

PSTATE_PACTIVE_THREAD_OPMODE_TRANSITION

Trace pactive and Opmode mode transitions for threads on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Operating mode after the transition.

OLD_STATE enum

Operating mode before the transition.

QUANTUM_START

Called at the start of every quantum.

RANGE_PRELOAD_REQUEST

Preload requests from RPRFM. Fields:

COUNT unsigned int

Number of blocks of data to access.

LENGTH unsigned int

Number of contiguous bytes to be accessed in each block.

OP unsigned int

Range prefetch operation .

REUSE unsigned int

Maximum number of bytes to be accessed before next RPRFM .

STRIDE signed int

Stride to advance after accessing LENGTH bytes.

VADDR unsigned int

Base virtual address of range prefetch request.

RUN_STATE

Run state transition. Fields:

INST_COUNT unsigned int

Ticks count at point of transition.

NEW enum

New run state.

OLD enum

Old run state.

SBP_SAMPLE_COLLISION

DISPLAY Statistical profiling samples collided .

SBP_SAMPLE_DISCARDED

DISPLAY Operation discarded by statistical profiling unit when configured in discard mode.
Fields:

BUFFER_POINTER unsigned int

Value of buffer pointer.

DATA unsigned int

Data discarded.

SIZE unsigned int

Size of sample, in bytes.

SBP_SAMPLE_END

DISPLAY Statistical profiling of an operation ended.

SBP_SAMPLE_POP

DISPLAY Operation which might be sampled by statistical profiling.

SBP_SAMPLE_RECORDED

DISPLAY Operation sampled by statistical profiling has been written to the buffer at %{BUFFER_ADDRESS} with size %{SIZE} and data %{DATA}. Fields:

BUFFER_ADDRESS unsigned int

Virtual Address of start of buffer write.

DATA unsigned int

Data written to the buffer.

SIZE unsigned int

Size of sample written to buffer, in bytes.

SBP_SAMPLE_START

DISPLAY Statistical profiling of an operation started.

SBP_SAMPLE_TAKEN

DISPLAY Operation sampled by statistical profiling after filtering.

SEMIHOSTING_CALL

Call of a semihost function occurred. Fields:

PC unsigned int

The program counter after the semihosting call.

SEMIHOSTING_PRECALL

About to call semihost. Fields:

EL2 bool

Use EL2 translation regime.

NS bool

Is Non-Secure.

REG_WIDTH unsigned int

The current register width in bytes.

SIGNAL

External signal state change. Fields:

SIGNAL enum

Signal that changed.

STATE bool

Signal asserted state.

SOFTWARE_STEP

Return the Debug Software Step state. Fields:

STATE enum

Software Step state.

SPSR

SPSR change. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

MODE enum

Which of the banked SPSR registers is written.

OLD_VALUE unsigned int

The old SPSR value.

VALUE unsigned int

The new SPSR value.

START_COMPILE

Compilation started. Fields:

VADDR unsigned int

Instruction where compilation begins.

STG_MULTIPLE_MICRO_OPS_ADDRESS_RANGE

Address range for MTE stores with multiple uops. Fields:

ADDR_END unsigned int

Final address.

ADDR_START unsigned int

Starting address.

STORE_MULTIPLE_REGS_ADDRESS_RANGE

Address range for stores with multiple registers. Fields:

ADDR_END unsigned int

Final address.

ADDR_START unsigned int

Starting address.

SUBOPTIMAL_LDST_RETIRED

suboptimal load/store. Fields:

ACCESS_SIZE unsigned int

Log2 of the access width in bytes used for alignment checking (1: half-word, 2: word ...).

OPTIMAL_ALIGN_SELECTOR unsigned int

Selector to decide the optimal alignment size (1: access width, 2: 4-bytes, 3: 8-bytes, ..., 12: 4-kbytes).

VADDR unsigned int

The virtual address of the access.

SVE_INST_SPEC

SVE operations speculatively executed.

SVE_LD_RETIRED

SVE load instructions architecturally executed.

SVE_LOADS

SVE load accesses. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ACQREL enum

Is this an acquire/release.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DATA unsigned int

The data read or written.

ELEMENT_SIZE unsigned int

Width of each element.

ENDIANNESS enum

Byte ordering of this access.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

LOCK enum

Normal, exclusive or locked access.

MECID unsigned int

Memory Encryption Context ID, if applicable.

MECID2 unsigned int

Memory Encryption Context ID of the second page, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PADDR_TAG unsigned int

If tag checked, the physical address tag for the access.

PAGE_SIZE unsigned int

Page size as $\log_2(\text{size in bytes})$.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK, 2=Exclusive Failed, 3=Unsupported.

SIZE unsigned int

Width of the access in bytes. Only required if DATA is not traced.

TAG_CHECKED bool

Is this access tag checked.

TAG_CHECKED2 bool

Is the second page access tag checked.

TRANS bool

Is this a translated access.

VADDR unsigned int

The virtual address of the access.

SVE_MEM_PREFETCH

Scalable Vector Extension memory prefetch. Fields:

CACHE_LEVEL enum

The preferred cache level.

HINT enum

What the memory is expected to be used for.

SIZE unsigned int

The prefetch size in bytes.

TEMPORAL bool

Whether the data is temporal.

VADDR unsigned int

The virtual address of the prefetch.

SVE_REG_READ

Scalable Vector Extension register read. Fields:

ID enum

The register identifier.

SM bool

Whether the PE is in Streaming Mode.

VALUE unsigned int

The read data.

SVE_REG_UPDATE

Scalable Vector Extension register update. Fields:

ID enum

The register identifier.

SM bool

Whether the PE is in Streaming Mode.

VALUE unsigned int

The written data.

SVE_REG_WRITE

Scalable Vector Extension register write. Fields:

ID enum

The register identifier.

SM bool

Whether the PE is in Streaming Mode.

VALUE unsigned int

The written data.

SVE_STORES

SVE store accesses. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ACQREL enum

Is this an acquire/release.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DATA unsigned int

The data read or written.

ELEMENT_SIZE unsigned int

Width of each element.

ENDIANNESS enum

Byte ordering of this access.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

LOCK enum

Normal, exclusive or locked access.

MECID unsigned int

Memory Encryption Context ID, if applicable.

MECID2 unsigned int

Memory Encryption Context ID of the second page, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PADDR_TAG unsigned int

If tag checked, the physical address tag for the access.

PAGE_SIZE unsigned int

Page size as $\log_2(\text{size in bytes})$.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK, 2=Exclusive Failed, 3=Unsupported.

SIZE unsigned int

Width of the access in bytes. Only required if DATA is not traced.

TAG_CHECKED bool

Is this access tag checked.

TAG_CHECKED2 bool

Is the second page access tag checked.

TRANS bool

Is this a translated access.

VADDR unsigned int

The virtual address of the access.

SVE_ST_RETIRED

SVE store instructions architecturally executed.

SVE_UNALIGNED_LDST_RETIRED

SVE load or store instructions architecturally executed that would cause alignment fault if A bit is set.

SVE_Z_PRED_MEM_ACCESS_INFO

Information about SVE/SME predicated accesses to Z registers (non Gather/Scatter). Fields:

MEM_ELEM_SIZE unsigned int

Width of elements accessed in memory (in bytes).

RnW bool

Whether this is a Read not a Write.

VA unsigned int

Virtual base address computed for accesses.

SYNC

Called for every synchronization, that is every quantum break. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The instruction count of this CPU.

LOCAL_QUANTUM unsigned int

The local quantum of this CPU.

LOCAL_TIME unsigned int

The local time of this CPU.

SYNC_REASON enum

Reason for sync.

SYSCALL

System call instruction executed. Fields:

IMM unsigned int

Immediate value of the system call instruction.

TYPE enum

System call type.

VADDR unsigned int

Instruction that caused the system call.

SYSREG_READ128

System Coprocessor 128 bit register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

REGNUM enum

Internal register number.

REGNUM_PHYS enum

Internal register number physically accessed (e.g. in presense of VHE redirection).

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_READ64

System Coprocessor register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REGNUM enum

Internal register number.

REGNUM_OPERAND enum

Register number corresponding to the instruction operand.

REGNUM_PHYS enum

Internal register number physically accessed (e.g. in presense of VHE redirection).

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int
opcode 2.

SYSREG_UPDATE128

Triggers when the system updates a register. Fields:

CRm unsigned int
CRm.

CRn unsigned int
CRn.

NS bool
Secure or nonsecure banked register is accessed.

REG enum
Register number.

UNKNOWN unsigned int
Bits of the register which became unknown.

VALUE unsigned int
Value written to the register.

opc unsigned int
opcode 1.

opc0 unsigned int
opcode 0.

opc2 unsigned int
opcode 2.

SYSREG_UPDATE32

Triggers when the system updates a register. Fields:

CRm unsigned int
CRm.

CRn unsigned int
CRn.

NS bool
Secure or nonsecure banked register is accessed.

REG enum
Register number.

UNKNOWN unsigned int
Bits of the register which became unknown.

VALUE unsigned int
Value written to the register.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_UPDATE64

Triggers when the system updates a register. Fields:

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG enum

Register number.

UNKNOWN unsigned int

Bits of the register which became unknown.

VALUE unsigned int

Value written to the register.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_WRITE128

System Coprocessor 128 bit register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

REGNUM enum

Internal register number.

REGNUM_PHYS enum

Internal register number physically accessed (e.g. in presense of VHE redirection).

UNDEF bool

The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_WRITE64

System Coprocessor register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REGNUM enum

Internal register number.

REGNUM_OPERAND enum

Register number corresponding to the instruction operand.

REGNUM_PHYS enum

Internal register number physically accessed (e.g. in presense of VHE redirection).

UNDEF bool

The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

TRBU_BUFFER_WRAP

DISPLAY Trace buffer current write pointer wrapped.

TRBU_TRIGGER_EVENT

DISPLAY Trace buffer Trigger Event.

TRBU_WRITE

DISPLAY PETU trace data has been written to the TRBU memory buffer at %{BUFFER_ADDRESS} with size %{SIZE} and data %{DATA}. Fields:

BUFFER_ADDRESS unsigned int

Virtual Address of start of buffer write.

DATA unsigned int

Data written to the buffer.

SIZE unsigned int

Size of sample written to buffer, in bytes.

UNALIGNED_LDST_RETIRED

Processor unaligned load/store. Fields:

ACCESS_SIZE unsigned int

Log2 of the access width in bytes used for alignment checking (1=>half-word, 2->word ...).

NUMBER_OF_BEATS unsigned int

Number of accesses (beats) in this burst (Total size in bytes is ACCESS_SIZE * NUMBER_OF_BEATS).

VADDR unsigned int

The virtual address of the access.

VFP_D_REGS

VFP/NEON D 64 bit register write. Fields:

ALIASING enum

Type of register aliasing used.

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a write to the corresponding bit in the register. For example, 0x1 would indicate a write to VALUE[0].

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

VFP_D_REGS_READ

VFP/NEON D 64 bit register read. Fields:

ID unsigned int

The register number.

VALUE unsigned int

The current value of the register.

VFP_Q_REGS

VFP/NEON Q 128 bit register write. Fields:

ALIASING enum

Type of register aliasing used.

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a write to the corresponding byte in the register. For example, 0x1 would indicate a write to VALUE[7:0].

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

VFP_Q_REGS_READ

VFP/NEON Q 128 bit register read. Fields:

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a read to the corresponding byte in the register. For example, 0x1 would indicate a read to VALUE[7:0].

VALUE unsigned int

The current value of the register.

VFP_SYS_REGS

Writes to the VFP/NEON units system registers. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ID enum

Which VFP system register is written.

MASK unsigned int

The mask indicating updated cumulative bits.

OLD_VALUE unsigned int

The register's old value overwritten.

VALUE unsigned int

The new value written to the VFP system register.

VFP_S_REGS

VFP/NEON S 32 bit register write. Fields:

ALIASING enum

Type of register aliasing used.

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a write to the corresponding bit in the register. For example, 0x1 would indicate a write to VALUE[0].

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

VFP_S_REGS_READ

VFP/NEON S 32 bit register read. Fields:

ID unsigned int

The register number.

VALUE unsigned int

The current value of the register.

VIRTUAL_TAG_ALLOCATED

Virtual tag information when allocated. Fields:

TAG_VADDR unsigned int

Virtual tag virtual address.

VADDR unsigned int

Data Virtual address.

VIRTUAL_TAG unsigned int

Virtual tag value.

WAYPOINT

Signals end of basic block execution. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ISET enum

Origin instruction set.

IS_COND bool

Indicates if this is a conditional waypoint.

PC unsigned int

Origin address (or 0 if unavailable).

TAKEN bool

Indicates if this waypoint was taken.

TARGET unsigned int

Destination address.

TARGET_ISET enum

Destination instruction set.

WFE_END

WFE ended. Fields:

INST_COUNT unsigned int

Ticks count when leaving WFE.

WFE_EVENT_REGISTER

WFE event register status: set/clear, reason. Fields:

INST_COUNT unsigned int

Ticks count.

REASON enum

Reason for set/clear. Only REASON==0 (Cleared by WFE) clears the bit, all other reasons set it.

WFE_IGNORED

WFE ignored. Fields:

EVENT bool

This WFE was ignored because the event register was set.

INST_COUNT unsigned int

Ticks count when ignoring WFE.

TRAPPED bool

This WFE was trapped.

WFE_START

WFE entered. Fields:

INST_COUNT unsigned int

Ticks count when entering WFE.

WFI_END

WFI ended. Fields:

INST_COUNT unsigned int

Ticks count when leaving WFI.

WFI_IGNORED

WFI ignored. Fields:

DISABLED bool

This WFI was ignored because WFI is disabled.

INST_COUNT unsigned int

Ticks count when ignoring WFI.

REASON enum

specifies reason why WFI trace was ignored.

TRAPPED bool

This WFI was trapped.

WFI_START

WFI entered. Fields:

INST_COUNT unsigned int

Ticks count when entering WFI.

WFI_WAKEUP

WFI wakeup. Fields:

INST_COUNT unsigned int

Ticks count when WFI wakeup occurred.

REASON enum

Reason for wakeup.

2.10 ARM_C1-Pro

This section describes the trace sources.

AA64_ASE_SVE_REGS

Modification of AArch64 Advanced SIMD or SVE Vector Registers (via SIMD or Scalar operations), SVE Predicate Registers and ZT0. Fields:

ID `enum`

The register identifier.

MASK `unsigned int`

Bitmask of the register to signal the modified bits in the VALUE field.

SM `bool`

Whether the PE is in Streaming Mode.

VALUE `unsigned int`

Value of AArch64 vector register.

ASYNC_MEMORY_FAULT

Context ID Register write. Fields:

FAULT `unsigned int`

Fault status present in ESR.ISS encoding for Serror in ARCH64 and present in DFSR encoding for external abort format in ARCH32.

PADDR `unsigned int`

Physical Address (or 0 if unavailable).

VADDR `signed int`

Virtual Address (or 0 if unavailable).

ATOMIC_END_ACCESS

Bus trace access for atomics. Fields:

ACCESS_FAIL `bool`

Memory access failed.

ADDR `unsigned int`

The virtual address of the access.

ATTR `unsigned int`

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

COMPARE_VALUE `unsigned int`

Compare value for CAS.

LOAD_VALUE unsigned int

Loaded values.

NSDESC unsigned int

The security state of the access.

OPERAND_VALUE unsigned int

Operation's operand.

OPERATION enum

Operation type.

PADDR unsigned int

The physical address of the access.

PRIV bool

Is this a privileged access?.

ATOMIC_START_ACCESS

Bus trace access for atomics. Fields:

ADDR unsigned int

The virtual address of the access.

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

COMPARE_VALUE unsigned int

Compare value for CAS.

NSDESC unsigned int

The security state of the access.

OPERAND_VALUE unsigned int

Operation's operand.

OPERATION enum

Operation type.

PADDR unsigned int

The physical address of the access.

PRIV bool

Is this a privileged access?.

ArchMsg.Error.exit_code

DISPLAY exit code: %{EXITCODE} component kind: %{KIND} reason: %{REASON}. Fields:

EXITCODE signed int

Exit code to be returned.

KIND string

Component kind that invoked the exit code trace.

REASON string

Reason why the exit code trace was invoked (optional).

ArchMsg.Warning.IgnoredWarmResetRequest

Event fires when software attempts to request warm reset through write to deprecated DBGPRCR.CWRR.

ArchMsg.Warning.RAS_ErrselrUnpredictableState

DISPLAY ERRSELR is in unpredictable state due to value greater than number_of_error_records written to it.

ArchMsg.Warning.RAS_UnpredictableStateWrite

DISPLAY %{REG_NAME} write ignored due to unpredictable ERRSELR state. Fields:

REG_NAME string

Name of register being written.

ArchMsg.Warning.ReportedRevisionRegisterRead

DISPLAY Reported revision and patch do not match that of the cpu. Fields:

ORIGINAL_VALUE unsigned int

The value the register would have based on the modelled revision.

REGISTER string

The name of the register that has ben read.

REPORTED_VALUE unsigned int

The value the register has been overridden to.

ArchMsg.Warning.Unpredictably Indexed PM Event Register

TAGS UNPREDICTABLE DISPLAY Event %{IsDirect:(Selector value|Register)} %{SEL} is >= %{N}. Fields:

IsDirect unsigned int

Direct Access rather than indirect.

N unsigned int

Number of Accessible Registers.

SEL unsigned int

Selector.

ArchMsg.Warning.cache_contents_unknown

A cache has been activated without being invalidated and may contain **UNKNOWN** content. DISPLAY %{SIDE:(D|I)}-Cache was enabled in %{REGIME:(SECURE|NON_SECURE|HYP)} regime but was not invalidated since power-on: cache lines could contain **UNKNOWN** content %{SIDE:(Note: D-side caches must invalidate SECURE lines even if the cache is only used by NON_SECURE code in order to ensure that unexpected dirty lines tagged SECURE are not naturally evicted)}. Fields:

REGIME unsigned int

Current translation regime.

SIDE unsigned int

0 d-side cache or unified cache, 1 i-side.

ArchMsg.Warning.dcimvac_matches_watchpoint

TAGS **UNPREDICTABLE** DISPLAY Watchpoints matching an AArch32 DCIMVA are implementation defined.

ArchMsg.Warning.decode_fieldmismatch

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with different fields %{FIELD1} and %{FIELD2}. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_fracbitsnegative

TAGS **UNPREDICTABLE** DISPLAY VCVT Instruction is unpredictable with a negative number of fraction bits.

ArchMsg.Warning.decode_initblock

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable in an IT block.

ArchMsg.Warning.decode_initblocknotlast

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable in an IT block when not the last.

ArchMsg.Warning.decode_invalid_condition

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable when the condition is not AL.

ArchMsg.Warning.decode_invalidfieldcombination

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with the values in %{FIELD1} and %{FIELD2}. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_invalidvalue

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with the value in %{FIELD}. Fields:

FIELD enum

Field name.

ArchMsg.Warning.decode_mem_hint_unallocated

TAGS **UNPREDICTABLE** DISPLAY There is no memory hint allocated to this bit pattern.

ArchMsg.Warning.decode_registermatch

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with %{FIELD1} and %{FIELD2} as the same register. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_registermismatch

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with %{FIELD1} and %{FIELD2} as different registers. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_registeroutofrange

TAGS **UNPREDICTABLE** DISPLAY Floating-point instruction is unpredictable indexing beyond the end of the register bank.

ArchMsg.Warning.decode_sbzsbo

TAGS **UNPREDICTABLE** DISPLAY Reserved bits in the instruction are not canonical.

ArchMsg.Warning.decode_transactiontoobig

TAGS **UNPREDICTABLE** DISPLAY Attempt to perform a memory transaction of more than 128 bytes.

ArchMsg.Warning.decode_unpred_other

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable.

ArchMsg.Warning.decode_unpreduseofpc

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable when %{FIELD} is PC. Fields:

FIELD enum

Field name.

ArchMsg.Warning.decode_unpreduseofr13

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable when %{FIELD} is R13. Fields:

FIELD enum

Field name.

ArchMsg.Warning.decode_unpreduseofr13inreglist

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with R13 in register-list.

ArchMsg.Warning.decode_useofd16orhigher

TAGS **UNPREDICTABLE** DISPLAY Attempt to use register D16 or higher where it is not allowed.

ArchMsg.Warning.decode_writebackandbaseinlist

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with write back when the base register is in the transfer list.

ArchMsg.Warning.decode_zeroregistersinlist

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable when no registers are to be transferred.

ArchMsg.Warning.misaligned_ptrs

DISPLAY TRBPTR_EL1(%{TRBPTR_EL1}) or TRBTRG_EL1(%{TRBTRG_EL1}) are not aligned as per TRBIDR_EL1.Align(%{TRBIDR_EL1}). Fields:

TRBIDR_EL1 unsigned int

TRBIDR_EL1.Align value.

TRBPTR_EL1 unsigned int

Current write pointer.

TRBTRG_EL1 unsigned int

Trigger pointer.

ArchMsg.Warning.out_of_range_ptrs

DISPLAY TRBPTR_EL1(%{TRBPTR_EL1}) is out of range as per TRBBASER_EL1(%{TRBBASER_EL1}) or TRBLIMITR_EL1.LIMIT(%{TRBLIMITR_EL1}). Fields:

TRBBASER_EL1 unsigned int

Base pointer.

TRBLIMITR_EL1 unsigned int

Limit pointer.

TRBPTR_EL1 unsigned int

Current write pointer.

ArchMsg.Warning.recursive_branch

An instruction is performing a branch that targets the same instruction. If this was intended then a WFI might be more effective DISPLAY recursive branch detected at PC=%{PC}.

PRIMARY KEY PC. Fields:

PC unsigned int

Address of instruction causing the branch.

ArchMsg.Warning.recursive_exception

An instruction has generated an exception that targets the same instruction. DISPLAY recursive %{TYPE} exception detected at PC=%{PC} (CPSR=%{CPSR}). PRIMARY KEY PC. Fields:

CPSR unsigned int

Processor state.

PC unsigned int

Address of instruction causing the exception.

TYPE enum

Type of exception.

ArchMsg.Warning.reentrant_vector_catch

TAGS **UNPREDICTABLE** DISPLAY Vector catch debug event on Prefetch or Data abort vector %{ADDR}. Fields:

ADDR unsigned int

vector address.

ArchMsg.Warning.reserved_it_state

TAGS **UNPREDICTABLE** DISPLAY Execution with reserved IT state %{ITSTATE}. Fields:

ITSTATE unsigned int

reserved IT state.

ArchMsg.Warning.tlb_contents_unknown

Translation has been enabled without invalidating the TLB. DISPLAY Translation was enabled in the %{PAS:(Secure|Non-secure|Root|Realm)} EL%{EL:d} translation regime (vmid=%{VMID:x}) without invalidating %{INVALIDITY:(|d-side |i-side |)}entries since power-on: %{INVALIDITY:(|D|I|)}TLB could contain **UNKNOWN** content. Fields:

EL unsigned int

Exception level of translation regime.

INVALIDITY unsigned int

Which TLB may be invalid (if TLBs are separate).

NON_SECURE bool

Whether the translation regime is non-secure.

PAS unsigned int

Physical address space of translation regime.

VMID unsigned int

Current VMID, if applicable.

ArchMsg.Warning.unknown_DLR_DSPSR

Exiting debug state with unknown DLR or DSPSR.

ArchMsg.Warning.unknown_ELR_SPSR

Returning from debug exception with unknown ELR or SPSR.

ArchMsg.Warning.unpred_FAR_on_LRCPC3_non_atomic_mem_access

TAGS **UNPREDICTABLE** Unpredictable FAR/FSTATUS on LRCPC3 pair instructions resulting in two separate memory accessesDISPLAY Unpredictable FAR on LRCPC3 non atomic memory access at %{ADDR1} with FSTATUS %{FSTATUS1} or %{ADDR2} with FSTATUS %{FSTATUS2}. Fields:

ADDR1 unsigned int

Possible FAR address 1.

ADDR2 unsigned int

Possible FAR address 2.

FSTATUS1 unsigned int

Possible fault status value 1.

FSTATUS2 unsigned int

Possible fault status value 2.

ArchMsg.Warning.unpred_async_tag_check_fail_on_sve_speculative_load_do_update_ffr

Unpredictable set to FALSE of the FFR element associated to SVE speculative load with an async tag check fail TAGS **UNPREDICTABLE**.

ArchMsg.Warning.unpredictable_a32_breakpoint

TAGS **UNPREDICTABLE** DISPLAY Unpredictable breakpoint on a A32 instruction at %{ADDR1} Offset %{OFFSET} BAS %{BAS} for a %{IS_ADDRESS_MISMATCH_BREAKPOINT:(Mismatch|Match)} Breakpoint number %{N:d} Breakpoint address %{ADDR2}. Fields:

ADDR1 unsigned int

Address.

ADDR2 unsigned int

Breakpoint address.

BAS unsigned int

BAS field.

IS_ADDRESS_MISMATCH_BREAKPOINT bool

Is a Mismatch breakpoint, not a Match breakpoint.

N unsigned int

Breakpoint number.

OFFSET unsigned int

Always has a value of zero.

ArchMsg.Warning.unpredictable_banked_register_access

TAGS **UNPREDICTABLE** DISPLAY Access to banked register (R=%{R:d} SYSm=%{SYSM:2x}) is unpredictable. Fields:

R unsigned int

R bit.

SYSM unsigned int

SYSm.

ArchMsg.Warning.unpredictable_ccfail_undef

TAGS **UNPREDICTABLE** DISPLAY An instruction %{OPCODE} at VA=%{VADDR} failed its condition codes check but would otherwise have caused a non-data-dependent trap or undefined exception. Behaviour is implementation-defined choice of exception or nop. Fields:

OPCODE unsigned int

opcode of the instruction.

VADDR unsigned int

Virtual address of the instruction being executed.

ArchMsg.Warning.unpredictable_ici_bits

TAGS **UNPREDICTABLE** DISPLAY ICI bits do not apply or are architecturally unpredictable for instruction.

ArchMsg.Warning.unpredictable_t32_breakpoint

TAGS **UNPREDICTABLE** DISPLAY Unpredictable breakpoint on a T32 instruction at %{ADDR1} Offset %{OFFSET} BAS %{BAS} for a %{IS_ADDRESS_MISMATCH_BREAKPOINT:(Mismatch|Match)} Breakpoint number %{N:d} Breakpoint address %{ADDR2}. Fields:

ADDR1 unsigned int

Address.

ADDR2 unsigned int

Breakpoint address.

BAS unsigned int

BAS field.

IS_ADDRESS_MISMATCH_BREAKPOINT bool

Is a Mismatch breakpoint, not a Match breakpoint.

N unsigned int

Breakpoint number.

OFFSET unsigned int

Indicates which halfword of a 32-bit Thumb instruction the breakpoint is set on. 0 for the first halfword or if the instruction is 16 bits wide, or 1 for the second halfword.

ArchMsg.Warning.unpredictable_t32_vector_catch

TAGS **UNPREDICTABLE** DISPLAY Unpredictable Vector catch exception on a T32 instruction at %{ADDR} Offset %{OFFSET}. Fields:

ADDR unsigned int

Address.

OFFSET unsigned int

Indicates which halfword of a 32-bit Thumb instruction the vector catch exception is set on. 0 for the first halfword or if the instruction is 16 bits wide, or 1 for the second halfword.

ArchMsg.Warning.unpredictable_watchpoint_far

TAGS **UNPREDICTABLE** DISPLAY The value for FAR/EDWAR for hitting this watchpoint may be anywhere in the range %{LOWER_BOUND} to %{UPPER_BOUND}. Fields:

LOWER_BOUND unsigned int

the lowest address accessed by the instruction that triggered the watchpoint.

UPPER_BOUND unsigned int

the highest watchpointed address accessed by that instruction.

ArchMsg.Warning.unpredictable_watchpoint_far_isscmp

DISPLAY The value for FAR/EDWAR for hitting this watchpoint may be anywhere in the range %{LOWER_BOUND} to %{UPPER_BOUND}. Fields:

LOWER_BOUND unsigned int

the lowest address accessed by the instruction that triggered the watchpoint.

UPPER_BOUND unsigned int

the highest watchpointed address accessed by that instruction.

ArchMsg.Warning.warning_AdvSIMDEExpandImmUnexpectedZero

TAGS UNPREDICTABLE DISPLAY AdvSIMDEExpandImm may treat this immediate value as UNPREDICTABLE.

ArchMsg.Warning.warning_ConditionalSMC

TAGS UNPREDICTABLE DISPLAY SMC instruction has UNPREDICTABLE effects when conditional and when combined with traps.

ArchMsg.Warning.warning_TLBID_out_of_range

TLBID out of range in TLBI according to the configuration of NIS, NVIS, NOS and NVOS. Fields:

NIS unsigned int

NIS.

NOS unsigned int

NOS.

NVIS unsigned int

NVIS.

NVOS unsigned int

NVOS.

TLBID unsigned int

The TLBID value used in the TLBI instruction.

ArchMsg.Warning.warning_TTL_or_TG_mismatches_current_translation_granule

TTL or TG mismatches current translation granule in TLBI VA/RVA/IPA/RIPA. Fields:

CURRENT_TG unsigned int

Current TG as indicated by TCR_ELx or VTCR_EL2.

TLBI_TG unsigned int

TG (or TTL[3:2]) used in TLBI instruction.

ArchMsg.Warning.warning_access_crosses_page_boundary

TAGS UNPREDICTABLE DISPLAY Access crosses page boundary.

ArchMsg.Warning.warning_access_crosses_page_boundary_has_fault

TAGS UNPREDICTABLE DISPLAY Access that crosses page boundary has fault(s).

ArchMsg.Warning.warning_access_crosses_page_boundary_has_fault_on_both_pages

TAGS **UNPREDICTABLE** DISPLAY Access that crosses page boundary has a fault on each page.

Fields:

ADDR1 unsigned int

FAR address on first page.

ADDR2 unsigned int

FAR address on second page.

FSTATUS1 unsigned int

Fault status on first page.

FSTATUS2 unsigned int

Fault status on second page.

ArchMsg.Warning.warning_access_crosses_page_boundary_spanning_different_memory

TAGS **UNPREDICTABLE** DISPLAY Access at address %{ADDR} crosses page from MEM_TYPE %{MEMTYPE_PAGE1} to %{MEMTYPE_PAGE2}. Fields:

ADDR unsigned int

address of access crossing page.

MEMTYPE_PAGE1 signed int

First Page's memory type (0-Normal, 1-Non-Normal).

MEMTYPE_PAGE2 signed int

Second Page's memory type (0-Normal, 1-Non-Normal).

ArchMsg.Warning.warning_access_wraps_around_memory

TAGS **UNPREDICTABLE** DISPLAY Access wraps around memory in %{WIDTH:d}bit mode.

Fields:

WIDTH unsigned int

Address width.

ArchMsg.Warning.warning_bcr_linking_status

TAGS **UNPREDICTABLE** DISPLAY DBGBCR.{BT2,BT} combination gives unpredictable behaviour.

ArchMsg.Warning.warning_bcr_mask_reserved

TAGS **UNPREDICTABLE** DISPLAY DBGBCR_ELn.MASK is set to a reserved value.

ArchMsg.Warning.warning_bvr_masked_bit_not_zero

TAGS **UNPREDICTABLE** DISPLAY DBGBVRn_EL1.VA contains a bit masked by DBGBCRn_EL1.MASK that is not zero.

ArchMsg.Warning.warning_ccsidr_unimplemented_level

TAGS **UNPREDICTABLE** DISPLAY CCSIDR read while CSSELR %{CSSELR:x} points to an unimplemented cache level. Fields:

CSSELR unsigned int

current effective value of CSSELR.

ArchMsg.Warning.warning_change_to_ns_when_tge_set

TAGS **UNPREDICTABLE** DISPLAY Attempting to change NS to 1 when HCR.TGE = 1.

ArchMsg.Warning.warning_contiguous_bit_check_abort

TAGS **UNPREDICTABLE** DISPLAY An abort occurred whilst attempting to check TLB entry is contiguous at %{ENTRY_ADDR}. Fields:

ENTRY_ADDR unsigned int

address of conflicting TLB entry.

ArchMsg.Warning.warning_contiguous_bit_error

TAGS **UNPREDICTABLE** DISPLAY TLB entry at %{ENTRY_ADDR} was not contiguous with entry at %{CONTIG_ADDR} Contents are %{ENTRY_DATA} but expected %{CONTIG_DATA}. Fields:

CONTIG_ADDR unsigned int

address of the TLB first read, with which this entry is expected to be contiguous.

CONTIG_DATA unsigned int

expected contents based on the entry at CONTIG_ADDR.

ENTRY_ADDR unsigned int

address of conflicting TLB entry.

ENTRY_DATA unsigned int

contents of conflicting TLB entry.

ArchMsg.Warning.warning_contiguous_bit_error_D128

TAGS **UNPREDICTABLE** DISPLAY TLB entry at %{ENTRY_ADDR} was not contiguous with entry at %{CONTIG_ADDR} Contents are %{ENTRY_DATA_HI} %{ENTRY_DATA_LO} but expected %{CONTIG_DATA}. Fields:

CONTIG_ADDR unsigned int

address of the TLB first read, with which this entry is expected to be contiguous.

CONTIG_DATA unsigned int

expected contents based on the entry at CONTIG_ADDR.

ENTRY_ADDR unsigned int

address of conflicting TLB entry.

ENTRY_DATA_HI unsigned int

Higher bit contents of conflicting TLB entry.

ENTRY_DATA_LO unsigned int

Lower bit contents of conflicting TLB entry.

ArchMsg.Warning.warning_cp10_cp11_mismatch

TAGS **UNPREDICTABLE** DISPLAY Unpredictable configuration of CP10 and CP11 controls at EL %{EL:d}. Fields:

EL unsigned int

EL owning the control.

ArchMsg.Warning.warning_cp10_cp11_reserved_value

TAGS **UNPREDICTABLE** DISPLAY Unpredictable configuration of CPACR CP10 ({CP10}) or CP11 ({CP11}). Fields:

CP10 unsigned int

CP10 access.

CP11 unsigned int

CP11 access.

ArchMsg.Warning.warning_csselr_level_out_of_range

TAGS **UNPREDICTABLE** DISPLAY CSSELR.Level out of range, written value {LEVEL} with only {IMPLEMENTED:d} levels implemented. Fields:

IMPLEMENTED unsigned int

number of cache levels implemented.

LEVEL unsigned int

written value of CSSELR.Level.

TYPE enum

The type of cache selected by CSSELR.InD and TnD.

ArchMsg.Warning.warning_debug_flow_control_bits_not_obeyed

TAGS **UNPREDICTABLE** DISPLAY Flow control bits for {REGISTER} were not obeyed, causing {ERROR}. Fields:

ERROR enum

type of error resulting.

REGISTER enum

register under consideration.

ArchMsg.Warning.warning_debug_register_access_during_reset

TAGS **UNPREDICTABLE** DISPLAY Unpredictable {IS_WRITE:(read|write)} access to debug register offset {OFFSET} during reset. Fields:

IS_WRITE unsigned int

Write Not Read.

OFFSET unsigned int

Register Offset.

ArchMsg.Warning.warning_decode_cps_inconsistent_fields

TAGS **UNPREDICTABLE** DISPLAY The mode setting fields in the CPS instruction are inconsistent.

ArchMsg.Warning.warning_decode_invalid_state

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable in current exception-level/security state.

ArchMsg.Warning.warning_default_cacheable_mmu_on

TAGS **UNPREDICTABLE** DISPLAY Default cacheable is enabled (HCR.DC = 1) while MMU is enabled.

ArchMsg.Warning.warning_default_cacheable_vmmu_off

TAGS **UNPREDICTABLE** DISPLAY Default cacheable is enabled (HCR.DC = 1) while HCR.VM = 0.

ArchMsg.Warning.warning_deprecated_wvr_bit_2_set

TAGS **UNPREDICTABLE** DISPLAY DBGWVR%{N:d}_EL1=%{WVR:h} has bit 2 set, deprecated in ARMv8. Fields:

N unsigned int

index.

WVR unsigned int

DBGWVR.

ArchMsg.Warning.warning_eret_while_software_step_active_pending

Missing ISB between setting MDSCR_EL1.SS and ERET TAGS **UNPREDICTABLE**.

ArchMsg.Warning.warning_exclusive_to_non_normal

TAGS **UNPREDICTABLE** DISPLAY Exclusive Access to Non-Normal Memory %{ADDR}. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_exclusive_to_non_writeback

TAGS **UNPREDICTABLE** DISPLAY Exclusive Access to Non-Writeback-Cacheable Memory %{ADDR}. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_execute_from_device_memory

TAGS **UNPREDICTABLE** DISPLAY An attempt was made to execute from Device memory.

ArchMsg.Warning.warning_illegal_cpsr_mode

TAGS **UNPREDICTABLE** DISPLAY Writing an illegal or unimplement mode (%{NEW_MODE:h}) to CPSR was unpredictable. Fields:

NEW_MODE unsigned int

New value of CPSR.M.

ArchMsg.Warning.warning_illegal_srs_mode

TAGS **UNPREDICTABLE** DISPLAY Illegal or **UNPREDICTABLE** mode (%{MODE:h}) used for SRS instruction. Fields:

MODE unsigned int

mode for Banked SP.

ArchMsg.Warning.warning_implementation_defined_read_debug_register_in_SCS

Read Access to Debug registers from the processor is IMPLEMENTATION DEFINED TAGS IMP_DEF. Fields:

PPB_OFFSET unsigned int
debug register offset.

ArchMsg.Warning.warning_implementation_defined_sequential_security_transitions_supported

The behavior of sequential instruction fetches that cross from non-secure to secure memory and contain an SG instruction is CONSTRAINED UNPREDICTABLE TAGS UNPREDICTABLE.

ArchMsg.Warning.warning_implementation_defined_stack_limit_check_supported

It is IMPLEMENTATION DEFINED whether stack pointer limit checking is performed for this instruction TAGS IMP_DEF. Fields:

ADDRESS unsigned int
Address of instruction.

OPCODE unsigned int
instruction opcode.

ArchMsg.Warning.warning_implementation_defined_taggednonsharedmemorytype

TAGS IMP_DEF DISPLAY Memory attribute has Tagged, non shared memory type; it is IMPLEMENTATION DEFINED whether the memory location is treated as Tagged or non-Tagged.

ArchMsg.Warning.warning_implementation_defined_write_debug_register_in_SCS

Write Access to Debug registers from the processor is IMPLEMENTATION DEFINED TAGS IMP_DEF. Fields:

DATA unsigned int
data attempted to be written.

PPB_OFFSET unsigned int
debug register offset.

ArchMsg.Warning.warning_invalid_tcr_granule

TAGS UNPREDICTABLE DISPLAY TCR.TG%{TG_ID:(0|1)} bits have been set to a granule size %{{REQUEST:d}}K not implemented - using %{{SUBSTITUTE:d}}K. Fields:

REQUEST signed int
page size requested (or 0 for reserved).

SUBSTITUTE signed int
best guess available page size.

TG_ID bool
bits TG0 or TG1.

ArchMsg.Warning.warning_load_multiple_user_registers_from_user_mode

TAGS UNPREDICTABLE DISPLAY An LDM(user registers) instruction executed from user mode.

ArchMsg.Warning.warning_mte_in_mem_attr_not_supported

TAGS UNPREDICTABLE DISPLAY Memory attribute %{ATTR} has Tagged memory type but Armv8.5-MTE is not implemented. Fields:

ATTR unsigned int

Memory attribute.

ArchMsg.Warning.warning_reserved_breakpoint_state_match

TAGS UNPREDICTABLE DISPLAY %{IS_BREAKPOINT:(Watchpoint|Breakpoint)} programmed with a reserved combination of HMC, SSC and %{IS_BREAKPOINT:(PAC|PMC)}. Fields:

IS_BREAKPOINT unsigned int

Is a breakpoint, not a watchpoint.

ArchMsg.Warning.warning_shareability

TAGS UNPREDICTABLE DISPLAY Unpredictable: combination of the 1st and 2nd stages of translation is Normal Inner Non-Cacheable, Outer Non-Cacheable.

ArchMsg.Warning.warning_software_step_set_while_enabled

TAGS UNPREDICTABLE DISPLAY MDSCR_EL1.SS set to 1 while software step debug exceptions are enabled.

ArchMsg.Warning.warning_thumb_instruction_wraps_around_memory

TAGS UNPREDICTABLE DISPLAY Thumb instruction wraps around memory.

ArchMsg.Warning.warning_ttbr_sbz_bits_are_not_zero

TAGS UNPREDICTABLE DISPLAY %{TTBR}[%{MSB:d};0] should be zero, but are not. Fields:

MSB unsigned int

x-1.

TTBR enum

which TTBR register.

ArchMsg.Warning.warning_unaligned_address_dbgdtrrx_write

TAGS UNPREDICTABLE DISPLAY Unaligned address %{ADDR} in X0/R0 when DBGDTRRX written in memory access mode. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unaligned_address_dbgdtrtx_read

TAGS UNPREDICTABLE DISPLAY Unaligned address %{ADDR} in X0/R0 when DBGDTRTX read in memory access mode. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unaligned_to_device

TAGS UNPREDICTABLE DISPLAY Unaligned Access to Device Memory %{ADDR}. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unaligned_to_strongly_ordered

TAGS **UNPREDICTABLE** DISPLAY Unaligned Access to Strongly Ordered Memory %{ADDR}.

Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unknown_sau_rnr

TAGS **UNPREDICTABLE** DISPLAY SAU_RNR was set to an unsupported value.

ArchMsg.Warning.warning_unpredictable_AIRCR_PRIS_and_BFHFNMINs

TAGS **UNPREDICTABLE** DISPLAY The effect of setting both AIRCR.BFHFNMINs and AIRCR.PRIS to 1 is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_AIRCR_VECTCLRACTIVE_when_not_in_debug

TAGS **UNPREDICTABLE** DISPLAY The effect of writing a 1 to AIRCR.VECTCLRACTIVE if the processor is not halted in Debug state is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_AIRCR_VECTRESET_and_SYSRESETREQ

TAGS **UNPREDICTABLE** DISPLAY When the processor is halted in Debug state, if a write to the register writes a 1 to both VECTRESET and SYSRESETREQ, the behavior is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_AIRCR_VECTRESET_when_not_in_debug

TAGS **UNPREDICTABLE** DISPLAY The effect of writing a 1 to AIRCR.VECTRESET if the processor is not halted in Debug state is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_AIRCR_incorrect_VKEY

TAGS **UNPREDICTABLE** DISPLAY The value 0x05FA must be written to AIRCR.VKEY, otherwise the register write is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_EXC_RETURN_Reserved_Bit

EXC_RETURN[23:7] are reserved with the special condition that all bits should be written as one. Values other than all 1s are **UNPREDICTABLE**. TAGS **UNPREDICTABLE**. Fields:

EXC_RETURN unsigned int

Special PC value.

SBO_MASK unsigned int

Which bits Should Be One to not UNPRED.

ArchMsg.Warning.warning_unpredictable_PMEVTYPEPn_ELO_VS_reserved_value

TAGS **UNPREDICTABLE** DISPLAY Unpredictable reserved value of PMEVTYPEn<n>_ELO.VS.

ArchMsg.Warning.warning_unpredictable_change_to_DEMCR_MON_STEP_at_insufficient_priority

TAGS **UNPREDICTABLE** DISPLAY The effect of changing DEMCR.MON_STEP at an execution priority that is lower than the priority of the DebugMonitor exception is **CONSTRAINED UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_change_to_priority_of_active_exception

TAGS UNPREDICTABLE DISPLAY Changing the priority of an active exception is UNPREDICTABLE.

ArchMsg.Warning.warning_unpredictable_clear_lspact

TAGS UNPREDICTABLE DISPLAY UNPREDICTABLE state on context switch. Fields:

DRVEXC_TAKEN unsigned int

Derived exception taken, not pended.

REACHED_LAST_STORE unsigned int

Only the last faulted.

ArchMsg.Warning.warning_unpredictable_dbg_exit_unaligned_dlr

TAGS UNPREDICTABLE DISPLAY Debug exit to AArch32 with DLR[0] set to 1. PC[0] can be set to 0 or to DLR[0].

ArchMsg.Warning.warning_unpredictable_exception_catch

TAGS UNPREDICTABLE DISPLAY Generation of an exception catch event is unpredictable.

ArchMsg.Warning.warning_unpredictable_exception_return_inconsistent_state

TAGS UNPREDICTABLE DISPLAY state on exception return is unpredictable.

ArchMsg.Warning.warning_unpredictable_exception_return_instruction

TAGS UNPREDICTABLE DISPLAY exception return instruction is unpredictable.

ArchMsg.Warning.warning_unpredictable_in_debug_state

TAGS UNPREDICTABLE DISPLAY Instruction is UNPREDICTABLE when executed in debug state.

ArchMsg.Warning.warning_unpredictable_mair_encoding

TAGS UNPREDICTABLE DISPLAY Memory attribute %{ATTR} has invalid MAIR encoding; this may lead to unpredictable results. Fields:

ATTR unsigned int

Memory attribute.

ArchMsg.Warning.warning_unpredictable_pmu_counter_access

TAGS UNPREDICTABLE DISPLAY Access to PMU counters from non-secure EL0 or EL1 is UNPREDICTABLE with MDCR_EL2.HPMN set to 0.

ArchMsg.Warning.warning_unpredictable_prioritization_breakpoint_match_vector_catch

TAGS UNPREDICTABLE DISPLAY Two events with the same priority occurred at the same instruction: Address Matching Vector Catch debug event, and Breakpoint debug event. It is constrained unpredictable which is taken.

ArchMsg.Warning.warning_unpredictable_stack_selection

TAGS UNPREDICTABLE DISPLAY UNPREDICTABLE state on context switch.

ArchMsg.Warning.warning_unpredictable_tsize_out_of_range

TAGS UNPREDICTABLE DISPLAY %{TCR}.T%{TTBR:d}SZ (%{TSIZE:d} bits) is out of range. Must be between 25 and %{TSIZE_MAX:d}. Fields:

TCR enum

Which TCR.

TSize signed int

Value of TSize.

TSize_MAX signed int

The maximum allowed value.

TTBR unsigned int

Which TTBR region.

ArchMsg.Warning.warning_unpredictable_unaligned_pc_as_base_register

TAGS **UNPREDICTABLE** DISPLAY The PC value must be word-aligned, otherwise the behavior of the instruction is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_unaligned_pop_stack

TAGS **UNPREDICTABLE** DISPLAY **CONSTRAINED UNPREDICTABLE** unaligned stack frame `%{FRAME_ADDR:x}` in exception-return. Fields:

FRAME_ADDR unsigned int

unaligned stack frame address.

ArchMsg.Warning.warning_unpredictable_unaligned_to_non_writeback

TAGS **UNPREDICTABLE** DISPLAY Unaligned Access to Non-Writeback-Cacheable Memory.

ArchMsg.Warning.warning_unpredictable_vmsa_memattrib

TAGS **UNPREDICTABLE** DISPLAY Unpredictable vmsa memory attribute with texcb value programmed to `%{TEXCB}` when tex remap is `%{TEX_REMAP}`. Fields:

TEXCB unsigned int

texcb value.

TEX_REMAP bool

use tex remap.

ArchMsg.Warning.warning_unpredictable_vtcr_t0sz_sl0

TAGS **UNPREDICTABLE** DISPLAY The combination of VTCR.TOSZ=`%{TOSZ:d}` and VTCR.SL0=`%{SL0:d}` is unpredictable. Fields:

SL0 signed int

VTCR.SL0, starting level for stage 2 translation table walks.

TOSZ signed int

VTCR.TOSZ, required input address range.

ArchMsg.Warning.warning_unpredictable_write_DHCSR

TAGS **UNPREDICTABLE** DISPLAY **UNPREDICTABLE** set of bit-changes in DHCSR. Fields:

NEW unsigned int

data attempted to be written.

OLD unsigned int

previous value.

ArchMsg.Warning.warning_unpredictable_write_SHCSR

TAGS UNPREDICTABLE DISPLAY UNPREDICTABLE setting a pending bit to 1 for an exception with priority \geq execution priority.

ArchMsg.Warning.warning_unsupported_access_to_memory_mapped_register

TAGS UNPREDICTABLE DISPLAY Access to memory mapped register is unsupported.

ArchMsg.Warning.warning_user_jmcr_access

TAGS UNPREDICTABLE DISPLAY User %{WRITE:(write|read)} access to JMCR is unpredictable. Fields:

WRITE unsigned int
access is write.

ArchMsg.Warning.warning_wcr_mask_and_bas

TAGS UNPREDICTABLE DISPLAY DBGWCR_EL1.MASK is non-zero and BAS is not 0xff.

ArchMsg.Warning.warning_wcr_mask_reserved

TAGS UNPREDICTABLE DISPLAY DBGWCR_ELn.MASK is set to a reserved value.

ArchMsg.Warning.warning_wcr_non_configuous_bas

TAGS UNPREDICTABLE DISPLAY DBGWCR_EL1.BAS has non-contiguous set of ones.

ArchMsg.Warning.warning_wvr_masked_bit_not_zero

TAGS UNPREDICTABLE DISPLAY DBGWVRn_EL1.VA contains a bit masked by DBGWCRn_EL1.MASK that is not zero.

ArchMsg.Why.why_branch_target_exception

A branch target exception has occurred. DISPLAY branch target exception: target pc=%{PC}, target opcode=%{OPCODE}. PRIMARY KEY PC. Fields:

BTYPE unsigned int
pstate.BTYPE at time branch taken.

OPCODE unsigned int
opcode at branch target.

PC unsigned int
Virtual Address of branch target.

ArchMsg.Why.why_illegal_state

An illegal mode change has occurred. DISPLAY illegal mode change: %{MESSAGE}. Fields:

MESSAGE string
The message.

ArchMsg.warning_invalid_interval_counter_reload_value

Invalid interval counter reload value in PMSIRR_EL1.INTERVAL: software should set this to a value greater than or equal to the minimum indicated by PMSIDR_EL1.Interval.

ArchMsg.warning_invalid_spe_buffer_write_pointer

Invalid value programmed in PMBPTR_EL1.PTR or PMBLIMITR_EL1.LIMIT: Current write pointer must be at least one sample record below the write limit pointer and PMBPTR_EL1.PTR[63:56] must be equal to PMBLIMITR_EL1.LIMIT[63:56].

BRANCH_MISPREDICT

Simulating branch mispredict. Fields:

PC unsigned int

Origin address (or 0 if unavailable).

BRA_DIR

Direct branches, to immediate address. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISET enum

The instruction set of the branch instruction.

IS_COND bool

Indicates if this is a conditional branch.

IS_HINTED bool

Whether the conditional instruction is hinted.

IS_ISB bool

Whether the instruction is ISB.

IS_LINK bool

Indicates if this is a branch with link.

PC unsigned int

The address of the branch instruction.

TARGET_ISET enum

The instruction set after the branch.

TARGET_PC unsigned int

The address the instruction branches to.

BRA_INDIR

Indirect branches, perhaps to a register. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISSET enum

The instruction set of the branch instruction.

IS_COND bool

Indicates if this is a conditional branch.

IS_HINTED bool

Whether the conditional instruction is hinted.

IS_ISB bool

Whether the instruction is ISB.

IS_LINK bool

Indicates if this is a branch with link.

PC unsigned int

The address of the branch instruction.

TARGET_ISET enum

The instruction set after the branch.

TARGET_PC unsigned int

The address the instruction branches to.

BRA_INDIR_CCFAIL

Indirect branches fail, perhaps to a register.

BRA_RET_CCFAIL

Branch return fail, perhaps to a register.

CACHE_MAINTENANCE_OP

Cache Maintenance Operation. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

DATA unsigned int

Specified MVA or Set/way.

FUNCTION enum

Clean / Invalidate.

SCOPE enum

Affected region.

SIDE enum

Inst / Data.

CCFAIL

Conditional instruction condition check fail. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CCFAIL_UNC

Unconditional instruction condition check fail. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CCPASS

Conditional instruction condition check pass. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CCPASS_UNC

Unconditional instruction condition check pass. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CODE_CACHE_MAINT

Code cache maintenance. Fields:

END_INCL unsigned int

Inclusive end address for VA or PA requests.

NS bool

Non-secure world for PA.

START unsigned int

Start address for VA or PA requests.

TYPE enum

Request type.

COMPILE_BLOCK_END

Last instruction of basic block translated. Fields:

VADDR unsigned int

Address of next instruction after this basic block.

COMPILE_BLOCK_START

First instruction of basic block translated. Fields:

VADDR unsigned int

Address of first instruction in basic block.

COMPILE_INST

ARM instruction compiled. Fields:

DISASS string

Disassembly of the instruction.

ISSET enum

The instruction set of this instruction.

ITSTATE unsigned int

The ITSTATE current for the instruction.

OPCODE unsigned int

The opcode of the instruction.

PC unsigned int

The address of the instruction.

SIZE unsigned int

The size of the instruction in bytes.

CONTEXTIDR

Context ID Register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

NS bool

Secure or nonsecure banked register is accessed.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The new value written.

CONTEXT_SYNC

Called for every context synchronization event. Fields:

ARCH_CSE bool

Architectural Context Synchronization Event.

PC unsigned int

The address of the synchronization instruction.

SYSREG_SYNC bool

System register synchronization event.

CORE_ENDIAN

Core BE8 Big-Endian state changed. Fields:

BE8 bool

Core BE8 Big-Endian state.

CORE_INFO

Static processor attributes. Only triggered by a call to DumpState(). Fields:

ARCH_PROFILE enum

The architecture profile of the core.

CLUSTER_ID unsigned int

The cluster ID of this processor.

CORE_NUM unsigned int

The number of this core in an MP processor.

DETAILED_CORE_FEATURES enum

Detailed core features as a bitfield. Enum values are bitmasks to extract the info from the field.

FPU_VERSION enum

The VFP version implemented by the core.

MEM_ARCH enum

The memory architecture of the core.

MIDR_EL1 unsigned int

MIDR_EL1 value.

NUM_CORES unsigned int

The number of cores in this MP processor.

QUANTUM_SIZE unsigned int

The default quantum size of the core (in ticks).

SECURITY_FEATURES bool

Does the core have security features?.

CORE_LOADS

Processor load accesses. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ACQREL enum

Is this an acquire/release.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DATA unsigned int

The data read or written.

ELEMENT_SIZE unsigned int

Width of each element.

ENDIANNESS enum

Byte ordering of this access.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

LOCK enum

Normal, exclusive or locked access.

MECID unsigned int

Memory Encryption Context ID, if applicable.

MECID2 unsigned int

Memory Encryption Context ID of the second page, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PADDR_TAG unsigned int

If tag checked, the physical address tag for the access.

PAGE_SIZE unsigned int

Page size as $\log_2(\text{size in bytes})$.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK, 2=Exclusive Failed, 3=Unsupported.

SIZE unsigned int

Width of the access in bytes. Only required if DATA is not traced.

TAG_CHECKED bool

Is this access tag checked.

TAG_CHECKED2 bool

Is the second page access tag checked.

TRANS bool

Is this a translated access.

VADDR unsigned int

The virtual address of the access.

CORE_REGS

Changes of the core registers R0 to R14. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number, 0 to 14.

MODE enum

Bank of the register accessed.

OLD_VALUE unsigned int

The old value overwritten.

PHYS_ID enum

The physical register accessed.

VALUE unsigned int

The new value written to the register.

CORE_REGS64

Changes of the core registers X0..X30, SP_ELn. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ID enum

The register number, 0-30 for X0-X30, >=32 for SP_ELn.

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

CORE_REGS64_READ

General purpose AArch64 register has been read X0..X30, SP_ELn. Fields:

ID enum

The register number, 0-30 for X0-X30, >=32 for SP_ELn.

VALUE unsigned int

The value read from the register.

CORE_STALLED_TRANSITION

Detect when a core is prevented from progressing PChannel states. Fields:

CURRENT_STATE enum

Current power mode of core.

REASON string

The reason a transition was stalled, and what must be done to unblock it.

TARGET_STATE enum

Desired power mode of core after transition.

CORE_STORES

Processor store accesses. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ACQREL enum

Is this an acquire/release.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DATA unsigned int

The data read or written.

ELEMENT_SIZE unsigned int

Width of each element.

ENDIANNESS enum

Byte ordering of this access.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

LOCK enum

Normal, exclusive or locked access.

MECID unsigned int

Memory Encryption Context ID, if applicable.

MECID2 unsigned int

Memory Encryption Context ID of the second page, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PADDR_TAG unsigned int

If tag checked, the physical address tag for the access.

PAGE_SIZE unsigned int

Page size as $\log_2(\text{size in bytes})$.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK, 2=Exclusive Failed, 3=Unsupported.

SIZE unsigned int

Width of the access in bytes. Only required if DATA is not traced.

TAG_CHECKED bool

Is this access tag checked.

TAG_CHECKED2 bool

Is the second page access tag checked.

TRANS bool

Is this a translated access.

VADDR unsigned int

The virtual address of the access.

CP14_READ

System Coprocessor register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP14 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc1 unsigned int

opcode 1.

opc2 unsigned int

opcode 2.

CP14_WRITE

System Coprocessor register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP14 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc1 unsigned int

opcode 1.

opc2 unsigned int

opcode 2.

CP15_READ

System Coprocessor register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP15 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc1 unsigned int

opcode 1.

opc2 unsigned int

opcode 2.

CP15_READ64

System Coprocessor register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP15 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc unsigned int

opcode 1.

CP15_WRITE

System Control Coprocessor register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP15 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc1 unsigned int

opcode 1.

opc2 unsigned int

opcode 2.

CP15_WRITE64

System Coprocessor register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP15 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc unsigned int

opcode 1.

CPSR

CPSR change. (In AArch64 PSTATE represented in SPSR format). Fields:

CORE_NUM unsigned int

Core number in a multi processor.

OLD_VALUE unsigned int

The old CPSR value.

UNKNOWN unsigned int

Bits within the register that have unknown value.

VALUE unsigned int

The new CPSR value.

CRYPTO_SPEC

Every crypto instruction speculatively executed.

CorePowerStateChange

Core power-up state has changed. DISPLAY Core powered %{PU:(up|down)}. Fields:

PU bool

Whether the core powered up.

DATA_CACHE_SET_ALLOC_TAGS

Set allocation tags from DC GVA instruction. Fields:

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

NSDESC unsigned int

The security state of the access.

PADDR unsigned int

The physical address of the access.

PAS enum

The physical address space of the page.

RESPONSE enum

0=Aborted, 1=OK.

SIZE unsigned int

The size of memory in bytes.

TAG unsigned int

The value of the tag being set.

TAGGING_TYPE enum

Type of Memory tagging (Physical or Virtual).

TAG_VADDR unsigned int

The virtual address of the Virtual Tag if Virtual tagging is used; otherwise 0.

VADDR unsigned int

Virtual address used for setting allocation tags.

DATA_CACHE_ZERO

Zero and invalidate cache line from DC ZVA instruction. Fields:

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

NSDESC unsigned int

The security state of the access.

PADDR unsigned int

The physical address of the access.

PADDR_TAG unsigned int

If tag checked, the physical address tag for the access.

PAS enum

The physical address space of the page.

RESPONSE enum

0=Aborted, 1=OK.

SIZE unsigned int

The size of zero'd memory in bytes.

TAG_CHECKED bool

Is this access tag checked.

VADDR unsigned int

Virtual address of the zero'd cache line.

DEBUG_EVENT

Hardware debug support event. Fields:

EVENT enum

Description of event.

VALUE unsigned int
data value.

DEBUG_TARGET_ISET

Hardware debug target ISA. Fields:

TARGET_ISET enum
The target instruction set of the debug state.

DMI_ALLOCATE

DMI allocated. Fields:

ACCESS_DIRECTION enum
READ / WRITE.

PADDR unsigned int
The physical address of the DMI.

PTR unsigned int
DMI range pointer.

SIDE enum
DSIDE / ISIDE.

SIZE unsigned int
The size of DMI.

VADDR unsigned int
The virtual address of the DMI..

DMI_HIT

DMI hit, access used DMI (only fire for small DMI regions i.e. less than a full page). Fields:

ACCESS_DIRECTION enum
READ / WRITE.

PADDR unsigned int
The physical address of the DMI.

SIDE enum
DSIDE / ISIDE.

SIZE unsigned int
The size of the DMI.

VADDR unsigned int
The virtual address of the DMI..

DMI_REVOKE

DMI revoked. Fields:

ACCESS_DIRECTION enum
READ / WRITE.

PADDR unsigned int

The physical address of the DMI.

SIDE enum

DSIDE / ISIDE.

SIZE unsigned int

The size of DMI.

VADDR unsigned int

The virtual address of the DMI..

END_COMPILE

Compilation end. Fields:

END_OF_PAGE_COUNT unsigned int

Number of basic blocks exited due to page boundary since START_COMPILE.

FETCHFAIL_COUNT unsigned int

Number of basic blocks exited due to ifetch failure START_COMPILE.

INST_COUNT unsigned int

Number of instructions compiled since START_COMPILE.

NONSEQ_COUNT unsigned int

Number of basic blocks exited due to non-sequential instructions since START_COMPILE.

PAGE_STRADDLE_COUNT unsigned int

Number of basic blocks exited due to unaligned instructions crossing page since START_COMPILE.

ESB

Error Synchronization Barrier hit. Fields:

ASYNC bool

ESB is async.

IMPLICIT bool

ESB is implicit.

EXCEPTION

Exception taken. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ESR unsigned int

The value assigned to ESR when exception is taken.

ESR_EC unsigned int

The Value assigned to ESR.EC when exception is taken.

IS_PHYSICAL bool

Physical or Virtual exception.

IS_TRAP bool

Exception coming from a trap.

LR unsigned int

The value assigned to the link register.

PC unsigned int

The location where the exception occurred.

PREFERRED_RETURN unsigned int

The preferred return address for the exception.

TAKEN_LOCALLY bool

Whether exception is taken locally(same exception level).

TARGET_ISET enum

The instruction set of the exception handler code.

TARGET_PC unsigned int

The address the exception branches to.

VECTOR enum

The exception vector.

VECTOR_OFFSET unsigned int

The offset of an exception vector from the base address.

EXCEPTION_END

Every exception completed.

EXCEPTION_INFO

Trace human-readable reason of the current exception for data/instruction aborts, floating point exceptions and unknown exceptions. Fields:

DESCRIPTION string

Reason for current exception.

ESR_EC unsigned int

The value of ESR.EC when exception is taken.

EXCEPTION_RAISE

Every exception raised.

EXCEPTION_RETURN

Branches on leaving exception. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISSET enum

The instruction set of the branch instruction.

IS_COND bool

Indicates if this is a conditional branch.

IS_HINTED bool

Whether the conditional instruction is hinted.

IS_ISB bool

Whether the instruction is ISB.

IS_LINK bool

Indicates if this is a branch with link.

PC unsigned int

The address of the branch instruction.

TARGET_ISET enum

The instruction set after the branch.

TARGET_PC unsigned int

The address the instruction branches to.

EXCEPTION_RETURN_PREBRANCH

Exception return event that must occur before PC and processor state have updated.

EXCEPTION_START

Every exception started.

FIQ_TAKEN

FIQ taken exception.

FP_EXCEPTIONS_SIGNALLED

Trace which FP exceptions were signalled as a part of the FP operations performed by an instruction. Fields:

FP_EXCEPTIONS_SIGNALLED_MASK unsigned int

Indicates which FP exceptions were signalled by an instruction. This uses the same encoding as the cumulative floating-point exception bitfields in FPSR.

FP_STATE

Floating point state. Fields:

FPSR unsigned int

FPSR register value.

MASK unsigned int

The mask indicating updated cumulative bits.

FREQ_CHANGED

Core frequency changed. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The instruction count of this core.

NEW_FREQ signed int

The new frequency of this core (expressed in Hz).

OLD_FREQ signed int

The old frequency of this core (expressed in Hz).

HLT

HLT instruction occurred. Fields:

IMM16 unsigned int

The 16-bit immediate value encoded in HLT instruction.

INST

Every instruction executed. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DEBUG_STATE bool

The instruction is executed in debug state.

DISASS string

Disassembly of instruction.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISSET enum

The current instruction set.

ITSTATE unsigned int

The current ITSTATE.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

MODE enum

The mode the core is in.

NS unsigned int

The core's non-secure bit.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

OPCODE unsigned int

The opcode of the instruction.

PADDR unsigned int

The physical address of the instruction.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the instruction.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the page.

PC unsigned int

The address of the instruction.

SECURITY_STATE enum

The core's security state.

SIZE unsigned int

The size of the instruction in bytes.

INST_END

Every instruction completed.

INST_START

Every instruction started. Fields:

DEBUG_STATE bool

The instruction is executed in debug state.

ISSET enum

The current instruction set.

MODE enum

The mode the core is in.

NS enum

The current Secure State.

PC unsigned int

The address of the conditional instruction.

SECURITY_STATE enum

The current Security State.

INST_STRADDLE

Instruction straddles boundary. Fields:

DEBUG_STATE bool

The instruction is executed in debug state.

ISET enum

The current instruction set.

MODE enum

The mode the core is in.

NS enum

The current Secure State.

PC unsigned int

The address of the conditional instruction.

SECURITY_STATE enum

The current Security State.

IRQ_TAKEN

IRQ taken exception.

ISB

ISB instruction occurred. Fields:

PC unsigned int

The address of the ISB instruction.

LOAD_MULTIPLE_REGS_LIST

List of destination registers (not source) of a multiple registers load instruction. Fields:

REG string

Register to add into the list.

LOCAL_MONITOR

Local monitor activity. Fields:

PADDR unsigned int

Local Monitor Address.

State enum

State of the monitor (Open/Exclusive).

MAPPING_CREATE

A memory mapping has been created. This is an implementation detail of the model and so may change without notice. Fields:

ASID unsigned int

The ASID used during translation, or 0 if no ASID was used.

IS_VALID bool

Whether the mapping is valid or not.

PADDR unsigned int

Base physical address of region, or 0 if not known.

SIDE enum

I-side or D-side.

SIZE unsigned int

Size of the page, or 0 if not known.

VADDR unsigned int

Base virtual address of region.

VMID unsigned int

The VMID used during translation, or 0 if no VMID was used.

MAPPING_DESTROY

A memory mapping has been destroyed. This is an implementation detail of the model and so may change without notice. Fields:

ASID unsigned int

The ASID used during translation.

CAUSE enum

The cause of invalidation.

PADDR unsigned int

Base physical address of page.

SIDE enum

I-side or D-side.

SIZE unsigned int

Size of the page.

VADDR unsigned int

Base virtual address of page.

VMID unsigned int

The VMID used during translation.

MEMMAP_DEBUG_READ

Memory mapped reads to the debug registers. Fields:

ADDR unsigned int

address.

CORE_NUM unsigned int

Core number in a multi processor.

EXT bool

Whether access is from an external device (such as the DAP).

NS enum

Secure state of the access.

REG_NAME string

Name of the debug register accessed.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

MEMMAP_DEBUG_WRITE

Memory mapped writes to the debug registers. Fields:

ADDR unsigned int

address.

CORE_NUM unsigned int

Core number in a multi processor.

EXT bool

Whether access is from an external device (such as the DAP).

NS enum

Secure state of the access.

REG_NAME string

Name of the debug register accessed.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The new value written.

MEMTAG_LOADS

Allocation tag memory read; when attributes and system configuration allow tag accesses. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

NUM_GRANULES unsigned int

Number of tags to be Loaded/stored.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK.

TAGGING_TYPE enum

Type of Memory tagging (Physical or Virtual).

TAGS unsigned int

The tags read or written. Tags for subsequent granules would be found at subsequent bytes in this field (One byte for each tag).

TAG_VADDR unsigned int

The virtual address of the Virtual Tag if Virtual tagging is used; otherwise 0.

VADDR unsigned int

The virtual address of the access.

MENTAG_LOAD_INST

Allocation tag load instruction; regardless of tags being actually accessed. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate

(Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

NUM_GRANULES unsigned int

Number of tags to be Loaded/stored.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK.

TAGGING_TYPE enum

Type of Memory tagging (Physical or Virtual).

TAGS unsigned int

The tags read or written. Tags for subsequent granules would be found at subsequent bytes in this field (One byte for each tag).

TAG_VADDR unsigned int

The virtual address of the Virtual Tag if Virtual tagging is used; otherwise 0.

VADDR unsigned int

The virtual address of the access.

MEMTAG_STORES

Allocation tag memory write; when attributes and system configuration allow tag accesses. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

NUM_GRANULES unsigned int

Number of tags to be Loaded/stored.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK.

TAGGING_TYPE enum

Type of Memory tagging (Physical or Virtual).

TAGS unsigned int

The tags read or written. Tags for subsequent granules would be found at subsequent bytes in this field (One byte for each tag).

TAG_VADDR unsigned int

The virtual address of the Virtual Tag if Virtual tagging is used; otherwise 0.

VADDR unsigned int

The virtual address of the access.

MEMTAG_STORE_INST

Allocation tag store instruction; regardless of tags being actually accessed. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes

([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

NUM_GRANULES unsigned int

Number of tags to be Loaded/stored.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK.

TAGGING_TYPE enum

Type of Memory tagging (Physical or Virtual).

TAGS unsigned int

The tags read or written. Tags for subsequent granules would be found at subsequent bytes in this field (One byte for each tag).

TAG_VADDR unsigned int

The virtual address of the Virtual Tag if Virtual tagging is used; otherwise 0.

VADDR unsigned int

The virtual address of the access.

MMU_D128_TTB_READ

This event is triggered by reads caused by a D128 translation table walk. Fields:

ABORTED bool

Was the walk successful.

DATA_HIGH unsigned int

The data upper bits read.

DATA_LOW unsigned int

The data lower bits read.

IPA unsigned int

For stage 1, the IPA of the read.

LEVEL unsigned int

Translation table level.

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

OUTADDR unsigned int

The output physical address.

PADDR unsigned int

The physical address of the read.

PASmem enum

Physical Address Space of the accessed memory.

PASreq enum

Physical Address Space of the requested lookup.

REGIME_EL enum

Entry matches in this translation regime.

REQUESTER enum

The requester of this table walk.

SIDE enum

Inst / Data.

STAGE unsigned int

Translation stage.

MMU_D128_TTB_WRITE

This event is triggered by writes caused by a D128 translation table walk. Fields:

ABORTED bool

Was the walk successful.

DATA_HIGH unsigned int

The data upper bits written.

DATA_LOW unsigned int

The data lower bits written.

IPA unsigned int

For stage 1, the IPA of the write.

IS_HACDBS_ACCESS bool

If the access was due to HACDBS_CLEAN.

LEVEL unsigned int

Translation table level.

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

OUTADDR unsigned int

The output physical address.

PADDR unsigned int

The physical address of the write.

PASmem enum

Physical Address Space of the accessed memory.

PASreq enum

Physical Address Space of the requested lookup.

REGIME_EL enum

Entry matches in this translation regime.

SIDE enum

Inst / Data.

STAGE unsigned int

Translation stage.

MMU_TRANS

Address translation information. Fields:

ASID unsigned int

Address space identifier.

CORE_NUM unsigned int

Core number in a multi processor.

Hyp bool

Entry matches in Hyp state only.

INNERCACHE_RA bool

Is the inner cache allocate on read.

INNERCACHE_TRANSIENT bool

Is the inner write-through transient.

INNERCACHE_TYPE enum

Inner Caching scheme (NC/MB/WA).

INNERCACHE_WA bool

Is the inner cache allocate on write.

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MEMTYPE enum

Memory type.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC enum

Is secure side supposed to access secure or nonsecure memory.

OUTERCACHE_RA bool

Is the outer cache allocate on read.

OUTERCACHE_TRANSIENT bool

Is the outer write-through transient.

OUTERCACHE_TYPE enum

Outer Caching scheme (NC/MB/WA).

OUTERCACHE_WA bool

Is the outer cache allocate on write.

PADDR unsigned int

Physical address of the access.

PAGESIZE unsigned int

Page size as $\log_2(\text{size})$.

PAS enum

PAS supposed to be accessed is secure or nonsecure memory or Root/Realm if RME is implemented.

SH enum

Shareability.

SIDE enum

Inst / Data.

STAGE1_PERM unsigned int

Stage 1 permissions, [5:3] Privileged access (XWR) [2:0] User access (XWR).

STAGE2_PERM unsigned int

Stage 2 permission mask [5:3] Privileged access (XWR) [2:0] User access (XWR).

VADDR unsigned int

Virtual address of the access.

VMID unsigned int

Virtual machine identifier.

XS bool

XS Attribute.

nG enum

Flag indicating whether ASID will be matched.

MMU_TTB_READ

This event is triggered by reads caused by a translation table walk. Fields:

ABORTED bool

Was the walk successful.

DATA unsigned int

The data read.

DESC_KIND enum

The encoding scheme of the descriptors used for this walk.

IPA unsigned int

For stage 1, the IPA of the read.

LEVEL unsigned int

Translation table level.

LPAE bool

Is this for an LPAE translation (*DEPRECATED*, use DESC_KIND).

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSmem bool

Non secure memory access.

NSreq bool

Non secure request.

OUTADDR unsigned int

The output physical address.

PADDR unsigned int

The physical address of the read.

PASmem enum

Physical Address Space of the accessed memory.

PASreq enum

Physical Address Space of the requested lookup.

REGIME_EL enum

Entry matches in this translation regime.

REQUESTER enum

The requester of this table walk.

SIDE enum

Inst / Data.

STAGE unsigned int

Translation stage.

MMU_TTB_WRITE

This event is triggered by writes caused by a translation table walk. Fields:

ABORTED bool

Was the walk successful.

DATA unsigned int

The data written.

DESC_KIND enum

The encoding scheme of the descriptors used for this walk.

IPA unsigned int

For stage 1, the IPA of the write.

IS_HACDBS_ACCESS bool

If the access was due to HACDBS_CLEAN.

LEVEL unsigned int

Translation table level.

LPAE bool

Is this for an LPAE translation (*DEPRECATED*, use DESC_KIND).

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSmem bool

Non secure memory access.

NSreq bool

Non secure request.

OUTADDR unsigned int

The output physical address.

PADDR unsigned int

The physical address of the write.

PASmem enum

Physical Address Space of the accessed memory.

PASreq enum

Physical Address Space of the requested lookup.

REGIME_EL enum

Entry matches in this translation regime.

SIDE enum

Inst / Data.

STAGE unsigned int

Translation stage.

MODE_CHANGE

Mode change. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

MODE enum

The new mode.

NON_SECURE enum

The core's new non-secure bit.

OLD_MODE enum

The old mode.

SECURITY_STATE enum

The new security state.

MOPS_LD_RETIRED

Load instructions architecturally executed for MOPS.

MOPS_ST_RETIRED

Store instructions architecturally executed for MOPS. Fields:

IS_MEMCPY bool

Whether the memory operation is MEMCPY.

NOT_TAG_CHECKED_INST

MTE Tag not checked for current instruction.

PERIODIC

Called for every quantum. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The instruction count of this CPU.

PC unsigned int

The address of the next instruction to be executed on this CPU.

PMU_COUNTER_OVERFLOW

PMU counter overflow. Fields:

COUNTER_GROUP enum

Counter group.

COUNTER_RANGE enum

Counter range.

EVENT_ID unsigned int

Event Identifier, field is valid only for Event counter group.

INDEX unsigned int

Counter index (as selected by PMSELR), field is valid only for Event/Cycle counter groups.

INTERRUPT bool

Is interrupt enabled on overflow for this counter.

POINTER_AUTH

Called when execution hits a pointer authentication check. Fields:

AUTH_STATE *bool*

Authentication result.

KEY_USED *enum*

Pointer authentication key.

POTINDEX_UPDATE

Value of POT index for current pc changed. Fields:

NEW_POTINDEX *unsigned int*

New POTIndex.

OLD_POTINDEX *unsigned int*

Old POTIndex.

PREFETCH_MEMORY64

Prefetch from PRFM or PRFUM instructions. Fields:

PRFOP *enum*

Prefetch hint.

VADDR *unsigned int*

Virtual address of the location that should be prefetched.

PRELOAD64_REQUEST

Preload requests from PRFM or PRFUM instructions. Fields:

PRFOP *enum*

Prefetch hint.

VADDR *unsigned int*

Virtual address of the location that should be prefetched.

PRELOAD_DATA

Data preload from PLD instruction. Fields:

VADDR *unsigned int*

Virtual address of the data that should be preloaded.

PRELOAD_INST

Instruction preload from PLI instruction. Fields:

VADDR *unsigned int*

Virtual address of the instruction that should be preloaded.

PRE_CORE_LOAD

Trace just before a core load. Fields:

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

PRE_CORE_STORE

Trace just before a core store. Fields:

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

PRE_TTB_READ

This event is triggered before reads caused by a translation table walk. Fields:

ADDR unsigned int

The physical address of the read.

PSTATE

PSTATE fields update. Fields:

ALLINT unsigned int

All interrupt mask bit.

BTYPE unsigned int

Branch target identification bit.

CORE_NUM unsigned int

Core number in a multi processor.

DAIF unsigned int

Exception mask bits.

DIT unsigned int

Data Independent Timing bit.

EL unsigned int

Current Exception level field.

EXLOCK unsigned int

Exception return state lock bit.

IL unsigned int

Illegal Execution state bit.

NZCV unsigned int

Condition flags.

PACM unsigned int

PACM bit.

PAN unsigned int

Privileged Access Never state bit.

PM unsigned int

PMU exception mask bit.

PPEND unsigned int

PMU exception pending bit.

SM unsigned int

StreamingMode status bit.

SP unsigned int

Stack pointer register selection bit.

SS unsigned int

Software Step bit.

SSBS unsigned int

Speculative Store Bypass Safe bit.

TCO unsigned int

Tag Check Override bit.

TOUCHED unsigned int

Mask of the fields that have been touched since this trace has last been emitted. Note that touched does not necessarily mean the value changed. The index of the field is the bit index in the mask plus the index of this field plus one.

UAO unsigned int

User Access Override bit.

UINJ unsigned int

Undefined Instruction exceptions bit.

ZA unsigned int

ZA storage and ZT0 register enable bit.

nRW unsigned int

Current Execution state bit.

PSTATE_PACTIVE_OPMODE_TRANSITION

Trace pactive and Opmode mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Operating mode after the transition.

OLD_STATE enum

Operating mode before the transition.

PSTATE_PACTIVE_OPMODE_TRANSITION_DENIED

Trace the denied pactive and Opmode mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Operating mode after the transition.

OLD_STATE enum

Operating mode before the transition.

REASON string

Reason for the transition being denied.

PSTATE_PACTIVE_POWER_TRANSITION

Trace pactive and power mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Power mode after the transition.

OLD_STATE enum

Power mode before the transition.

PSTATE_PACTIVE_POWER_TRANSITION_DENIED

Trace the denied pactive and power mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Power mode after the transition.

OLD_STATE enum

Power mode before the transition.

REASON string

Reason for the transition being denied.

PSTATE_PACTIVE_THREAD_OPMODE_TRANSITION

Trace pactive and Opmode mode transitions for threads on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Operating mode after the transition.

OLD_STATE enum

Operating mode before the transition.

QUANTUM_START

Called at the start of every quantum.

RANGE_PRELOAD_REQUEST

Preload requests from RPRFM. Fields:

COUNT unsigned int

Number of blocks of data to access.

LENGTH unsigned int

Number of contiguous bytes to be accessed in each block.

OP unsigned int

Range prefetch operation .

REUSE unsigned int

Maximum number of bytes to be accessed before next RPRFM .

STRIDE signed int

Stride to advance after accessing LENGTH bytes.

VADDR unsigned int

Base virtual address of range prefetch request.

RUN_STATE

Run state transition. Fields:

INST_COUNT unsigned int

Ticks count at point of transition.

NEW enum

New run state.

OLD enum

Old run state.

SBP_SAMPLE_COLLISION

DISPLAY Statistical profiling samples collided .

SBP_SAMPLE_DISCARDED

DISPLAY Operation discarded by statistical profiling unit when configured in discard mode.
Fields:

BUFFER_POINTER unsigned int

Value of buffer pointer.

DATA unsigned int

Data discarded.

SIZE unsigned int

Size of sample, in bytes.

SBP_SAMPLE_END

DISPLAY Statistical profiling of an operation ended.

SBP_SAMPLE_POP

DISPLAY Operation which might be sampled by statistical profiling.

SBP_SAMPLE_RECORDED

DISPLAY Operation sampled by statistical profiling has been written to the buffer at %{BUFFER_ADDRESS} with size %{SIZE} and data %{DATA}. Fields:

BUFFER_ADDRESS unsigned int

Virtual Address of start of buffer write.

DATA unsigned int

Data written to the buffer.

SIZE unsigned int

Size of sample written to buffer, in bytes.

SBP_SAMPLE_START

DISPLAY Statistical profiling of an operation started.

SBP_SAMPLE_TAKEN

DISPLAY Operation sampled by statistical profiling after filtering.

SEMIHOSTING_CALL

Call of a semihost function occurred. Fields:

PC unsigned int

The program counter after the semihosting call.

SEMIHOSTING_PRECALL

About to call semihost. Fields:

EL2 bool

Use EL2 translation regime.

NS bool

Is Non-Secure.

REG_WIDTH unsigned int

The current register width in bytes.

SIGNAL

External signal state change. Fields:

SIGNAL enum

Signal that changed.

STATE bool

Signal asserted state.

SOFTWARE_STEP

Return the Debug Software Step state. Fields:

STATE enum

Software Step state.

SPSR

SPSR change. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

MODE enum

Which of the banked SPSR registers is written.

OLD_VALUE unsigned int

The old SPSR value.

VALUE unsigned int

The new SPSR value.

START_COMPILE

Compilation started. Fields:

VADDR unsigned int

Instruction where compilation begins.

STG_MULTIPLE_MICRO_OPS_ADDRESS_RANGE

Address range for MTE stores with multiple uops. Fields:

ADDR_END unsigned int

Final address.

ADDR_START unsigned int

Starting address.

STORE_MULTIPLE_REGS_ADDRESS_RANGE

Address range for stores with multiple registers. Fields:

ADDR_END unsigned int

Final address.

ADDR_START unsigned int

Starting address.

SUBOPTIMAL_LDST_RETIRE

suboptimal load/store. Fields:

ACCESS_SIZE unsigned int

Log2 of the access width in bytes used for alignment checking (1: half-word, 2: word ...).

OPTIMAL_ALIGN_SELECTOR unsigned int

Selector to decide the optimal alignment size (1: access width, 2: 4-bytes, 3: 8-bytes, ..., 12: 4-kbytes).

VADDR unsigned int

The virtual address of the access.

SVE_INST_SPEC

SVE operations speculatively executed.

SVE_LD_RETIRED

SVE load instructions architecturally executed.

SVE_LOADS

SVE load accesses. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ACQREL enum

Is this an acquire/release.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DATA unsigned int

The data read or written.

ELEMENT_SIZE unsigned int

Width of each element.

ENDIANNESS enum

Byte ordering of this access.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

LOCK enum

Normal, exclusive or locked access.

MECID unsigned int

Memory Encryption Context ID, if applicable.

MECID2 unsigned int

Memory Encryption Context ID of the second page, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PADDR_TAG unsigned int

If tag checked, the physical address tag for the access.

PAGE_SIZE unsigned int

Page size as $\log_2(\text{size in bytes})$.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK, 2=Exclusive Failed, 3=Unsupported.

SIZE unsigned int

Width of the access in bytes. Only required if DATA is not traced.

TAG_CHECKED bool

Is this access tag checked.

TAG_CHECKED2 bool

Is the second page access tag checked.

TRANS bool

Is this a translated access.

VADDR unsigned int

The virtual address of the access.

SVE_MEM_PREFETCH

Scalable Vector Extension memory prefetch. Fields:

CACHE_LEVEL enum

The preferred cache level.

HINT enum

What the memory is expected to be used for.

SIZE unsigned int

The prefetch size in bytes.

TEMPORAL bool

Whether the data is temporal.

VADDR unsigned int

The virtual address of the prefetch.

SVE_REG_READ

Scalable Vector Extension register read. Fields:

ID enum

The register identifier.

SM bool

Whether the PE is in Streaming Mode.

VALUE unsigned int

The read data.

SVE_REG_UPDATE

Scalable Vector Extension register update. Fields:

ID enum

The register identifier.

SM bool

Whether the PE is in Streaming Mode.

VALUE unsigned int

The written data.

SVE_REG_WRITE

Scalable Vector Extension register write. Fields:

ID enum

The register identifier.

SM bool

Whether the PE is in Streaming Mode.

VALUE unsigned int

The written data.

SVE_STORES

SVE store accesses. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ACQREL enum

Is this an acquire/release.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DATA unsigned int

The data read or written.

ELEMENT_SIZE unsigned int

Width of each element.

ENDIANNESS enum

Byte ordering of this access.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

LOCK enum

Normal, exclusive or locked access.

MECID unsigned int

Memory Encryption Context ID, if applicable.

MECID2 unsigned int

Memory Encryption Context ID of the second page, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PADDR_TAG unsigned int

If tag checked, the physical address tag for the access.

PAGE_SIZE unsigned int

Page size as $\log_2(\text{size in bytes})$.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK, 2=Exclusive Failed, 3=Unsupported.

SIZE unsigned int

Width of the access in bytes. Only required if DATA is not traced.

TAG_CHECKED bool

Is this access tag checked.

TAG_CHECKED2 bool

Is the second page access tag checked.

TRANS bool

Is this a translated access.

VADDR unsigned int

The virtual address of the access.

SVE_ST_RETIRED

SVE store instructions architecturally executed.

SVE_UNALIGNED_LDST_RETIRED

SVE load or store instructions architecturally executed that would cause alignment fault if A bit is set.

SVE_Z_PRED_MEM_ACCESS_INFO

Information about SVE/SME predicated accesses to Z registers (non Gather/Scatter). Fields:

MEM_ELEM_SIZE unsigned int

Width of elements accessed in memory (in bytes).

RnW bool

Whether this is a Read not a Write.

VA unsigned int

Virtual base address computed for accesses.

SYNC

Called for every synchronization, that is every quantum break. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The instruction count of this CPU.

LOCAL_QUANTUM unsigned int

The local quantum of this CPU.

LOCAL_TIME unsigned int

The local time of this CPU.

SYNC_REASON enum

Reason for sync.

SYSCALL

System call instruction executed. Fields:

IMM unsigned int

Immediate value of the system call instruction.

TYPE enum

System call type.

VADDR unsigned int

Instruction that caused the system call.

SYSREG_READ128

System Coprocessor 128 bit register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

REGNUM enum

Internal register number.

REGNUM_PHYS enum

Internal register number physically accessed (e.g. in presense of VHE redirection).

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_READ64

System Coprocessor register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REGNUM enum

Internal register number.

REGNUM_OPERAND enum

Register number corresponding to the instruction operand.

REGNUM_PHYS enum

Internal register number physically accessed (e.g. in presense of VHE redirection).

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_UPDATE128

Triggers when the system updates a register. Fields:

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG enum

Register number.

UNKNOWN unsigned int

Bits of the register which became unknown.

VALUE unsigned int

Value written to the register.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_UPDATE32

Triggers when the system updates a register. Fields:

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG enum

Register number.

UNKNOWN unsigned int

Bits of the register which became unknown.

VALUE unsigned int

Value written to the register.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int
opcode 2.

SYSREG_UPDATE64

Triggers when the system updates a register. Fields:

CRm unsigned int
CRm.

CRn unsigned int
CRn.

NS bool
Secure or nonsecure banked register is accessed.

REG enum
Register number.

UNKNOWN unsigned int
Bits of the register which became unknown.

VALUE unsigned int
Value written to the register.

opc unsigned int
opcode 1.

opc0 unsigned int
opcode 0.

opc2 unsigned int
opcode 2.

SYSREG_WRITE128

System Coprocessor 128 bit register write. Fields:

CORE_NUM unsigned int
Core number in a multi processor.

CRm unsigned int
CRm.

CRn unsigned int
CRn.

REGNUM enum
Internal register number.

REGNUM_PHYS enum
Internal register number physically accessed (e.g. in presense of VHE redirection).

UNDEF bool
The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_WRITE64

System Coprocessor register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REGNUM enum

Internal register number.

REGNUM_OPERAND enum

Register number corresponding to the instruction operand.

REGNUM_PHYS enum

Internal register number physically accessed (e.g. in presense of VHE redirection).

UNDEF bool

The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

TRBU_BUFFER_WRAP

DISPLAY Trace buffer current write pointer wrapped.

TRBU_TRIGGER_EVENT

DISPLAY Trace buffer Trigger Event.

TRBU_WRITE

DISPLAY PETU trace data has been written to the TRBU memory buffer at %{BUFFER_ADDRESS} with size %{SIZE} and data %{DATA}. Fields:

BUFFER_ADDRESS unsigned int

Virtual Address of start of buffer write.

DATA unsigned int

Data written to the buffer.

SIZE unsigned int

Size of sample written to buffer, in bytes.

UNALIGNED_LDST_RETIRED

Processor unaligned load/store. Fields:

ACCESS_SIZE unsigned int

Log2 of the access width in bytes used for alignment checking (1=>half-word, 2->word ...).

NUMBER_OF_BEATS unsigned int

Number of accesses (beats) in this burst (Total size in bytes is ACCESS_SIZE * NUMBER_OF_BEATS).

VADDR unsigned int

The virtual address of the access.

VFP_D_REGS

VFP/NEON D 64 bit register write. Fields:

ALIASING enum

Type of register aliasing used.

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a write to the corresponding bit in the register. For example, 0x1 would indicate a write to VALUE[0].

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

VFP_D_REGS_READ

VFP/NEON D 64 bit register read. Fields:

ID unsigned int

The register number.

VALUE unsigned int

The current value of the register.

VFP_Q_REGS

VFP/NEON Q 128 bit register write. Fields:

ALIASING enum

Type of register aliasing used.

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a write to the corresponding byte in the register. For example, 0x1 would indicate a write to VALUE[7:0].

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

VFP_Q_REGS_READ

VFP/NEON Q 128 bit register read. Fields:

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a read to the corresponding byte in the register. For example, 0x1 would indicate a read to VALUE[7:0].

VALUE unsigned int

The current value of the register.

VFP_SYS_REGS

Writes to the VFP/NEON units system registers. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ID enum

Which VFP system register is written.

MASK unsigned int

The mask indicating updated cumulative bits.

OLD_VALUE unsigned int

The register's old value overwritten.

VALUE unsigned int

The new value written to the VFP system register.

VFP_S_REGS

VFP/NEON S 32 bit register write. Fields:

ALIASING enum

Type of register aliasing used.

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a write to the corresponding bit in the register. For example, 0x1 would indicate a write to VALUE[0].

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

VFP_S_REGS_READ

VFP/NEON S 32 bit register read. Fields:

ID unsigned int

The register number.

VALUE unsigned int

The current value of the register.

VIRTUAL_TAG_ALLOCATED

Virtual tag information when allocated. Fields:

TAG_VADDR unsigned int

Virtual tag virtual address.

VADDR unsigned int

Data Virtual address.

VIRTUAL_TAG unsigned int

Virtual tag value.

WAYPOINT

Signals end of basic block execution. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ISET enum

Origin instruction set.

IS_COND bool

Indicates if this is a conditional waypoint.

PC unsigned int

Origin address (or 0 if unavailable).

TAKEN bool

Indicates if this waypoint was taken.

TARGET unsigned int

Destination address.

TARGET_ISET enum

Destination instruction set.

WFE_END

WFE ended. Fields:

INST_COUNT unsigned int

Ticks count when leaving WFE.

WFE_EVENT_REGISTER

WFE event register status: set/clear, reason. Fields:

INST_COUNT unsigned int

Ticks count.

REASON enum

Reason for set/clear. Only REASON==0 (Cleared by WFE) clears the bit, all other reasons set it.

WFE_IGNORED

WFE ignored. Fields:

EVENT bool

This WFE was ignored because the event register was set.

INST_COUNT unsigned int

Ticks count when ignoring WFE.

TRAPPED bool

This WFE was trapped.

WFE_START

WFE entered. Fields:

INST_COUNT unsigned int

Ticks count when entering WFE.

WFI_END

WFI ended. Fields:

INST_COUNT unsigned int

Ticks count when leaving WFI.

WFI_IGNORED

WFI ignored. Fields:

DISABLED bool

This WFI was ignored because WFI is disabled.

INST_COUNT unsigned int

Ticks count when ignoring WFI.

REASON enum

specifies reason why WFI trace was ignored.

TRAPPED bool

This WFI was trapped.

WFI_START

WFI entered. Fields:

INST_COUNT unsigned int

Ticks count when entering WFI.

WFI_WAKEUP

WFI wakeup. Fields:

INST_COUNT unsigned int

Ticks count when WFI wakeup occurred.

REASON enum

Reason for wakeup.

2.11 ARM_C1-Ultra

This section describes the trace sources.

AA64_ASE_SVE_REGS

Modification of AArch64 Advanced SIMD or SVE Vector Registers (via SIMD or Scalar operations), SVE Predicate Registers and ZT0. Fields:

ID **enum**

The register identifier.

MASK **unsigned int**

Bitmask of the register to signal the modified bits in the VALUE field.

SM **bool**

Whether the PE is in Streaming Mode.

VALUE **unsigned int**

Value of AArch64 vector register.

ASYNC_MEMORY_FAULT

Context ID Register write. Fields:

FAULT **unsigned int**

Fault status present in ESR.ISS encoding for Serror in ARCH64 and present in DFSR encoding for external abort format in ARCH32.

PADDR **unsigned int**

Physical Address (or 0 if unavailable).

VADDR **signed int**

Virtual Address (or 0 if unavailable).

ATOMIC_END_ACCESS

Bus trace access for atomics. Fields:

ACCESS_FAIL **bool**

Memory access failed.

ADDR **unsigned int**

The virtual address of the access.

ATTR **unsigned int**

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

COMPARE_VALUE **unsigned int**

Compare value for CAS.

LOAD_VALUE unsigned int

Loaded values.

NSDESC unsigned int

The security state of the access.

OPERAND_VALUE unsigned int

Operation's operand.

OPERATION enum

Operation type.

PADDR unsigned int

The physical address of the access.

PRIV bool

Is this a privileged access?.

ATOMIC_START_ACCESS

Bus trace access for atomics. Fields:

ADDR unsigned int

The virtual address of the access.

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

COMPARE_VALUE unsigned int

Compare value for CAS.

NSDESC unsigned int

The security state of the access.

OPERAND_VALUE unsigned int

Operation's operand.

OPERATION enum

Operation type.

PADDR unsigned int

The physical address of the access.

PRIV bool

Is this a privileged access?.

ArchMsg.Error.exit_code

DISPLAY exit code: %{EXITCODE} component kind: %{KIND} reason: %{REASON}. Fields:

EXITCODE signed int

Exit code to be returned.

KIND string

Component kind that invoked the exit code trace.

REASON string

Reason why the exit code trace was invoked (optional).

ArchMsg.Warning.IgnoredWarmResetRequest

Event fires when software attempts to request warm reset through write to deprecated DBGPRCR.CWRR.

ArchMsg.Warning.RAS_ErrselrUnpredictableState

DISPLAY ERRSELR is in unpredictable state due to value greater than number_of_error_records written to it.

ArchMsg.Warning.RAS_UnpredictableStateWrite

DISPLAY %{REG_NAME} write ignored due to unpredictable ERRSELR state. Fields:

REG_NAME string

Name of register being written.

ArchMsg.Warning.ReportedRevisionRegisterRead

DISPLAY Reported revision and patch do not match that of the cpu. Fields:

ORIGINAL_VALUE unsigned int

The value the register would have based on the modelled revision.

REGISTER string

The name of the register that has ben read.

REPORTED_VALUE unsigned int

The value the register has been overridden to.

ArchMsg.Warning.Unpredictably Indexed PM Event Register

TAGS UNPREDICTABLE DISPLAY Event %{IsDirect:(Selector value|Register)} %{SEL} is >= %{N}. Fields:

IsDirect unsigned int

Direct Access rather than indirect.

N unsigned int

Number of Accessible Registers.

SEL unsigned int

Selector.

ArchMsg.Warning.cache_contents_unknown

A cache has been activated without being invalidated and may contain **UNKNOWN** content. DISPLAY %{SIDE:(D|I)}-Cache was enabled in %{REGIME:(SECURE|NON_SECURE|HYP)} regime but was not invalidated since power-on: cache lines could contain **UNKNOWN** content %{SIDE:(Note: D-side caches must invalidate SECURE lines even if the cache is only used by NON_SECURE code in order to ensure that unexpected dirty lines tagged SECURE are not naturally evicted)}. Fields:

REGIME unsigned int

Current translation regime.

SIDE unsigned int

0 d-side cache or unified cache, 1 i-side.

ArchMsg.Warning.dcimvac_matches_watchpoint

TAGS **UNPREDICTABLE** DISPLAY Watchpoints matching an AArch32 DCIMVA are implementation defined.

ArchMsg.Warning.decode_fieldmismatch

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with different fields %{FIELD1} and %{FIELD2}. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_fracbitsnegative

TAGS **UNPREDICTABLE** DISPLAY VCVT Instruction is unpredictable with a negative number of fraction bits.

ArchMsg.Warning.decode_initblock

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable in an IT block.

ArchMsg.Warning.decode_initblocknotlast

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable in an IT block when not the last.

ArchMsg.Warning.decode_invalid_condition

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable when the condition is not AL.

ArchMsg.Warning.decode_invalidfieldcombination

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with the values in %{FIELD1} and %{FIELD2}. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_invalidvalue

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with the value in %{FIELD}. Fields:

FIELD enum

Field name.

ArchMsg.Warning.decode_mem_hint_unallocated

TAGS **UNPREDICTABLE** DISPLAY There is no memory hint allocated to this bit pattern.

ArchMsg.Warning.decode_registermatch

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with %{FIELD1} and %{FIELD2} as the same register. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_registermismatch

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with %{FIELD1} and %{FIELD2} as different registers. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_registeroutofrange

TAGS **UNPREDICTABLE** DISPLAY Floating-point instruction is unpredictable indexing beyond the end of the register bank.

ArchMsg.Warning.decode_sbzsbo

TAGS **UNPREDICTABLE** DISPLAY Reserved bits in the instruction are not canonical.

ArchMsg.Warning.decode_transactiontoobig

TAGS **UNPREDICTABLE** DISPLAY Attempt to perform a memory transaction of more than 128 bytes.

ArchMsg.Warning.decode_unpred_other

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable.

ArchMsg.Warning.decode_unpreduseofpc

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable when %{FIELD} is PC. Fields:

FIELD enum

Field name.

ArchMsg.Warning.decode_unpreduseofr13

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable when %{FIELD} is R13. Fields:

FIELD enum

Field name.

ArchMsg.Warning.decode_unpreduseofr13inreglist

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with R13 in register-list.

ArchMsg.Warning.decode_useofd16orhigher

TAGS **UNPREDICTABLE** DISPLAY Attempt to use register D16 or higher where it is not allowed.

ArchMsg.Warning.decode_writebackandbaseinlist

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with write back when the base register is in the transfer list.

ArchMsg.Warning.decode_zeroregistersinlist

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable when no registers are to be transferred.

ArchMsg.Warning.misaligned_ptrs

DISPLAY TRBPTR_EL1(%{TRBPTR_EL1}) or TRBTRG_EL1(%{TRBTRG_EL1}) are not aligned as per TRBIDR_EL1.Align(%{TRBIDR_EL1}). Fields:

TRBIDR_EL1 unsigned int

TRBIDR_EL1.Align value.

TRBPTR_EL1 unsigned int

Current write pointer.

TRBTRG_EL1 unsigned int

Trigger pointer.

ArchMsg.Warning.out_of_range_ptrs

DISPLAY TRBPTR_EL1(%{TRBPTR_EL1}) is out of range as per TRBBASER_EL1(%{TRBBASER_EL1}) or TRBLIMITR_EL1.LIMIT(%{TRBLIMITR_EL1}). Fields:

TRBBASER_EL1 unsigned int

Base pointer.

TRBLIMITR_EL1 unsigned int

Limit pointer.

TRBPTR_EL1 unsigned int

Current write pointer.

ArchMsg.Warning.recursive_branch

An instruction is performing a branch that targets the same instruction. If this was intended then a WFI might be more effective DISPLAY recursive branch detected at PC=%{PC}.

PRIMARY KEY PC. Fields:

PC unsigned int

Address of instruction causing the branch.

ArchMsg.Warning.recursive_exception

An instruction has generated an exception that targets the same instruction. DISPLAY recursive %{TYPE} exception detected at PC=%{PC} (CPSR=%{CPSR}). PRIMARY KEY PC. Fields:

CPSR unsigned int

Processor state.

PC unsigned int

Address of instruction causing the exception.

TYPE enum

Type of exception.

ArchMsg.Warning.reentrant_vector_catch

TAGS **UNPREDICTABLE** DISPLAY Vector catch debug event on Prefetch or Data abort vector %{ADDR}. Fields:

ADDR unsigned int

vector address.

ArchMsg.Warning.reserved_it_state

TAGS **UNPREDICTABLE** DISPLAY Execution with reserved IT state %{ITSTATE}. Fields:

ITSTATE unsigned int

reserved IT state.

ArchMsg.Warning.tlb_contents_unknown

Translation has been enabled without invalidating the TLB. DISPLAY Translation was enabled in the %{PAS:(Secure|Non-secure|Root|Realm)} EL%{EL:d} translation regime (vmid=%{VMID:x}) without invalidating %{INVALIDITY:(|d-side |i-side |)}entries since power-on: %{INVALIDITY:(|D|I|)}TLB could contain **UNKNOWN** content. Fields:

EL unsigned int

Exception level of translation regime.

INVALIDITY unsigned int

Which TLB may be invalid (if TLBs are separate).

NON_SECURE bool

Whether the translation regime is non-secure.

PAS unsigned int

Physical address space of translation regime.

VMID unsigned int

Current VMID, if applicable.

ArchMsg.Warning.unknown_DLR_DSPSR

Exiting debug state with unknown DLR or DSPSR.

ArchMsg.Warning.unknown_ELR_SPSR

Returning from debug exception with unknown ELR or SPSR.

ArchMsg.Warning.unpred_FAR_on_LRCPC3_non_atomic_mem_access

TAGS **UNPREDICTABLE** Unpredictable FAR/FSTATUS on LRCPC3 pair instructions resulting in two separate memory accessesDISPLAY Unpredictable FAR on LRCPC3 non atomic memory access at %{ADDR1} with FSTATUS %{FSTATUS1} or %{ADDR2} with FSTATUS %{FSTATUS2}. Fields:

ADDR1 unsigned int

Possible FAR address 1.

ADDR2 unsigned int

Possible FAR address 2.

FSTATUS1 unsigned int

Possible fault status value 1.

FSTATUS2 unsigned int

Possible fault status value 2.

ArchMsg.Warning.unpred_async_tag_check_fail_on_sve_speculative_load_do_update_ffr

Unpredictable set to FALSE of the FFR element associated to SVE speculative load with an async tag check fail TAGS **UNPREDICTABLE**.

ArchMsg.Warning.unpredictable_a32_breakpoint

TAGS **UNPREDICTABLE** DISPLAY Unpredictable breakpoint on a A32 instruction at %{ADDR1} Offset %{OFFSET} BAS %{BAS} for a %{IS_ADDRESS_MISMATCH_BREAKPOINT:(Mismatch|Match)} Breakpoint number %{N:d} Breakpoint address %{ADDR2}. Fields:

ADDR1 unsigned int

Address.

ADDR2 unsigned int

Breakpoint address.

BAS unsigned int

BAS field.

IS_ADDRESS_MISMATCH_BREAKPOINT bool

Is a Mismatch breakpoint, not a Match breakpoint.

N unsigned int

Breakpoint number.

OFFSET unsigned int

Always has a value of zero.

ArchMsg.Warning.unpredictable_banked_register_access

TAGS **UNPREDICTABLE** DISPLAY Access to banked register (R=%{R:d} SYSm=%{SYSM:2x}) is unpredictable. Fields:

R unsigned int

R bit.

SYSM unsigned int

SYSm.

ArchMsg.Warning.unpredictable_ccfail_undef

TAGS **UNPREDICTABLE** DISPLAY An instruction %{OPCODE} at VA=%{VADDR} failed its condition codes check but would otherwise have caused a non-data-dependent trap or undefined exception. Behaviour is implementation-defined choice of exception or nop. Fields:

OPCODE unsigned int

opcode of the instruction.

VADDR unsigned int

Virtual address of the instruction being executed.

ArchMsg.Warning.unpredictable_ici_bits

TAGS **UNPREDICTABLE** DISPLAY ICI bits do not apply or are architecturally unpredictable for instruction.

ArchMsg.Warning.unpredictable_t32_breakpoint

TAGS **UNPREDICTABLE** DISPLAY Unpredictable breakpoint on a T32 instruction at %{ADDR1} Offset %{OFFSET} BAS %{BAS} for a %{IS_ADDRESS_MISMATCH_BREAKPOINT:(Mismatch|Match)} Breakpoint number %{N:d} Breakpoint address %{ADDR2}. Fields:

ADDR1 unsigned int

Address.

ADDR2 unsigned int

Breakpoint address.

BAS unsigned int

BAS field.

IS_ADDRESS_MISMATCH_BREAKPOINT bool

Is a Mismatch breakpoint, not a Match breakpoint.

N unsigned int

Breakpoint number.

OFFSET unsigned int

Indicates which halfword of a 32-bit Thumb instruction the breakpoint is set on. 0 for the first halfword or if the instruction is 16 bits wide, or 1 for the second halfword.

ArchMsg.Warning.unpredictable_t32_vector_catch

TAGS **UNPREDICTABLE** DISPLAY Unpredictable Vector catch exception on a T32 instruction at %{ADDR} Offset %{OFFSET}. Fields:

ADDR unsigned int

Address.

OFFSET unsigned int

Indicates which halfword of a 32-bit Thumb instruction the vector catch exception is set on. 0 for the first halfword or if the instruction is 16 bits wide, or 1 for the second halfword.

ArchMsg.Warning.unpredictable_watchpoint_far

TAGS **UNPREDICTABLE** DISPLAY The value for FAR/EDWAR for hitting this watchpoint may be anywhere in the range %{LOWER_BOUND} to %{UPPER_BOUND}. Fields:

LOWER_BOUND unsigned int

the lowest address accessed by the instruction that triggered the watchpoint.

UPPER_BOUND unsigned int

the highest watchpointed address accessed by that instruction.

ArchMsg.Warning.unpredictable_watchpoint_far_isscmp

DISPLAY The value for FAR/EDWAR for hitting this watchpoint may be anywhere in the range %{LOWER_BOUND} to %{UPPER_BOUND}. Fields:

LOWER_BOUND unsigned int

the lowest address accessed by the instruction that triggered the watchpoint.

UPPER_BOUND unsigned int

the highest watchpointed address accessed by that instruction.

ArchMsg.Warning.warning_AdvSIMDExpandImmUnexpectedZero

TAGS **UNPREDICTABLE** DISPLAY AdvSIMDExpandImm may treat this immediate value as **UNPREDICTABLE**.

ArchMsg.Warning.warning_ConditionalSMC

TAGS **UNPREDICTABLE** DISPLAY SMC instruction has **UNPREDICTABLE** effects when conditional and when combined with traps.

ArchMsg.Warning.warning_TLBID_out_of_range

TLBID out of range in TLBI according to the configuration of NIS, NVIS, NOS and NVOS. Fields:

NIS unsigned int

NIS.

NOS unsigned int

NOS.

NVIS unsigned int

NVIS.

NVOS unsigned int

NVOS.

TLBID unsigned int

The TLBID value used in the TLBI instruction.

ArchMsg.Warning.warning_TTL_or_TG_mismatches_current_translation_granule

TTL or TG mismatches current translation granule in TLBI VA/RVA/IPA/RIPA. Fields:

CURRENT_TG unsigned int

Current TG as indicated by TCR_ELx or VTCR_EL2.

TLBI_TG unsigned int

TG (or TTL[3:2]) used in TLBI instruction.

ArchMsg.Warning.warning_access_crosses_page_boundary

TAGS **UNPREDICTABLE** DISPLAY Access crosses page boundary.

ArchMsg.Warning.warning_access_crosses_page_boundary_has_fault

TAGS **UNPREDICTABLE** DISPLAY Access that crosses page boundary has fault(s).

ArchMsg.Warning.warning_access_crosses_page_boundary_has_fault_on_both_pages

TAGS **UNPREDICTABLE** DISPLAY Access that crosses page boundary has a fault on each page.

Fields:

ADDR1 unsigned int

FAR address on first page.

ADDR2 unsigned int

FAR address on second page.

FSTATUS1 unsigned int

Fault status on first page.

FSTATUS2 unsigned int

Fault status on second page.

ArchMsg.Warning.warning_access_crosses_page_boundary_spanning_different_memory

TAGS **UNPREDICTABLE** DISPLAY Access at address %{ADDR} crosses page from MEM_TYPE %{MEMTYPE_PAGE1} to %{MEMTYPE_PAGE2}. Fields:

ADDR unsigned int

address of access crossing page.

MEMTYPE_PAGE1 signed int

First Page's memory type (0-Normal, 1-Non-Normal).

MEMTYPE_PAGE2 signed int

Second Page's memory type (0-Normal, 1-Non-Normal).

ArchMsg.Warning.warning_access_wraps_around_memory

TAGS **UNPREDICTABLE** DISPLAY Access wraps around memory in %{WIDTH:d}bit mode.

Fields:

WIDTH unsigned int

Address width.

ArchMsg.Warning.warning_bcr_linking_status

TAGS **UNPREDICTABLE** DISPLAY DBGBCR.{BT2,BT} combination gives unpredictable behaviour.

ArchMsg.Warning.warning_bcr_mask_reserved

TAGS **UNPREDICTABLE** DISPLAY DBGBCR_ELn.MASK is set to a reserved value.

ArchMsg.Warning.warning_bvr_masked_bit_not_zero

TAGS **UNPREDICTABLE** DISPLAY DBGBVRn_EL1.VA contains a bit masked by DBGBCRn_EL1.MASK that is not zero.

ArchMsg.Warning.warning_ccsidr_unimplemented_level

TAGS **UNPREDICTABLE** DISPLAY CCSIDR read while CSSELR %{CSSELR:x} points to an unimplemented cache level. Fields:

CSSELR unsigned int

current effective value of CSSELR.

ArchMsg.Warning.warning_change_to_ns_when_tge_set

TAGS **UNPREDICTABLE** DISPLAY Attempting to change NS to 1 when HCR.TGE = 1.

ArchMsg.Warning.warning_contiguous_bit_check_abort

TAGS **UNPREDICTABLE** DISPLAY An abort occurred whilst attempting to check TLB entry is contiguous at %{ENTRY_ADDR}. Fields:

ENTRY_ADDR unsigned int

address of conflicting TLB entry.

ArchMsg.Warning.warning_contiguous_bit_error

TAGS **UNPREDICTABLE** DISPLAY TLB entry at %{ENTRY_ADDR} was not contiguous with entry at %{CONTIG_ADDR} Contents are %{ENTRY_DATA} but expected %{CONTIG_DATA}. Fields:

CONTIG_ADDR unsigned int

address of the TLB first read, with which this entry is expected to be contiguous.

CONTIG_DATA unsigned int

expected contents based on the entry at CONTIG_ADDR.

ENTRY_ADDR unsigned int

address of conflicting TLB entry.

ENTRY_DATA unsigned int

contents of conflicting TLB entry.

ArchMsg.Warning.warning_contiguous_bit_error_D128

TAGS **UNPREDICTABLE** DISPLAY TLB entry at %{ENTRY_ADDR} was not contiguous with entry at %{CONTIG_ADDR} Contents are %{ENTRY_DATA_HI} %{ENTRY_DATA_LO} but expected %{CONTIG_DATA}. Fields:

CONTIG_ADDR unsigned int

address of the TLB first read, with which this entry is expected to be contiguous.

CONTIG_DATA unsigned int

expected contents based on the entry at CONTIG_ADDR.

ENTRY_ADDR unsigned int

address of conflicting TLB entry.

ENTRY_DATA_HI unsigned int

Higher bit contents of conflicting TLB entry.

ENTRY_DATA_LO unsigned int

Lower bit contents of conflicting TLB entry.

ArchMsg.Warning.warning_cp10_cp11_mismatch

TAGS **UNPREDICTABLE** DISPLAY Unpredictable configuration of CP10 and CP11 controls at EL %{EL:d}. Fields:

EL unsigned int

EL owning the control.

ArchMsg.Warning.warning_cp10_cp11_reserved_value

TAGS **UNPREDICTABLE** DISPLAY Unpredictable configuration of CPACR CP10 ({CP10}) or CP11 ({CP11}). Fields:

CP10 unsigned int

CP10 access.

CP11 unsigned int

CP11 access.

ArchMsg.Warning.warning_csselr_level_out_of_range

TAGS **UNPREDICTABLE** DISPLAY CSSELR.Level out of range, written value {LEVEL} with only {IMPLEMENTED:d} levels implemented. Fields:

IMPLEMENTED unsigned int

number of cache levels implemented.

LEVEL unsigned int

written value of CSSELR.Level.

TYPE enum

The type of cache selected by CSSELR.InD and TnD.

ArchMsg.Warning.warning_debug_flow_control_bits_not_obeyed

TAGS **UNPREDICTABLE** DISPLAY Flow control bits for {REGISTER} were not obeyed, causing {ERROR}. Fields:

ERROR enum

type of error resulting.

REGISTER enum

register under consideration.

ArchMsg.Warning.warning_debug_register_access_during_reset

TAGS **UNPREDICTABLE** DISPLAY Unpredictable {IS_WRITE:(read|write)} access to debug register offset {OFFSET} during reset. Fields:

IS_WRITE unsigned int

Write Not Read.

OFFSET unsigned int

Register Offset.

ArchMsg.Warning.warning_decode_cps_inconsistent_fields

TAGS **UNPREDICTABLE** DISPLAY The mode setting fields in the CPS instruction are inconsistent.

ArchMsg.Warning.warning_decode_invalid_state

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable in current exception-level/security state.

ArchMsg.Warning.warning_default_cacheable_mmu_on

TAGS **UNPREDICTABLE** DISPLAY Default cacheable is enabled (HCR.DC = 1) while MMU is enabled.

ArchMsg.Warning.warning_default_cacheable_vmmu_off

TAGS **UNPREDICTABLE** DISPLAY Default cacheable is enabled (HCR.DC = 1) while HCR.VM = 0.

ArchMsg.Warning.warning_deprecated_wvr_bit_2_set

TAGS **UNPREDICTABLE** DISPLAY DBGWVR%{N:d}_EL1=%{WVR:h} has bit 2 set, deprecated in ARMv8. Fields:

N unsigned int

index.

WVR unsigned int

DBGWVR.

ArchMsg.Warning.warning_eret_while_software_step_active_pending

Missing ISB between setting MDSCR_EL1.SS and ERET TAGS **UNPREDICTABLE**.

ArchMsg.Warning.warning_exclusive_to_non_normal

TAGS **UNPREDICTABLE** DISPLAY Exclusive Access to Non-Normal Memory %{ADDR}. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_exclusive_to_non_writeback

TAGS **UNPREDICTABLE** DISPLAY Exclusive Access to Non-Writeback-Cacheable Memory %{ADDR}. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_execute_from_device_memory

TAGS **UNPREDICTABLE** DISPLAY An attempt was made to execute from Device memory.

ArchMsg.Warning.warning_illegal_cpsr_mode

TAGS **UNPREDICTABLE** DISPLAY Writing an illegal or unimplement mode (%{NEW_MODE:h}) to CPSR was unpredictable. Fields:

NEW_MODE unsigned int

New value of CPSR.M.

ArchMsg.Warning.warning_illegal_srs_mode

TAGS **UNPREDICTABLE** DISPLAY Illegal or **UNPREDICTABLE** mode (%{MODE:h}) used for SRS instruction. Fields:

MODE unsigned int

mode for Banked SP.

ArchMsg.Warning.warning_implementation_defined_read_debug_register_in_SCS

Read Access to Debug registers from the processor is IMPLEMENTATION DEFINED TAGS IMP_DEF. Fields:

PPB_OFFSET unsigned int
debug register offset.

ArchMsg.Warning.warning_implementation_defined_sequential_security_transitions_supported

The behavior of sequential instruction fetches that cross from non-secure to secure memory and contain an SG instruction is CONSTRAINED UNPREDICTABLE TAGS UNPREDICTABLE.

ArchMsg.Warning.warning_implementation_defined_stack_limit_check_supported

It is IMPLEMENTATION DEFINED whether stack pointer limit checking is performed for this instruction TAGS IMP_DEF. Fields:

ADDRESS unsigned int
Address of instruction.

OPCODE unsigned int
instruction opcode.

ArchMsg.Warning.warning_implementation_defined_taggednonsharedmemorytype

TAGS IMP_DEF DISPLAY Memory attribute has Tagged, non shared memory type; it is IMPLEMENTATION DEFINED whether the memory location is treated as Tagged or non-Tagged.

ArchMsg.Warning.warning_implementation_defined_write_debug_register_in_SCS

Write Access to Debug registers from the processor is IMPLEMENTATION DEFINED TAGS IMP_DEF. Fields:

DATA unsigned int
data attempted to be written.

PPB_OFFSET unsigned int
debug register offset.

ArchMsg.Warning.warning_invalid_tcr_granule

TAGS UNPREDICTABLE DISPLAY TCR.TG%{TG_ID:(0|1)} bits have been set to a granule size %{{REQUEST:d}}K not implemented - using %{{SUBSTITUTE:d}}K. Fields:

REQUEST signed int
page size requested (or 0 for reserved).

SUBSTITUTE signed int
best guess available page size.

TG_ID bool
bits TG0 or TG1.

ArchMsg.Warning.warning_load_multiple_user_registers_from_user_mode

TAGS UNPREDICTABLE DISPLAY An LDM(user registers) instruction executed from user mode.

ArchMsg.Warning.warning_mte_in_mem_attr_not_supported

TAGS UNPREDICTABLE DISPLAY Memory attribute %{ATTR} has Tagged memory type but Armv8.5-MTE is not implemented. Fields:

ATTR unsigned int

Memory attribute.

ArchMsg.Warning.warning_reserved_breakpoint_state_match

TAGS UNPREDICTABLE DISPLAY %{IS_BREAKPOINT:(Watchpoint|Breakpoint)} programmed with a reserved combination of HMC, SSC and %{IS_BREAKPOINT:(PAC|PMC)}. Fields:

IS_BREAKPOINT unsigned int

Is a breakpoint, not a watchpoint.

ArchMsg.Warning.warning_shareability

TAGS UNPREDICTABLE DISPLAY Unpredictable: combination of the 1st and 2nd stages of translation is Normal Inner Non-Cacheable, Outer Non-Cacheable.

ArchMsg.Warning.warning_software_step_set_while_enabled

TAGS UNPREDICTABLE DISPLAY MDSCR_EL1.SS set to 1 while software step debug exceptions are enabled.

ArchMsg.Warning.warning_thumb_instruction_wraps_around_memory

TAGS UNPREDICTABLE DISPLAY Thumb instruction wraps around memory.

ArchMsg.Warning.warning_ttbr_sbz_bits_are_not_zero

TAGS UNPREDICTABLE DISPLAY %{TTBR}[%{MSB:d}:0] should be zero, but are not. Fields:

MSB unsigned int

x-1.

TTBR enum

which TTBR register.

ArchMsg.Warning.warning_unaligned_address_dbgdtrrx_write

TAGS UNPREDICTABLE DISPLAY Unaligned address %{ADDR} in X0/R0 when DBGDTRRX written in memory access mode. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unaligned_address_dbgdtrtx_read

TAGS UNPREDICTABLE DISPLAY Unaligned address %{ADDR} in X0/R0 when DBGDTRTX read in memory access mode. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unaligned_to_device

TAGS UNPREDICTABLE DISPLAY Unaligned Access to Device Memory %{ADDR}. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unaligned_to_strongly_ordered

TAGS **UNPREDICTABLE** DISPLAY Unaligned Access to Strongly Ordered Memory %{ADDR}.

Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unknown_sau_rnr

TAGS **UNPREDICTABLE** DISPLAY SAU_RNR was set to an unsupported value.

ArchMsg.Warning.warning_unpredictable_AIRCR_PRIS_and_BFHFNMINs

TAGS **UNPREDICTABLE** DISPLAY The effect of setting both AIRCR.BFHFNMINs and AIRCR.PRIS to 1 is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_AIRCR_VECTCLRACTIVE_when_not_in_debug

TAGS **UNPREDICTABLE** DISPLAY The effect of writing a 1 to AIRCR.VECTCLRACTIVE if the processor is not halted in Debug state is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_AIRCR_VECTRESET_and_SYSRESETREQ

TAGS **UNPREDICTABLE** DISPLAY When the processor is halted in Debug state, if a write to the register writes a 1 to both VECTRESET and SYSRESETREQ, the behavior is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_AIRCR_VECTRESET_when_not_in_debug

TAGS **UNPREDICTABLE** DISPLAY The effect of writing a 1 to AIRCR.VECTRESET if the processor is not halted in Debug state is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_AIRCR_incorrect_VKEY

TAGS **UNPREDICTABLE** DISPLAY The value 0x05FA must be written to AIRCR.VKEY, otherwise the register write is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_EXC_RETURN_Reserved_Bit

EXC_RETURN[23:7] are reserved with the special condition that all bits should be written as one. Values other than all 1s are **UNPREDICTABLE**. TAGS **UNPREDICTABLE**. Fields:

EXC_RETURN unsigned int

Special PC value.

SBO_MASK unsigned int

Which bits Should Be One to not UNPRED.

ArchMsg.Warning.warning_unpredictable_PMEVTYPEPERn_ELO_VS_reserved_value

TAGS **UNPREDICTABLE** DISPLAY Unpredictable reserved value of PMEVTYPER<n>_ELO.VS.

ArchMsg.Warning.warning_unpredictable_change_to_DEMCR_MON_STEP_at_insufficient_priority

TAGS **UNPREDICTABLE** DISPLAY The effect of changing DEMCR.MON_STEP at an execution priority that is lower than the priority of the DebugMonitor exception is **CONSTRAINED UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_change_to_priority_of_active_exception

TAGS UNPREDICTABLE DISPLAY Changing the priority of an active exception is UNPREDICTABLE.

ArchMsg.Warning.warning_unpredictable_clear_lspact

TAGS UNPREDICTABLE DISPLAY UNPREDICTABLE state on context switch. Fields:

DRVEXC_TAKEN unsigned int

Derived exception taken, not pended.

REACHED_LAST_STORE unsigned int

Only the last faulted.

ArchMsg.Warning.warning_unpredictable_dbg_exit_unaligned_dlr

TAGS UNPREDICTABLE DISPLAY Debug exit to AArch32 with DLR[0] set to 1. PC[0] can be set to 0 or to DLR[0].

ArchMsg.Warning.warning_unpredictable_exception_catch

TAGS UNPREDICTABLE DISPLAY Generation of an exception catch event is unpredictable.

ArchMsg.Warning.warning_unpredictable_exception_return_inconsistent_state

TAGS UNPREDICTABLE DISPLAY state on exception return is unpredictable.

ArchMsg.Warning.warning_unpredictable_exception_return_instruction

TAGS UNPREDICTABLE DISPLAY exception return instruction is unpredictable.

ArchMsg.Warning.warning_unpredictable_in_debug_state

TAGS UNPREDICTABLE DISPLAY Instruction is UNPREDICTABLE when executed in debug state.

ArchMsg.Warning.warning_unpredictable_mair_encoding

TAGS UNPREDICTABLE DISPLAY Memory attribute %{ATTR} has invalid MAIR encoding; this may lead to unpredictable results. Fields:

ATTR unsigned int

Memory attribute.

ArchMsg.Warning.warning_unpredictable_pmu_counter_access

TAGS UNPREDICTABLE DISPLAY Access to PMU counters from non-secure EL0 or EL1 is UNPREDICTABLE with MDCR_EL2.HPMN set to 0.

ArchMsg.Warning.warning_unpredictable_prioritization_breakpoint_match_vector_catch

TAGS UNPREDICTABLE DISPLAY Two events with the same priority occurred at the same instruction: Address Matching Vector Catch debug event, and Breakpoint debug event. It is constrained unpredictable which is taken.

ArchMsg.Warning.warning_unpredictable_stack_selection

TAGS UNPREDICTABLE DISPLAY UNPREDICTABLE state on context switch.

ArchMsg.Warning.warning_unpredictable_tsize_out_of_range

TAGS UNPREDICTABLE DISPLAY %{TCR}.T%{TTBR:d}SZ (%{TSIZE:d} bits) is out of range. Must be between 25 and %{TSIZE_MAX:d}. Fields:

TCR enum

Which TCR.

TSize signed int

Value of TSize.

TSize_MAX signed int

The maximum allowed value.

TTBR unsigned int

Which TTBR region.

ArchMsg.Warning.warning_unpredictable_unaligned_pc_as_base_register

TAGS **UNPREDICTABLE** DISPLAY The PC value must be word-aligned, otherwise the behavior of the instruction is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_unaligned_pop_stack

TAGS **UNPREDICTABLE** DISPLAY **CONSTRAINED UNPREDICTABLE** unaligned stack frame `%{FRAME_ADDR:x}` in exception-return. Fields:

FRAME_ADDR unsigned int

unaligned stack frame address.

ArchMsg.Warning.warning_unpredictable_unaligned_to_non_writeback

TAGS **UNPREDICTABLE** DISPLAY Unaligned Access to Non-Writeback-Cacheable Memory.

ArchMsg.Warning.warning_unpredictable_vmsa_memattrib

TAGS **UNPREDICTABLE** DISPLAY Unpredictable vmsa memory attribute with texcb value programmed to `%{TEXCB}` when tex remap is `%{TEX_REMAP}`. Fields:

TEXCB unsigned int

texcb value.

TEX_REMAP bool

use tex remap.

ArchMsg.Warning.warning_unpredictable_vtcr_t0sz_sl0

TAGS **UNPREDICTABLE** DISPLAY The combination of VTCR.TOSZ=`%{TOSZ:d}` and VTCR.SL0=`%{SL0:d}` is unpredictable. Fields:

SL0 signed int

VTCR.SL0, starting level for stage 2 translation table walks.

TOSZ signed int

VTCR.TOSZ, required input address range.

ArchMsg.Warning.warning_unpredictable_write_DHCSR

TAGS **UNPREDICTABLE** DISPLAY **UNPREDICTABLE** set of bit-changes in DHCSR. Fields:

NEW unsigned int

data attempted to be written.

OLD unsigned int

previous value.

ArchMsg.Warning.warning_unpredictable_write_SHCSR

TAGS UNPREDICTABLE DISPLAY UNPREDICTABLE setting a pending bit to 1 for an exception with priority \geq execution priority.

ArchMsg.Warning.warning_unsupported_access_to_memory_mapped_register

TAGS UNPREDICTABLE DISPLAY Access to memory mapped register is unsupported.

ArchMsg.Warning.warning_user_jmcr_access

TAGS UNPREDICTABLE DISPLAY User %{WRITE:(write|read)} access to JMCR is unpredictable. Fields:

WRITE unsigned int
access is write.

ArchMsg.Warning.warning_wcr_mask_and_bas

TAGS UNPREDICTABLE DISPLAY DBGWCR_EL1.MASK is non-zero and BAS is not 0xff.

ArchMsg.Warning.warning_wcr_mask_reserved

TAGS UNPREDICTABLE DISPLAY DBGWCR_ELn.MASK is set to a reserved value.

ArchMsg.Warning.warning_wcr_non_configuous_bas

TAGS UNPREDICTABLE DISPLAY DBGWCR_EL1.BAS has non-contiguous set of ones.

ArchMsg.Warning.warning_wvr_masked_bit_not_zero

TAGS UNPREDICTABLE DISPLAY DBGWVRn_EL1.VA contains a bit masked by DBGWCRn_EL1.MASK that is not zero.

ArchMsg.Why.why_branch_target_exception

A branch target exception has occurred. DISPLAY branch target exception: target pc=%{PC}, target opcode=%{OPCODE}. PRIMARY KEY PC. Fields:

BTYPE unsigned int
pstate.BTYPE at time branch taken.

OPCODE unsigned int
opcode at branch target.

PC unsigned int
Virtual Address of branch target.

ArchMsg.Why.why_illegal_state

An illegal mode change has occurred. DISPLAY illegal mode change: %{MESSAGE}. Fields:

MESSAGE string
The message.

ArchMsg.warning_invalid_interval_counter_reload_value

Invalid interval counter reload value in PMSIRR_EL1.INTERVAL: software should set this to a value greater than or equal to the minimum indicated by PMSIDR_EL1.Interval.

ArchMsg.warning_invalid_spe_buffer_write_pointer

Invalid value programmed in PMBPTR_EL1.PTR or PMBLIMITR_EL1.LIMIT: Current write pointer must be at least one sample record below the write limit pointer and PMBPTR_EL1.PTR[63:56] must be equal to PMBLIMITR_EL1.LIMIT[63:56].

BRANCH_MISPREDICT

Simulating branch mispredict. Fields:

PC unsigned int

Origin address (or 0 if unavailable).

BRA_DIR

Direct branches, to immediate address. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISET enum

The instruction set of the branch instruction.

IS_COND bool

Indicates if this is a conditional branch.

IS_HINTED bool

Whether the conditional instruction is hinted.

IS_ISB bool

Whether the instruction is ISB.

IS_LINK bool

Indicates if this is a branch with link.

PC unsigned int

The address of the branch instruction.

TARGET_ISET enum

The instruction set after the branch.

TARGET_PC unsigned int

The address the instruction branches to.

BRA_INDIR

Indirect branches, perhaps to a register. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISSET enum

The instruction set of the branch instruction.

IS_COND bool

Indicates if this is a conditional branch.

IS_HINTED bool

Whether the conditional instruction is hinted.

IS_ISB bool

Whether the instruction is ISB.

IS_LINK bool

Indicates if this is a branch with link.

PC unsigned int

The address of the branch instruction.

TARGET_ISET enum

The instruction set after the branch.

TARGET_PC unsigned int

The address the instruction branches to.

BRA_INDIR_CCFAIL

Indirect branches fail, perhaps to a register.

BRA_RET_CCFAIL

Branch return fail, perhaps to a register.

CACHE_MAINTENANCE_OP

Cache Maintenance Operation. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

DATA unsigned int

Specified MVA or Set/way.

FUNCTION enum

Clean / Invalidate.

SCOPE enum

Affected region.

SIDE enum

Inst / Data.

CCFAIL

Conditional instruction condition check fail. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CCFAIL_UNC

Unconditional instruction condition check fail. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CCPASS

Conditional instruction condition check pass. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CCPASS_UNC

Unconditional instruction condition check pass. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CODE_CACHE_MAINT

Code cache maintenance. Fields:

END_INCL unsigned int

Inclusive end address for VA or PA requests.

NS bool

Non-secure world for PA.

START unsigned int

Start address for VA or PA requests.

TYPE enum

Request type.

COMPILE_BLOCK_END

Last instruction of basic block translated. Fields:

VADDR unsigned int

Address of next instruction after this basic block.

COMPILE_BLOCK_START

First instruction of basic block translated. Fields:

VADDR unsigned int

Address of first instruction in basic block.

COMPILE_INST

ARM instruction compiled. Fields:

DISASS string

Disassembly of the instruction.

ISSET enum

The instruction set of this instruction.

ITSTATE unsigned int

The ITSTATE current for the instruction.

OPCODE unsigned int

The opcode of the instruction.

PC unsigned int

The address of the instruction.

SIZE unsigned int

The size of the instruction in bytes.

CONTEXTIDR

Context ID Register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

NS bool

Secure or nonsecure banked register is accessed.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The new value written.

CONTEXT_SYNC

Called for every context synchronization event. Fields:

ARCH_CSE bool

Architectural Context Synchronization Event.

PC unsigned int

The address of the synchronization instruction.

SYSREG_SYNC bool

System register synchronization event.

CORE_ENDIAN

Core BE8 Big-Endian state changed. Fields:

BE8 bool

Core BE8 Big-Endian state.

CORE_INFO

Static processor attributes. Only triggered by a call to DumpState(). Fields:

ARCH_PROFILE enum

The architecture profile of the core.

CLUSTER_ID unsigned int

The cluster ID of this processor.

CORE_NUM unsigned int

The number of this core in an MP processor.

DETAILED_CORE_FEATURES enum

Detailed core features as a bitfield. Enum values are bitmasks to extract the info from the field.

FPU_VERSION enum

The VFP version implemented by the core.

MEM_ARCH enum

The memory architecture of the core.

MIDR_EL1 unsigned int

MIDR_EL1 value.

NUM_CORES unsigned int

The number of cores in this MP processor.

QUANTUM_SIZE unsigned int

The default quantum size of the core (in ticks).

SECURITY_FEATURES bool

Does the core have security features?.

CORE_LOADS

Processor load accesses. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ACQREL enum

Is this an acquire/release.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DATA unsigned int

The data read or written.

ELEMENT_SIZE unsigned int

Width of each element.

ENDIANNESS enum

Byte ordering of this access.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

LOCK enum

Normal, exclusive or locked access.

MECID unsigned int

Memory Encryption Context ID, if applicable.

MECID2 unsigned int

Memory Encryption Context ID of the second page, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PADDR_TAG unsigned int

If tag checked, the physical address tag for the access.

PAGE_SIZE unsigned int

Page size as $\log_2(\text{size in bytes})$.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK, 2=Exclusive Failed, 3=Unsupported.

SIZE unsigned int

Width of the access in bytes. Only required if DATA is not traced.

TAG_CHECKED bool

Is this access tag checked.

TAG_CHECKED2 bool

Is the second page access tag checked.

TRANS bool

Is this a translated access.

VADDR unsigned int

The virtual address of the access.

CORE_REGS

Changes of the core registers R0 to R14. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number, 0 to 14.

MODE enum

Bank of the register accessed.

OLD_VALUE unsigned int

The old value overwritten.

PHYS_ID enum

The physical register accessed.

VALUE **unsigned int**

The new value written to the register.

CORE_REGS64

Changes of the core registers X0..X30, SP_ELn. Fields:

CORE_NUM **unsigned int**

Core number in a multi processor.

ID **enum**

The register number, 0-30 for X0-X30, >=32 for SP_ELn.

OLD_VALUE **unsigned int**

The old value overwritten.

VALUE **unsigned int**

The new value written to the register.

CORE_REGS64_READ

General purpose AArch64 register has been read X0..X30, SP_ELn. Fields:

ID **enum**

The register number, 0-30 for X0-X30, >=32 for SP_ELn.

VALUE **unsigned int**

The value read from the register.

CORE_STALLED_TRANSITION

Detect when a core is prevented from progressing PChannel states. Fields:

CURRENT_STATE **enum**

Current power mode of core.

REASON **string**

The reason a transition was stalled, and what must be done to unblock it.

TARGET_STATE **enum**

Desired power mode of core after transition.

CORE_STORES

Processor store accesses. Fields:

ACCESS_TYPE **enum**

The type of instruction performing the access.

ACQREL **enum**

Is this an acquire/release.

ATTR **unsigned int**

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DATA unsigned int

The data read or written.

ELEMENT_SIZE unsigned int

Width of each element.

ENDIANNESS enum

Byte ordering of this access.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

LOCK enum

Normal, exclusive or locked access.

MECID unsigned int

Memory Encryption Context ID, if applicable.

MECID2 unsigned int

Memory Encryption Context ID of the second page, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PADDR_TAG unsigned int

If tag checked, the physical address tag for the access.

PAGE_SIZE unsigned int

Page size as $\log_2(\text{size in bytes})$.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK, 2=Exclusive Failed, 3=Unsupported.

SIZE unsigned int

Width of the access in bytes. Only required if DATA is not traced.

TAG_CHECKED bool

Is this access tag checked.

TAG_CHECKED2 bool

Is the second page access tag checked.

TRANS bool

Is this a translated access.

VADDR unsigned int

The virtual address of the access.

CP14_READ

System Coprocessor register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP14 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc1 unsigned int
opcode 1.

opc2 unsigned int
opcode 2.

CP14_WRITE

System Coprocessor register write. Fields:

CORE_NUM unsigned int
Core number in a multi processor.

CRm unsigned int
CRm.

CRn unsigned int
CRn.

NS bool
Secure or nonsecure banked register is accessed.

REG_NAME string
Name of the CP14 register accessed.

SECURITY_STATE enum
The current security state.

UNDEF bool
The register accessed is undefined.

UPDATED_VALUE unsigned int
Updated value of the register now it has been written.

VALUE unsigned int
The new value written.

opc1 unsigned int
opcode 1.

opc2 unsigned int
opcode 2.

CP15_READ

System Coprocessor register read. Fields:

CORE_NUM unsigned int
Core number in a multi processor.

CRm unsigned int
CRm.

CRn unsigned int
CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP15 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc1 unsigned int

opcode 1.

opc2 unsigned int

opcode 2.

CP15_READ64

System Coprocessor register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP15 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc unsigned int

opcode 1.

CP15_WRITE

System Control Coprocessor register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP15 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc1 unsigned int

opcode 1.

opc2 unsigned int

opcode 2.

CP15_WRITE64

System Coprocessor register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP15 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc unsigned int

opcode 1.

CPSR

CPSR change. (In AArch64 PSTATE represented in SPSR format). Fields:

CORE_NUM unsigned int

Core number in a multi processor.

OLD_VALUE unsigned int

The old CPSR value.

UNKNOWN unsigned int

Bits within the register that have unknown value.

VALUE unsigned int

The new CPSR value.

CRYPTO_SPEC

Every crypto instruction speculatively executed.

CorePowerStateChange

Core power-up state has changed. DISPLAY Core powered %{PU:(up|down)}. Fields:

PU bool

Whether the core powered up.

DATA_CACHE_SET_ALLOC_TAGS

Set allocation tags from DC GVA instruction. Fields:

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

NSDESC unsigned int

The security state of the access.

PADDR unsigned int

The physical address of the access.

PAS enum

The physical address space of the page.

RESPONSE enum

0=Aborted, 1=OK.

SIZE unsigned int

The size of memory in bytes.

TAG unsigned int

The value of the tag being set.

TAGGING_TYPE enum

Type of Memory tagging (Physical or Virtual).

TAG_VADDR unsigned int

The virtual address of the Virtual Tag if Virtual tagging is used; otherwise 0.

VADDR unsigned int

Virtual address used for setting allocation tags.

DATA_CACHE_ZERO

Zero and invalidate cache line from DC ZVA instruction. Fields:

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

NSDESC unsigned int

The security state of the access.

PADDR unsigned int

The physical address of the access.

PADDR_TAG unsigned int

If tag checked, the physical address tag for the access.

PAS enum

The physical address space of the page.

RESPONSE enum

0=Aborted, 1=OK.

SIZE unsigned int

The size of zero'd memory in bytes.

TAG_CHECKED bool

Is this access tag checked.

VADDR unsigned int

Virtual address of the zero'd cache line.

DEBUG_EVENT

Hardware debug support event. Fields:

EVENT enum

Description of event.

VALUE unsigned int
data value.

DEBUG_TARGET_ISET

Hardware debug target ISA. Fields:

TARGET_ISET enum
The target instruction set of the debug state.

DMI_ALLOCATE

DMI allocated. Fields:

ACCESS_DIRECTION enum
READ / WRITE.

PADDR unsigned int
The physical address of the DMI.

PTR unsigned int
DMI range pointer.

SIDE enum
DSIDE / ISIDE.

SIZE unsigned int
The size of DMI.

VADDR unsigned int
The virtual address of the DMI..

DMI_HIT

DMI hit, access used DMI (only fire for small DMI regions i.e. less than a full page). Fields:

ACCESS_DIRECTION enum
READ / WRITE.

PADDR unsigned int
The physical address of the DMI.

SIDE enum
DSIDE / ISIDE.

SIZE unsigned int
The size of the DMI.

VADDR unsigned int
The virtual address of the DMI..

DMI_REVOKE

DMI revoked. Fields:

ACCESS_DIRECTION enum
READ / WRITE.

PADDR unsigned int

The physical address of the DMI.

SIDE enum

DSIDE / ISIDE.

SIZE unsigned int

The size of DMI.

VADDR unsigned int

The virtual address of the DMI..

END_COMPILE

Compilation end. Fields:

END_OF_PAGE_COUNT unsigned int

Number of basic blocks exited due to page boundary since START_COMPILE.

FETCHFAIL_COUNT unsigned int

Number of basic blocks exited due to ifetch failure START_COMPILE.

INST_COUNT unsigned int

Number of instructions compiled since START_COMPILE.

NONSEQ_COUNT unsigned int

Number of basic blocks exited due to non-sequential instructions since START_COMPILE.

PAGE_STRADDLE_COUNT unsigned int

Number of basic blocks exited due to unaligned instructions crossing page since START_COMPILE.

ESB

Error Synchronization Barrier hit. Fields:

ASYNC bool

ESB is async.

IMPLICIT bool

ESB is implicit.

EXCEPTION

Exception taken. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ESR unsigned int

The value assigned to ESR when exception is taken.

ESR_EC unsigned int

The Value assigned to ESR.EC when exception is taken.

IS_PHYSICAL bool

Physical or Virtual exception.

IS_TRAP bool

Exception coming from a trap.

LR unsigned int

The value assigned to the link register.

PC unsigned int

The location where the exception occurred.

PREFERRED_RETURN unsigned int

The preferred return address for the exception.

TAKEN_LOCALLY bool

Whether exception is taken locally(same exception level).

TARGET_ISET enum

The instruction set of the exception handler code.

TARGET_PC unsigned int

The address the exception branches to.

VECTOR enum

The exception vector.

VECTOR_OFFSET unsigned int

The offset of an exception vector from the base address.

EXCEPTION_END

Every exception completed.

EXCEPTION_INFO

Trace human-readable reason of the current exception for data/instruction aborts, floating point exceptions and unknown exceptions. Fields:

DESCRIPTION string

Reason for current exception.

ESR_EC unsigned int

The value of ESR.EC when exception is taken.

EXCEPTION_RAISE

Every exception raised.

EXCEPTION_RETURN

Branches on leaving exception. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISSET enum

The instruction set of the branch instruction.

IS_COND bool

Indicates if this is a conditional branch.

IS_HINTED bool

Whether the conditional instruction is hinted.

IS_ISB bool

Whether the instruction is ISB.

IS_LINK bool

Indicates if this is a branch with link.

PC unsigned int

The address of the branch instruction.

TARGET_ISSET enum

The instruction set after the branch.

TARGET_PC unsigned int

The address the instruction branches to.

EXCEPTION_RETURN_PREBRANCH

Exception return event that must occur before PC and processor state have updated.

EXCEPTION_START

Every exception started.

FIQ_TAKEN

FIQ taken exception.

FP_EXCEPTIONS_SIGNALLED

Trace which FP exceptions were signalled as a part of the FP operations performed by an instruction. Fields:

FP_EXCEPTIONS_SIGNALLED_MASK unsigned int

Indicates which FP exceptions were signalled by an instruction. This uses the same encoding as the cumulative floating-point exception bitfields in FPSR.

FP_STATE

Floating point state. Fields:

FPSR unsigned int

FPSR register value.

MASK unsigned int

The mask indicating updated cumulative bits.

FREQ_CHANGED

Core frequency changed. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The instruction count of this core.

NEW_FREQ signed int

The new frequency of this core (expressed in Hz).

OLD_FREQ signed int

The old frequency of this core (expressed in Hz).

HLT

HLT instruction occurred. Fields:

IMM16 unsigned int

The 16-bit immediate value encoded in HLT instruction.

INST

Every instruction executed. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DEBUG_STATE bool

The instruction is executed in debug state.

DISASS string

Disassembly of instruction.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISSET enum

The current instruction set.

ITSTATE unsigned int

The current ITSTATE.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

MODE enum

The mode the core is in.

NS unsigned int

The core's non-secure bit.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

OPCODE unsigned int

The opcode of the instruction.

PADDR unsigned int

The physical address of the instruction.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the instruction.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the page.

PC unsigned int

The address of the instruction.

SECURITY_STATE enum

The core's security state.

SIZE unsigned int

The size of the instruction in bytes.

INST_END

Every instruction completed.

INST_START

Every instruction started. Fields:

DEBUG_STATE bool

The instruction is executed in debug state.

ISET enum

The current instruction set.

MODE enum

The mode the core is in.

NS enum

The current Secure State.

PC unsigned int

The address of the conditional instruction.

SECURITY_STATE enum

The current Security State.

INST_STRADDLE

Instruction straddles boundary. Fields:

DEBUG_STATE bool

The instruction is executed in debug state.

ISET enum

The current instruction set.

MODE enum

The mode the core is in.

NS enum

The current Secure State.

PC unsigned int

The address of the conditional instruction.

SECURITY_STATE enum

The current Security State.

IRQ_TAKEN

IRQ taken exception.

ISB

ISB instruction occurred. Fields:

PC unsigned int

The address of the ISB instruction.

LOAD_MULTIPLE_REGS_LIST

List of destination registers (not source) of a multiple registers load instruction. Fields:

REG string

Register to add into the list.

LOCAL_MONITOR

Local monitor activity. Fields:

PADDR unsigned int

Local Monitor Address.

State enum

State of the monitor (Open/Exclusive).

MAPPING_CREATE

A memory mapping has been created. This is an implementation detail of the model and so may change without notice. Fields:

ASID unsigned int

The ASID used during translation, or 0 if no ASID was used.

IS_VALID bool

Whether the mapping is valid or not.

PADDR unsigned int

Base physical address of region, or 0 if not known.

SIDE enum

I-side or D-side.

SIZE unsigned int

Size of the page, or 0 if not known.

VADDR unsigned int

Base virtual address of region.

VMID unsigned int

The VMID used during translation, or 0 if no VMID was used.

MAPPING_DESTROY

A memory mapping has been destroyed. This is an implementation detail of the model and so may change without notice. Fields:

ASID unsigned int

The ASID used during translation.

CAUSE enum

The cause of invalidation.

PADDR unsigned int

Base physical address of page.

SIDE enum

I-side or D-side.

SIZE unsigned int

Size of the page.

VADDR unsigned int

Base virtual address of page.

VMID unsigned int

The VMID used during translation.

MEMMAP_DEBUG_READ

Memory mapped reads to the debug registers. Fields:

ADDR unsigned int

address.

CORE_NUM unsigned int

Core number in a multi processor.

EXT bool

Whether access is from an external device (such as the DAP).

NS enum

Secure state of the access.

REG_NAME string

Name of the debug register accessed.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

MEMMAP_DEBUG_WRITE

Memory mapped writes to the debug registers. Fields:

ADDR unsigned int

address.

CORE_NUM unsigned int

Core number in a multi processor.

EXT bool

Whether access is from an external device (such as the DAP).

NS enum

Secure state of the access.

REG_NAME string

Name of the debug register accessed.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The new value written.

MEMTAG_LOADS

Allocation tag memory read; when attributes and system configuration allow tag accesses. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

NUM_GRANULES unsigned int

Number of tags to be Loaded/stored.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK.

TAGGING_TYPE enum

Type of Memory tagging (Physical or Virtual).

TAGS unsigned int

The tags read or written. Tags for subsequent granules would be found at subsequent bytes in this field (One byte for each tag).

TAG_VADDR unsigned int

The virtual address of the Virtual Tag if Virtual tagging is used; otherwise 0.

VADDR unsigned int

The virtual address of the access.

MENTAG_LOAD_INST

Allocation tag load instruction; regardless of tags being actually accessed. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate

(Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

NUM_GRANULES unsigned int

Number of tags to be Loaded/stored.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK.

TAGGING_TYPE enum

Type of Memory tagging (Physical or Virtual).

TAGS unsigned int

The tags read or written. Tags for subsequent granules would be found at subsequent bytes in this field (One byte for each tag).

TAG_VADDR unsigned int

The virtual address of the Virtual Tag if Virtual tagging is used; otherwise 0.

VADDR unsigned int

The virtual address of the access.

MEMTAG_STORES

Allocation tag memory write; when attributes and system configuration allow tag accesses. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

NUM_GRANULES unsigned int

Number of tags to be Loaded/stored.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK.

TAGGING_TYPE enum

Type of Memory tagging (Physical or Virtual).

TAGS unsigned int

The tags read or written. Tags for subsequent granules would be found at subsequent bytes in this field (One byte for each tag).

TAG_VADDR unsigned int

The virtual address of the Virtual Tag if Virtual tagging is used; otherwise 0.

VADDR unsigned int

The virtual address of the access.

MEMTAG_STORE_INST

Allocation tag store instruction; regardless of tags being actually accessed. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes

([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

NUM_GRANULES unsigned int

Number of tags to be Loaded/stored.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK.

TAGGING_TYPE enum

Type of Memory tagging (Physical or Virtual).

TAGS unsigned int

The tags read or written. Tags for subsequent granules would be found at subsequent bytes in this field (One byte for each tag).

TAG_VADDR unsigned int

The virtual address of the Virtual Tag if Virtual tagging is used; otherwise 0.

VADDR unsigned int

The virtual address of the access.

MMU_D128_TTB_READ

This event is triggered by reads caused by a D128 translation table walk. Fields:

ABORTED bool

Was the walk successful.

DATA_HIGH unsigned int

The data upper bits read.

DATA_LOW unsigned int

The data lower bits read.

IPA unsigned int

For stage 1, the IPA of the read.

LEVEL unsigned int

Translation table level.

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

OUTADDR unsigned int

The output physical address.

PADDR unsigned int

The physical address of the read.

PASmem enum

Physical Address Space of the accessed memory.

PASreq enum

Physical Address Space of the requested lookup.

REGIME_EL enum

Entry matches in this translation regime.

REQUESTER enum

The requester of this table walk.

SIDE enum

Inst / Data.

STAGE unsigned int

Translation stage.

MMU_D128_TTB_WRITE

This event is triggered by writes caused by a D128 translation table walk. Fields:

ABORTED bool

Was the walk successful.

DATA_HIGH unsigned int

The data upper bits written.

DATA_LOW unsigned int

The data lower bits written.

IPA unsigned int

For stage 1, the IPA of the write.

IS_HACDBS_ACCESS bool

If the access was due to HACDBS_CLEAN.

LEVEL unsigned int

Translation table level.

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

OUTADDR unsigned int

The output physical address.

PADDR unsigned int

The physical address of the write.

PASmem enum

Physical Address Space of the accessed memory.

PASreq enum

Physical Address Space of the requested lookup.

REGIME_EL enum

Entry matches in this translation regime.

SIDE enum

Inst / Data.

STAGE unsigned int

Translation stage.

MMU_TRANS

Address translation information. Fields:

ASID unsigned int

Address space identifier.

CORE_NUM unsigned int

Core number in a multi processor.

Hyp bool

Entry matches in Hyp state only.

INNERCACHE_RA bool

Is the inner cache allocate on read.

INNERCACHE_TRANSIENT bool

Is the inner write-through transient.

INNERCACHE_TYPE enum

Inner Caching scheme (NC/MB/WA).

INNERCACHE_WA bool

Is the inner cache allocate on write.

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MEMTYPE enum

Memory type.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC enum

Is secure side supposed to access secure or nonsecure memory.

OUTERCACHE_RA bool

Is the outer cache allocate on read.

OUTERCACHE_TRANSIENT bool

Is the outer write-through transient.

OUTERCACHE_TYPE enum

Outer Caching scheme (NC/MB/WA).

OUTERCACHE_WA bool

Is the outer cache allocate on write.

PADDR unsigned int

Physical address of the access.

PAGESIZE unsigned int

Page size as $\log_2(\text{size})$.

PAS enum

PAS supposed to be accessed is secure or nonsecure memory or Root/Realm if RME is implemented.

SH enum

Shareability.

SIDE enum

Inst / Data.

STAGE1_PERM unsigned int

Stage 1 permissions, [5:3] Privileged access (XWR) [2:0] User access (XWR).

STAGE2_PERM unsigned int

Stage 2 permission mask [5:3] Privileged access (XWR) [2:0] User access (XWR).

VADDR unsigned int

Virtual address of the access.

VMID unsigned int

Virtual machine identifier.

XS bool

XS Attribute.

nG enum

Flag indicating whether ASID will be matched.

MMU_TTB_READ

This event is triggered by reads caused by a translation table walk. Fields:

ABORTED bool

Was the walk successful.

DATA unsigned int

The data read.

DESC_KIND enum

The encoding scheme of the descriptors used for this walk.

IPA unsigned int

For stage 1, the IPA of the read.

LEVEL unsigned int

Translation table level.

LPAAE bool

Is this for an LPAAE translation (*DEPRECATED*, use DESC_KIND).

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSmem bool

Non secure memory access.

NSreq bool

Non secure request.

OUTADDR unsigned int

The output physical address.

PADDR unsigned int

The physical address of the read.

PASmem enum

Physical Address Space of the accessed memory.

PASreq enum

Physical Address Space of the requested lookup.

REGIME_EL enum

Entry matches in this translation regime.

REQUESTER enum

The requester of this table walk.

SIDE enum

Inst / Data.

STAGE unsigned int

Translation stage.

MMU_TTB_WRITE

This event is triggered by writes caused by a translation table walk. Fields:

ABORTED bool

Was the walk successful.

DATA unsigned int

The data written.

DESC_KIND enum

The encoding scheme of the descriptors used for this walk.

IPA unsigned int

For stage 1, the IPA of the write.

IS_HACDBS_ACCESS bool

If the access was due to HACDBS_CLEAN.

LEVEL unsigned int

Translation table level.

LPAE bool

Is this for an LPAE translation (*DEPRECATED*, use DESC_KIND).

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSmem bool

Non secure memory access.

NSreq bool

Non secure request.

OUTADDR unsigned int

The output physical address.

PADDR unsigned int

The physical address of the write.

PASmem enum

Physical Address Space of the accessed memory.

PASreq enum

Physical Address Space of the requested lookup.

REGIME_EL enum

Entry matches in this translation regime.

SIDE enum

Inst / Data.

STAGE unsigned int

Translation stage.

MODE_CHANGE

Mode change. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

MODE enum

The new mode.

NON_SECURE enum

The core's new non-secure bit.

OLD_MODE enum

The old mode.

SECURITY_STATE enum

The new security state.

MOPS_LD_RETIRED

Load instructions architecturally executed for MOPS.

MOPS_ST_RETIRED

Store instructions architecturally executed for MOPS. Fields:

IS_MEMCPY bool

Whether the memory operation is MEMCPY.

NOT_TAG_CHECKED_INST

MTE Tag not checked for current instruction.

PERIODIC

Called for every quantum. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The instruction count of this CPU.

PC unsigned int

The address of the next instruction to be executed on this CPU.

PMU_COUNTER_OVERFLOW

PMU counter overflow. Fields:

COUNTER_GROUP enum

Counter group.

COUNTER_RANGE enum

Counter range.

EVENT_ID unsigned int

Event Identifier, field is valid only for Event counter group.

INDEX unsigned int

Counter index (as selected by PMSELR), field is valid only for Event/Cycle counter groups.

INTERRUPT bool

Is interrupt enabled on overflow for this counter.

POINTER_AUTH

Called when execution hits a pointer authentication check. Fields:

AUTH_STATE *bool*

Authentication result.

KEY_USED *enum*

Pointer authentication key.

POTINDEX_UPDATE

Value of POT index for current pc changed. Fields:

NEW_POTINDEX *unsigned int*

New POTIndex.

OLD_POTINDEX *unsigned int*

Old POTIndex.

PREFETCH_MEMORY64

Prefetch from PRFM or PRFUM instructions. Fields:

PRFOP *enum*

Prefetch hint.

VADDR *unsigned int*

Virtual address of the location that should be prefetched.

PRELOAD64_REQUEST

Preload requests from PRFM or PRFUM instructions. Fields:

PRFOP *enum*

Prefetch hint.

VADDR *unsigned int*

Virtual address of the location that should be prefetched.

PRELOAD_DATA

Data preload from PLD instruction. Fields:

VADDR *unsigned int*

Virtual address of the data that should be preloaded.

PRELOAD_INST

Instruction preload from PLI instruction. Fields:

VADDR *unsigned int*

Virtual address of the instruction that should be preloaded.

PRE_CORE_LOAD

Trace just before a core load. Fields:

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

PRE_CORE_STORE

Trace just before a core store. Fields:

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

PRE_TTB_READ

This event is triggered before reads caused by a translation table walk. Fields:

ADDR unsigned int

The physical address of the read.

PSTATE

PSTATE fields update. Fields:

ALLINT unsigned int

All interrupt mask bit.

BTYPE unsigned int

Branch target identification bit.

CORE_NUM unsigned int

Core number in a multi processor.

DAIF unsigned int

Exception mask bits.

DIT unsigned int

Data Independent Timing bit.

EL unsigned int

Current Exception level field.

EXLOCK unsigned int

Exception return state lock bit.

IL unsigned int

Illegal Execution state bit.

NZCV unsigned int

Condition flags.

PACM unsigned int

PACM bit.

PAN unsigned int

Privileged Access Never state bit.

PM unsigned int

PMU exception mask bit.

PPEND unsigned int

PMU exception pending bit.

SM unsigned int

StreamingMode status bit.

SP unsigned int

Stack pointer register selection bit.

SS unsigned int

Software Step bit.

SSBS unsigned int

Speculative Store Bypass Safe bit.

TCO unsigned int

Tag Check Override bit.

TOUCHED unsigned int

Mask of the fields that have been touched since this trace has last been emitted. Note that touched does not necessarily mean the value changed. The index of the field is the bit index in the mask plus the index of this field plus one.

UAO unsigned int

User Access Override bit.

UINJ unsigned int

Undefined Instruction exceptions bit.

ZA unsigned int

ZA storage and ZT0 register enable bit.

nRW unsigned int

Current Execution state bit.

PSTATE_PACTIVE_OPMODE_TRANSITION

Trace pactive and Opmode mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Operating mode after the transition.

OLD_STATE enum

Operating mode before the transition.

PSTATE_PACTIVE_OPMODE_TRANSITION_DENIED

Trace the denied pactive and Opmode mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Operating mode after the transition.

OLD_STATE enum

Operating mode before the transition.

REASON string

Reason for the transition being denied.

PSTATE_PACTIVE_POWER_TRANSITION

Trace pactive and power mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Power mode after the transition.

OLD_STATE enum

Power mode before the transition.

PSTATE_PACTIVE_POWER_TRANSITION_DENIED

Trace the denied pactive and power mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Power mode after the transition.

OLD_STATE enum

Power mode before the transition.

REASON string

Reason for the transition being denied.

PSTATE_PACTIVE_THREAD_OPMODE_TRANSITION

Trace pactive and Opmode mode transitions for threads on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Operating mode after the transition.

OLD_STATE enum

Operating mode before the transition.

QUANTUM_START

Called at the start of every quantum.

RANGE_PRELOAD_REQUEST

Preload requests from RPRFM. Fields:

COUNT unsigned int

Number of blocks of data to access.

LENGTH unsigned int

Number of contiguous bytes to be accessed in each block.

OP unsigned int

Range prefetch operation .

REUSE unsigned int

Maximum number of bytes to be accessed before next RPRFM .

STRIDE signed int

Stride to advance after accessing LENGTH bytes.

VADDR unsigned int

Base virtual address of range prefetch request.

RUN_STATE

Run state transition. Fields:

INST_COUNT unsigned int

Ticks count at point of transition.

NEW enum

New run state.

OLD enum

Old run state.

SBP_SAMPLE_COLLISION

DISPLAY Statistical profiling samples collided .

SBP_SAMPLE_DISCARDED

DISPLAY Operation discarded by statistical profiling unit when configured in discard mode.
Fields:

BUFFER_POINTER unsigned int

Value of buffer pointer.

DATA unsigned int

Data discarded.

SIZE unsigned int

Size of sample, in bytes.

SBP_SAMPLE_END

DISPLAY Statistical profiling of an operation ended.

SBP_SAMPLE_POP

DISPLAY Operation which might be sampled by statistical profiling.

SBP_SAMPLE_RECORDED

DISPLAY Operation sampled by statistical profiling has been written to the buffer at %{BUFFER_ADDRESS} with size %{SIZE} and data %{DATA}. Fields:

BUFFER_ADDRESS unsigned int

Virtual Address of start of buffer write.

DATA unsigned int

Data written to the buffer.

SIZE unsigned int

Size of sample written to buffer, in bytes.

SBP_SAMPLE_START

DISPLAY Statistical profiling of an operation started.

SBP_SAMPLE_TAKEN

DISPLAY Operation sampled by statistical profiling after filtering.

SEMIHOSTING_CALL

Call of a semihost function occurred. Fields:

PC unsigned int

The program counter after the semihosting call.

SEMIHOSTING_PRECALL

About to call semihost. Fields:

EL2 bool

Use EL2 translation regime.

NS bool

Is Non-Secure.

REG_WIDTH unsigned int

The current register width in bytes.

SIGNAL

External signal state change. Fields:

SIGNAL enum

Signal that changed.

STATE bool

Signal asserted state.

SOFTWARE_STEP

Return the Debug Software Step state. Fields:

STATE enum

Software Step state.

SPSR

SPSR change. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

MODE enum

Which of the banked SPSR registers is written.

OLD_VALUE unsigned int

The old SPSR value.

VALUE unsigned int

The new SPSR value.

START_COMPILE

Compilation started. Fields:

VADDR unsigned int

Instruction where compilation begins.

STG_MULTIPLE_MICRO_OPS_ADDRESS_RANGE

Address range for MTE stores with multiple uops. Fields:

ADDR_END unsigned int

Final address.

ADDR_START unsigned int

Starting address.

STORE_MULTIPLE_REGS_ADDRESS_RANGE

Address range for stores with multiple registers. Fields:

ADDR_END unsigned int

Final address.

ADDR_START unsigned int

Starting address.

SUBOPTIMAL_LDST_RETIRED

suboptimal load/store. Fields:

ACCESS_SIZE unsigned int

Log2 of the access width in bytes used for alignment checking (1: half-word, 2: word ...).

OPTIMAL_ALIGN_SELECTOR unsigned int

Selector to decide the optimal alignment size (1: access width, 2: 4-bytes, 3: 8-bytes, ..., 12: 4-kbytes).

VADDR unsigned int

The virtual address of the access.

SVE_INST_SPEC

SVE operations speculatively executed.

SVE_LD_RETIRED

SVE load instructions architecturally executed.

SVE_LOADS

SVE load accesses. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ACQREL enum

Is this an acquire/release.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DATA unsigned int

The data read or written.

ELEMENT_SIZE unsigned int

Width of each element.

ENDIANNESS enum

Byte ordering of this access.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

LOCK enum

Normal, exclusive or locked access.

MECID unsigned int

Memory Encryption Context ID, if applicable.

MECID2 unsigned int

Memory Encryption Context ID of the second page, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PADDR_TAG unsigned int

If tag checked, the physical address tag for the access.

PAGE_SIZE unsigned int

Page size as $\log_2(\text{size in bytes})$.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK, 2=Exclusive Failed, 3=Unsupported.

SIZE unsigned int

Width of the access in bytes. Only required if DATA is not traced.

TAG_CHECKED bool

Is this access tag checked.

TAG_CHECKED2 bool

Is the second page access tag checked.

TRANS bool

Is this a translated access.

VADDR unsigned int

The virtual address of the access.

SVE_MEM_PREFETCH

Scalable Vector Extension memory prefetch. Fields:

CACHE_LEVEL enum

The preferred cache level.

HINT enum

What the memory is expected to be used for.

SIZE unsigned int

The prefetch size in bytes.

TEMPORAL bool

Whether the data is temporal.

VADDR unsigned int

The virtual address of the prefetch.

SVE_REG_READ

Scalable Vector Extension register read. Fields:

ID enum

The register identifier.

SM bool

Whether the PE is in Streaming Mode.

VALUE unsigned int

The read data.

SVE_REG_UPDATE

Scalable Vector Extension register update. Fields:

ID enum

The register identifier.

SM bool

Whether the PE is in Streaming Mode.

VALUE unsigned int

The written data.

SVE_REG_WRITE

Scalable Vector Extension register write. Fields:

ID enum

The register identifier.

SM bool

Whether the PE is in Streaming Mode.

VALUE unsigned int

The written data.

SVE_STORES

SVE store accesses. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ACQREL enum

Is this an acquire/release.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DATA unsigned int

The data read or written.

ELEMENT_SIZE unsigned int

Width of each element.

ENDIANNESS enum

Byte ordering of this access.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

LOCK enum

Normal, exclusive or locked access.

MECID unsigned int

Memory Encryption Context ID, if applicable.

MECID2 unsigned int

Memory Encryption Context ID of the second page, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PADDR_TAG unsigned int

If tag checked, the physical address tag for the access.

PAGE_SIZE unsigned int

Page size as $\log_2(\text{size in bytes})$.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK, 2=Exclusive Failed, 3=Unsupported.

SIZE unsigned int

Width of the access in bytes. Only required if DATA is not traced.

TAG_CHECKED bool

Is this access tag checked.

TAG_CHECKED2 bool

Is the second page access tag checked.

TRANS bool

Is this a translated access.

VADDR unsigned int

The virtual address of the access.

SVE_ST_RETIRED

SVE store instructions architecturally executed.

SVE_UNALIGNED_LDST_RETIRED

SVE load or store instructions architecturally executed that would cause alignment fault if A bit is set.

SVE_Z_PRED_MEM_ACCESS_INFO

Information about SVE/SME predicated accesses to Z registers (non Gather/Scatter). Fields:

MEM_ELEM_SIZE unsigned int

Width of elements accessed in memory (in bytes).

RnW bool

Whether this is a Read not a Write.

VA unsigned int

Virtual base address computed for accesses.

SYNC

Called for every synchronization, that is every quantum break. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The instruction count of this CPU.

LOCAL_QUANTUM unsigned int

The local quantum of this CPU.

LOCAL_TIME unsigned int

The local time of this CPU.

SYNC_REASON enum

Reason for sync.

SYSCALL

System call instruction executed. Fields:

IMM unsigned int

Immediate value of the system call instruction.

TYPE enum

System call type.

VADDR unsigned int

Instruction that caused the system call.

SYSREG_READ128

System Coprocessor 128 bit register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

REGNUM enum

Internal register number.

REGNUM_PHYS enum

Internal register number physically accessed (e.g. in presense of VHE redirection).

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_READ64

System Coprocessor register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REGNUM enum

Internal register number.

REGNUM_OPERAND enum

Register number corresponding to the instruction operand.

REGNUM_PHYS enum

Internal register number physically accessed (e.g. in presense of VHE redirection).

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_UPDATE128

Triggers when the system updates a register. Fields:

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG enum

Register number.

UNKNOWN unsigned int

Bits of the register which became unknown.

VALUE unsigned int

Value written to the register.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_UPDATE32

Triggers when the system updates a register. Fields:

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG enum

Register number.

UNKNOWN unsigned int

Bits of the register which became unknown.

VALUE unsigned int

Value written to the register.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int
opcode 2.

SYSREG_UPDATE64

Triggers when the system updates a register. Fields:

CRm unsigned int
CRm.

CRn unsigned int
CRn.

NS bool
Secure or nonsecure banked register is accessed.

REG enum
Register number.

UNKNOWN unsigned int
Bits of the register which became unknown.

VALUE unsigned int
Value written to the register.

opc unsigned int
opcode 1.

opc0 unsigned int
opcode 0.

opc2 unsigned int
opcode 2.

SYSREG_WRITE128

System Coprocessor 128 bit register write. Fields:

CORE_NUM unsigned int
Core number in a multi processor.

CRm unsigned int
CRm.

CRn unsigned int
CRn.

REGNUM enum
Internal register number.

REGNUM_PHYS enum
Internal register number physically accessed (e.g. in presense of VHE redirection).

UNDEF bool
The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_WRITE64

System Coprocessor register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REGNUM enum

Internal register number.

REGNUM_OPERAND enum

Register number corresponding to the instruction operand.

REGNUM_PHYS enum

Internal register number physically accessed (e.g. in presense of VHE redirection).

UNDEF bool

The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

TRBU_BUFFER_WRAP

DISPLAY Trace buffer current write pointer wrapped.

TRBU_TRIGGER_EVENT

DISPLAY Trace buffer Trigger Event.

TRBU_WRITE

DISPLAY PETU trace data has been written to the TRBU memory buffer at %{BUFFER_ADDRESS} with size %{SIZE} and data %{DATA}. Fields:

BUFFER_ADDRESS unsigned int

Virtual Address of start of buffer write.

DATA unsigned int

Data written to the buffer.

SIZE unsigned int

Size of sample written to buffer, in bytes.

UNALIGNED_LDST_RETIRED

Processor unaligned load/store. Fields:

ACCESS_SIZE unsigned int

Log2 of the access width in bytes used for alignment checking (1=>half-word, 2->word ...).

NUMBER_OF_BEATS unsigned int

Number of accesses (beats) in this burst (Total size in bytes is ACCESS_SIZE * NUMBER_OF_BEATS).

VADDR unsigned int

The virtual address of the access.

VFP_D_REGS

VFP/NEON D 64 bit register write. Fields:

ALIASING enum

Type of register aliasing used.

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a write to the corresponding bit in the register. For example, 0x1 would indicate a write to VALUE[0].

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

VFP_D_REGS_READ

VFP/NEON D 64 bit register read. Fields:

ID unsigned int

The register number.

VALUE unsigned int

The current value of the register.

VFP_Q_REGS

VFP/NEON Q 128 bit register write. Fields:

ALIASING enum

Type of register aliasing used.

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a write to the corresponding byte in the register. For example, 0x1 would indicate a write to VALUE[7:0].

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

VFP_Q_REGS_READ

VFP/NEON Q 128 bit register read. Fields:

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a read to the corresponding byte in the register. For example, 0x1 would indicate a read to VALUE[7:0].

VALUE unsigned int

The current value of the register.

VFP_SYS_REGS

Writes to the VFP/NEON units system registers. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ID enum

Which VFP system register is written.

MASK unsigned int

The mask indicating updated cumulative bits.

OLD_VALUE unsigned int

The register's old value overwritten.

VALUE unsigned int

The new value written to the VFP system register.

VFP_S_REGS

VFP/NEON S 32 bit register write. Fields:

ALIASING enum

Type of register aliasing used.

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a write to the corresponding bit in the register. For example, 0x1 would indicate a write to VALUE[0].

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

VFP_S_REGS_READ

VFP/NEON S 32 bit register read. Fields:

ID unsigned int

The register number.

VALUE unsigned int

The current value of the register.

VIRTUAL_TAG_ALLOCATED

Virtual tag information when allocated. Fields:

TAG_VADDR unsigned int

Virtual tag virtual address.

VADDR unsigned int

Data Virtual address.

VIRTUAL_TAG unsigned int

Virtual tag value.

WAYPOINT

Signals end of basic block execution. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ISET enum

Origin instruction set.

IS_COND bool

Indicates if this is a conditional waypoint.

PC unsigned int

Origin address (or 0 if unavailable).

TAKEN bool

Indicates if this waypoint was taken.

TARGET unsigned int

Destination address.

TARGET_ISET enum

Destination instruction set.

WFE_END

WFE ended. Fields:

INST_COUNT unsigned int

Ticks count when leaving WFE.

WFE_EVENT_REGISTER

WFE event register status: set/clear, reason. Fields:

INST_COUNT unsigned int

Ticks count.

REASON enum

Reason for set/clear. Only REASON==0 (Cleared by WFE) clears the bit, all other reasons set it.

WFE_IGNORED

WFE ignored. Fields:

EVENT bool

This WFE was ignored because the event register was set.

INST_COUNT unsigned int

Ticks count when ignoring WFE.

TRAPPED bool

This WFE was trapped.

WFE_START

WFE entered. Fields:

INST_COUNT unsigned int

Ticks count when entering WFE.

WFI_END

WFI ended. Fields:

INST_COUNT unsigned int

Ticks count when leaving WFI.

WFI_IGNORED

WFI ignored. Fields:

DISABLED bool

This WFI was ignored because WFI is disabled.

INST_COUNT unsigned int

Ticks count when ignoring WFI.

REASON enum

specifies reason why WFI trace was ignored.

TRAPPED bool

This WFI was trapped.

WFI_START

WFI entered. Fields:

INST_COUNT unsigned int

Ticks count when entering WFI.

WFI_WAKEUP

WFI wakeup. Fields:

INST_COUNT unsigned int

Ticks count when WFI wakeup occurred.

REASON enum

Reason for wakeup.

2.12 ARM_Cortex-A32

This section describes the trace sources.

AA64_ASE_SVE_REGS

Modification of AArch64 Advanced SIMD or SVE Vector Registers (via SIMD or Scalar operations), SVE Predicate Registers and ZT0. Fields:

ID **enum**

The register identifier.

MASK **unsigned int**

Bitmask of the register to signal the modified bits in the VALUE field.

SM **bool**

Whether the PE is in Streaming Mode.

VALUE **unsigned int**

Value of AArch64 vector register.

ASYNC_MEMORY_FAULT

Context ID Register write. Fields:

FAULT **unsigned int**

Fault status present in ESR.ISS encoding for Serror in ARCH64 and present in DFSR encoding for external abort format in ARCH32.

PADDR **unsigned int**

Physical Address (or 0 if unavailable).

VADDR **signed int**

Virtual Address (or 0 if unavailable).

ATOMIC_END_ACCESS

Bus trace access for atomics. Fields:

ACCESS_FAIL **bool**

Memory access failed.

ADDR **unsigned int**

The virtual address of the access.

ATTR **unsigned int**

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

COMPARE_VALUE **unsigned int**

Compare value for CAS.

LOAD_VALUE unsigned int

Loaded values.

NSDESC unsigned int

The security state of the access.

OPERAND_VALUE unsigned int

Operation's operand.

OPERATION enum

Operation type.

PADDR unsigned int

The physical address of the access.

PRIV bool

Is this a privileged access?.

ATOMIC_START_ACCESS

Bus trace access for atomics. Fields:

ADDR unsigned int

The virtual address of the access.

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

COMPARE_VALUE unsigned int

Compare value for CAS.

NSDESC unsigned int

The security state of the access.

OPERAND_VALUE unsigned int

Operation's operand.

OPERATION enum

Operation type.

PADDR unsigned int

The physical address of the access.

PRIV bool

Is this a privileged access?.

ArchMsg.Error.exit_code

DISPLAY exit code: %{EXITCODE} component kind: %{KIND} reason: %{REASON}. Fields:

EXITCODE signed int

Exit code to be returned.

KIND string

Component kind that invoked the exit code trace.

REASON string

Reason why the exit code trace was invoked (optional).

ArchMsg.Warning.IgnoredWarmResetRequest

Event fires when software attempts to request warm reset through write to deprecated DBGPRCR.CWRR.

ArchMsg.Warning.ReportedRevisionRegisterRead

DISPLAY Reported revision and patch do not match that of the cpu. Fields:

ORIGINAL_VALUE unsigned int

The value the register would have based on the modelled revision.

REGISTER string

The name of the register that has ben read.

REPORTED_VALUE unsigned int

The value the register has been overridden to.

ArchMsg.Warning.Unpredictably Indexed PM Event Register

TAGS **UNPREDICTABLE** DISPLAY Event %{IsDirect:(Selector value|Register)} %{SEL} is >= %{N}. Fields:

IsDirect unsigned int

Direct Access rather than indirect.

N unsigned int

Number of Accessible Registers.

SEL unsigned int

Selector.

ArchMsg.Warning.cache_contents_unknown

A cache has been activated without being invalidated and may contain **UNKNOWN** content. DISPLAY %{SIDE:(D|I)}-Cache was enabled in %{REGIME:(SECURE|NON_SECURE|HYP)} regime but was not invalidated since power-on: cache lines could contain **UNKNOWN** content %{SIDE:(Note: D-side caches must invalidate SECURE lines even if the cache is only used by NON_SECURE code in order to ensure that unexpected dirty lines tagged SECURE are not naturally evicted)}. Fields:

REGIME unsigned int

Current translation regime.

SIDE unsigned int

0 d-side cache or unified cache, 1 i-side.

ArchMsg.Warning.dcimvac_matches_watchpoint

TAGS **UNPREDICTABLE** DISPLAY Watchpoints matching an AArch32 DCIMVA are implementation defined.

ArchMsg.Warning.decode_fieldmismatch

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with different fields %{FIELD1} and %{FIELD2}. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_fracbitsnegative

TAGS **UNPREDICTABLE** DISPLAY VCVT Instruction is unpredictable with a negative number of fraction bits.

ArchMsg.Warning.decode_initblock

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable in an IT block.

ArchMsg.Warning.decode_initblocknotlast

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable in an IT block when not the last.

ArchMsg.Warning.decode_invalid_condition

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable when the condition is not AL.

ArchMsg.Warning.decode_invalidfieldcombination

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with the values in %{FIELD1} and %{FIELD2}. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_invalidvalue

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with the value in %{FIELD}. Fields:

FIELD enum

Field name.

ArchMsg.Warning.decode_mem_hint_unallocated

TAGS **UNPREDICTABLE** DISPLAY There is no memory hint allocated to this bit pattern.

ArchMsg.Warning.decode_registermatch

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with %{FIELD1} and %{FIELD2} as the same register. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_registermismatch

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with %{FIELD1} and %{FIELD2} as different registers. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_registeroutofrange

TAGS **UNPREDICTABLE** DISPLAY Floating-point instruction is unpredictable indexing beyond the end of the register bank.

ArchMsg.Warning.decode_sbzsbo

TAGS **UNPREDICTABLE** DISPLAY Reserved bits in the instruction are not canonical.

ArchMsg.Warning.decode_transactiontoobig

TAGS **UNPREDICTABLE** DISPLAY Attempt to perform a memory transaction of more than 128 bytes.

ArchMsg.Warning.decode_unpred_other

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable.

ArchMsg.Warning.decode_unpreduseofpc

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable when %{FIELD} is PC. Fields:

FIELD enum

Field name.

ArchMsg.Warning.decode_unpreduseofr13

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable when %{FIELD} is R13. Fields:

FIELD enum

Field name.

ArchMsg.Warning.decode_unpreduseofr13inreglist

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with R13 in register-list.

ArchMsg.Warning.decode_useofd16orhigher

TAGS **UNPREDICTABLE** DISPLAY Attempt to use register D16 or higher where it is not allowed.

ArchMsg.Warning.decode_writebackandbaseinlist

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with write back when the base register is in the transfer list.

ArchMsg.Warning.decode_zeroregistersinlist

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable when no registers are to be transferred.

ArchMsg.Warning.recursive_branch

An instruction is performing a branch that targets the same instruction. If this was intended then a WFI might be more effective. DISPLAY recursive branch detected at PC=%{PC}.

PRIMARY KEY PC. Fields:

PC unsigned int

Address of instruction causing the branch.

ArchMsg.Warning.recursive_exception

An instruction has generated an exception that targets the same instruction. DISPLAY recursive %{TYPE} exception detected at PC=%{PC} (CPSR=%{CPSR}). PRIMARY KEY PC. Fields:

CPSR unsigned int

Processor state.

PC unsigned int

Address of instruction causing the exception.

TYPE enum

Type of exception.

ArchMsg.Warning.reentrant_vector_catch

TAGS **UNPREDICTABLE** DISPLAY Vector catch debug event on Prefetch or Data abort vector %{ADDR}. Fields:

ADDR unsigned int

vector address.

ArchMsg.Warning.reserved_it_state

TAGS **UNPREDICTABLE** DISPLAY Execution with reserved IT state %{ITSTATE}. Fields:

ITSTATE unsigned int

reserved IT state.

ArchMsg.Warning.tlb_contents_unknown

Translation has been enabled without invalidating the TLB. DISPLAY Translation was enabled in the %{PAS:(Secure|Non-secure|Root|Realm)} EL%{EL:d} translation regime (vmid=%{VMID:x}) without invalidating %{INVALIDITY:(|d-side |i-side |)}entries since power-on: %{INVALIDITY:(|D||I)}TLB could contain **UNKNOWN** content. Fields:

EL unsigned int

Exception level of translation regime.

INVALIDITY unsigned int

Which TLB may be invalid (if TLBs are separate).

NON_SECURE bool

Whether the translation regime is non-secure.

PAS unsigned int

Physical address space of translation regime.

VMID unsigned int

Current VMID, if applicable.

ArchMsg.Warning.unknown_DLR_DSPSR

Exiting debug state with unknown DLR or DSPSR.

ArchMsg.Warning.unknown_ELR_SPSR

Returning from debug exception with unknown ELR or SPSR.

ArchMsg.Warning.unpred_FAR_on_LRCPC3_non_atomic_mem_access

TAGS **UNPREDICTABLE** Unpredictable FAR/FSTATUS on LRCPC3 pair instructions resulting in two separate memory accesses
DISPLAY Unpredictable FAR on LRCPC3 non atomic memory access at %{ADDR1} with FSTATUS %{FSTATUS1} or %{ADDR2} with FSTATUS %{FSTATUS2}. Fields:

ADDR1 unsigned int

Possible FAR address 1.

ADDR2 unsigned int

Possible FAR address 2.

FSTATUS1 unsigned int

Possible fault status value 1.

FSTATUS2 unsigned int

Possible fault status value 2.

ArchMsg.Warning.unpred_async_tag_check_fail_on_sve_speculative_load_do_update_ffr

Unpredictable set to FALSE of the FFR element associated to SVE speculative load with an async tag check fail TAGS **UNPREDICTABLE**.

ArchMsg.Warning.unpredictable_a32_breakpoint

TAGS **UNPREDICTABLE** DISPLAY Unpredictable breakpoint on a A32 instruction at %{ADDR1} Offset %{OFFSET} BAS %{BAS} for a %{IS_ADDRESS_MISMATCH_BREAKPOINT:(Mismatch|Match)} Breakpoint number %{N:d} Breakpoint address %{ADDR2}. Fields:

ADDR1 unsigned int

Address.

ADDR2 unsigned int

Breakpoint address.

BAS unsigned int

BAS field.

IS_ADDRESS_MISMATCH_BREAKPOINT bool

Is a Mismatch breakpoint, not a Match breakpoint.

N unsigned int

Breakpoint number.

OFFSET unsigned int

Always has a value of zero.

ArchMsg.Warning.unpredictable_banked_register_access

TAGS **UNPREDICTABLE** DISPLAY Access to banked register (R=%{R:d} SYSm=%{SYSM:2x}) is unpredictable. Fields:

R unsigned int

R bit.

SYSM unsigned int

SYSm.

ArchMsg.Warning.unpredictable_ccfail_undef

TAGS **UNPREDICTABLE** DISPLAY An instruction %{OPCODE} at VA=%{VADDR} failed its condition codes check but would otherwise have caused a non-data-dependent trap or undefined exception. Behaviour is implementation-defined choice of exception or nop. Fields:

OPCODE unsigned int

opcode of the instruction.

VADDR unsigned int

Virtual address of the instruction being executed.

ArchMsg.Warning.unpredictable_ici_bits

TAGS **UNPREDICTABLE** DISPLAY ICI bits do not apply or are architecturally unpredictable for instruction.

ArchMsg.Warning.unpredictable_t32_breakpoint

TAGS **UNPREDICTABLE** DISPLAY Unpredictable breakpoint on a T32 instruction at %{ADDR1} Offset %{OFFSET} BAS %{BAS} for a %{IS_ADDRESS_MISMATCH_BREAKPOINT:(Mismatch|Match)} Breakpoint number %{N:d} Breakpoint address %{ADDR2}. Fields:

ADDR1 unsigned int

Address.

ADDR2 unsigned int

Breakpoint address.

BAS unsigned int

BAS field.

IS_ADDRESS_MISMATCH_BREAKPOINT bool

Is a Mismatch breakpoint, not a Match breakpoint.

N unsigned int

Breakpoint number.

OFFSET unsigned int

Indicates which halfword of a 32-bit Thumb instruction the breakpoint is set on. 0 for the first halfword or if the instruction is 16 bits wide, or 1 for the second halfword.

ArchMsg.Warning.unpredictable_t32_vector_catch

TAGS **UNPREDICTABLE** DISPLAY Unpredictable Vector catch exception on a T32 instruction at %{ADDR} Offset %{OFFSET}. Fields:

ADDR unsigned int

Address.

OFFSET unsigned int

Indicates which halfword of a 32-bit Thumb instruction the vector catch exception is set on. 0 for the first halfword or if the instruction is 16 bits wide, or 1 for the second halfword.

ArchMsg.Warning.unpredictable_watchpoint_far

TAGS **UNPREDICTABLE** DISPLAY The value for FAR/EDWAR for hitting this watchpoint may be anywhere in the range %{LOWER_BOUND} to %{UPPER_BOUND}. Fields:

LOWER_BOUND unsigned int

the lowest address accessed by the instruction that triggered the watchpoint.

UPPER_BOUND unsigned int

the highest watchpointed address accessed by that instruction.

ArchMsg.Warning.unpredictable_watchpoint_far_isscmp

DISPLAY The value for FAR/EDWAR for hitting this watchpoint may be anywhere in the range %{LOWER_BOUND} to %{UPPER_BOUND}. Fields:

LOWER_BOUND unsigned int

the lowest address accessed by the instruction that triggered the watchpoint.

UPPER_BOUND unsigned int

the highest watchpointed address accessed by that instruction.

ArchMsg.Warning.warning_AdvSIMDExpandImmUnexpectedZero

TAGS **UNPREDICTABLE** DISPLAY AdvSIMDExpandImm may treat this immediate value as **UNPREDICTABLE**.

ArchMsg.Warning.warning_ConditionalSMC

TAGS **UNPREDICTABLE** DISPLAY SMC instruction has **UNPREDICTABLE** effects when conditional and when combined with traps.

ArchMsg.Warning.warning_TLBID_out_of_range

TLBID out of range in TLBI according to the configuration of NIS, NVIS, NOS and NVOS. Fields:

NIS unsigned int

NIS.

NOS unsigned int

NOS.

NVIS unsigned int

NVIS.

NVOS unsigned int

NVOS.

TLBID unsigned int

The TLBID value used in the TLBI instruction.

ArchMsg.Warning.warning_TTL_or_TG_mismatches_current_translation_granule

TTL or TG mismatches current translation granule in TLBI VA/RVA/IPA/RIPA. Fields:

CURRENT_TG unsigned int

Current TG as indicated by TCR_ELx or VTCR_EL2.

TLBI_TG unsigned int

TG (or TTL[3:2]) used in TLBI instruction.

ArchMsg.Warning.warning_access_crosses_page_boundary

TAGS **UNPREDICTABLE** DISPLAY Access crosses page boundary.

ArchMsg.Warning.warning_access_crosses_page_boundary_has_fault

TAGS **UNPREDICTABLE** DISPLAY Access that crosses page boundary has fault(s).

ArchMsg.Warning.warning_access_crosses_page_boundary_has_fault_on_both_pages

TAGS **UNPREDICTABLE** DISPLAY Access that crosses page boundary has a fault on each page.
Fields:

ADDR1 unsigned int

FAR address on first page.

ADDR2 unsigned int

FAR address on second page.

FSTATUS1 unsigned int

Fault status on first page.

FSTATUS2 unsigned int

Fault status on second page.

ArchMsg.Warning.warning_access_crosses_page_boundary_spanning_different_memory

TAGS **UNPREDICTABLE** DISPLAY Access at address %{ADDR} crosses page from MEM_TYPE %{MEMTYPE_PAGE1} to %{MEMTYPE_PAGE2}. Fields:

ADDR unsigned int

address of access crossing page.

MEMTYPE_PAGE1 signed int

First Page's memory type (0-Normal, 1-Non-Normal).

MEMTYPE_PAGE2 signed int

Second Page's memory type (0-Normal, 1-Non-Normal).

ArchMsg.Warning.warning_access_wraps_around_memory

TAGS **UNPREDICTABLE** DISPLAY Access wraps around memory in %{WIDTH:d}bit mode.
Fields:

WIDTH unsigned int

Address width.

ArchMsg.Warning.warning_bcr_linking_status

TAGS **UNPREDICTABLE** DISPLAY DBGBCR.{BT2,BT} combination gives unpredictable behaviour.

ArchMsg.Warning.warning_bcr_mask_reserved

TAGS **UNPREDICTABLE** DISPLAY DBGBCR_ELn.MASK is set to a reserved value.

ArchMsg.Warning.warning_bvr_masked_bit_not_zero

TAGS **UNPREDICTABLE** DISPLAY DBGBVRn_EL1.VA contains a bit masked by DBGBCRn_EL1.MASK that is not zero.

ArchMsg.Warning.warning_ccsidr_unimplemented_level

TAGS **UNPREDICTABLE** DISPLAY CCSIDR read while CSSELR %{CSSELR:x} points to an unimplemented cache level. Fields:

CSSELR unsigned int

current effective value of CSSELR.

ArchMsg.Warning.warning_change_to_ns_when_tge_set

TAGS **UNPREDICTABLE** DISPLAY Attempting to change NS to 1 when HCR.TGE = 1.

ArchMsg.Warning.warning_contiguous_bit_check_abort

TAGS **UNPREDICTABLE** DISPLAY An abort occurred whilst attempting to check TLB entry is contiguous at %{ENTRY_ADDR}. Fields:

ENTRY_ADDR unsigned int

address of conflicting TLB entry.

ArchMsg.Warning.warning_contiguous_bit_error

TAGS **UNPREDICTABLE** DISPLAY TLB entry at %{ENTRY_ADDR} was not contiguous with entry at %{CONTIG_ADDR} Contents are %{ENTRY_DATA} but expected %{CONTIG_DATA}. Fields:

CONTIG_ADDR unsigned int

address of the TLB first read, with which this entry is expected to be contiguous.

CONTIG_DATA unsigned int

expected contents based on the entry at CONTIG_ADDR.

ENTRY_ADDR unsigned int

address of conflicting TLB entry.

ENTRY_DATA unsigned int

contents of conflicting TLB entry.

ArchMsg.Warning.warning_contiguous_bit_error_D128

TAGS **UNPREDICTABLE** DISPLAY TLB entry at %{ENTRY_ADDR} was not contiguous with entry at %{CONTIG_ADDR} Contents are %{ENTRY_DATA_HI} %{ENTRY_DATA_LO} but expected %{CONTIG_DATA}. Fields:

CONTIG_ADDR unsigned int

address of the TLB first read, with which this entry is expected to be contiguous.

CONTIG_DATA unsigned int

expected contents based on the entry at CONTIG_ADDR.

ENTRY_ADDR unsigned int

address of conflicting TLB entry.

ENTRY_DATA_HI unsigned int

Higher bit contents of conflicting TLB entry.

ENTRY_DATA_LO unsigned int

Lower bit contents of conflicting TLB entry.

ArchMsg.Warning.warning_cp10_cp11_mismatch

TAGS **UNPREDICTABLE** DISPLAY Unpredictable configuration of CP10 and CP11 controls at EL
%{EL:d}. Fields:

EL unsigned int

EL owning the control.

ArchMsg.Warning.warning_cp10_cp11_reserved_value

TAGS **UNPREDICTABLE** DISPLAY Unpredictable configuration of CPACR CP10 (%{CP10}) or
CP11 (%{CP11}). Fields:

CP10 unsigned int

CP10 access.

CP11 unsigned int

CP11 access.

ArchMsg.Warning.warning_csselr_level_out_of_range

TAGS **UNPREDICTABLE** DISPLAY CSSELR.Level out of range, written value %{LEVEL} with only
%{IMPLEMENTED:d} levels implemented. Fields:

IMPLEMENTED unsigned int

number of cache levels implemented.

LEVEL unsigned int

written value of CSSELR.Level.

TYPE enum

The type of cache selected by CSSELR.InD and TnD.

ArchMsg.Warning.warning_debug_flow_control_bits_not_obeyed

TAGS **UNPREDICTABLE** DISPLAY Flow control bits for %{REGISTER} were not obeyed, causing
%{ERROR}. Fields:

ERROR enum

type of error resulting.

REGISTER enum

register under consideration.

ArchMsg.Warning.warning_debug_register_access_during_reset

TAGS **UNPREDICTABLE** DISPLAY Unpredictable %{IS_WRITE:(read|write)} access to debug register offset %{OFFSET} during reset. Fields:

IS_WRITE unsigned int

Write Not Read.

OFFSET unsigned int

Register Offset.

ArchMsg.Warning.warning_decode_cps_inconsistent_fields

TAGS **UNPREDICTABLE** DISPLAY The mode setting fields in the CPS instruction are inconsistent.

ArchMsg.Warning.warning_decode_invalid_state

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable in current exception-level/security state.

ArchMsg.Warning.warning_default_cacheable_mmu_on

TAGS **UNPREDICTABLE** DISPLAY Default cacheable is enabled (HCR.DC = 1) while MMU is enabled.

ArchMsg.Warning.warning_default_cacheable_vmmu_off

TAGS **UNPREDICTABLE** DISPLAY Default cacheable is enabled (HCR.DC = 1) while HCR.VM = 0.

ArchMsg.Warning.warning_deprecated_wvr_bit_2_set

TAGS **UNPREDICTABLE** DISPLAY DBGWVR%{N:d}_EL1=%{WVR:h} has bit 2 set, deprecated in ARMv8. Fields:

N unsigned int

index.

WVR unsigned int

DBGWVR.

ArchMsg.Warning.warning_eret_while_software_step_active_pending

Missing ISB between setting MDSCR_EL1.SS and ERET TAGS **UNPREDICTABLE**.

ArchMsg.Warning.warning_exclusive_to_non_normal

TAGS **UNPREDICTABLE** DISPLAY Exclusive Access to Non-Normal Memory %{ADDR}. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_exclusive_to_non_writeback

TAGS **UNPREDICTABLE** DISPLAY Exclusive Access to Non-Writeback-Cacheable Memory %{ADDR}. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_execute_from_device_memory

TAGS **UNPREDICTABLE** DISPLAY An attempt was made to execute from Device memory.

ArchMsg.Warning.warning_illegal_cpsr_mode

TAGS **UNPREDICTABLE** DISPLAY Writing an illegal or unimplement mode (%{NEW_MODE:h}) to CPSR was unpredictable. Fields:

NEW_MODE unsigned int

New value of CPSR.M.

ArchMsg.Warning.warning_illegal_srs_mode

TAGS **UNPREDICTABLE** DISPLAY Illegal or **UNPREDICTABLE** mode (%{MODE:h}) used for SRS instruction. Fields:

MODE unsigned int

mode for Banked SP.

ArchMsg.Warning.warning_implementation_defined_read_debug_register_in_SCS

Read Access to Debug registers from the processor is **IMPLEMENTATION DEFINED** TAGS **IMP_DEF**. Fields:

PPB_OFFSET unsigned int

debug register offset.

ArchMsg.Warning.warning_implementation_defined_sequential_security_transitions_supported

The behavior of sequential instruction fetches that cross from non-secure to secure memory and contain an SG instruction is **CONSTRAINED UNPREDICTABLE** TAGS **UNPREDICTABLE**.

ArchMsg.Warning.warning_implementation_defined_stack_limit_check_supported

It is **IMPLEMENTATION DEFINED** whether stack pointer limit checking is performed for this instruction TAGS **IMP_DEF**. Fields:

ADDRESS unsigned int

Address of instruction.

OPCODE unsigned int

instruction opcode.

ArchMsg.Warning.warning_implementation_defined_taggednonsharedmemorytype

TAGS **IMP_DEF** DISPLAY Memory attribute has Tagged, non shared memory type; it is **IMPLEMENTATION DEFINED** whether the memory location is treated as Tagged or non-Tagged.

ArchMsg.Warning.warning_implementation_defined_write_debug_register_in_SCS

Write Access to Debug registers from the processor is **IMPLEMENTATION DEFINED** TAGS **IMP_DEF**. Fields:

DATA unsigned int

data attempted to be written.

PPB_OFFSET unsigned int

debug register offset.

ArchMsg.Warning.warning_invalid_tcr_granule

TAGS UNPREDICTABLE DISPLAY TCR.TG{%TG_ID:(0|1)} bits have been set to a granule size {%REQUEST:d}K not implemented - using {%SUBSTITUTE:d}K. Fields:

REQUEST signed int

page size requested (or 0 for reserved).

SUBSTITUTE signed int

best guess available page size.

TG_ID bool

bits TG0 or TG1.

ArchMsg.Warning.warning_load_multiple_user_registers_from_user_mode

TAGS UNPREDICTABLE DISPLAY An LDM(user registers) instruction executed from user mode.

ArchMsg.Warning.warning_mte_in_mem_attr_not_supported

TAGS UNPREDICTABLE DISPLAY Memory attribute {%ATTR} has Tagged memory type but Armv8.5-MTE is not implemented. Fields:

ATTR unsigned int

Memory attribute.

ArchMsg.Warning.warning_reserved_breakpoint_state_match

TAGS UNPREDICTABLE DISPLAY {%IS_BREAKPOINT:(Watchpoint|Breakpoint)} programmed with a reserved combination of HMC, SSC and {%IS_BREAKPOINT:(PAC|PMC)}. Fields:

IS_BREAKPOINT unsigned int

Is a breakpoint, not a watchpoint.

ArchMsg.Warning.warning_shareability

TAGS UNPREDICTABLE DISPLAY Unpredictable: combination of the 1st and 2nd stages of translation is Normal Inner Non-Cacheable, Outer Non-Cacheable.

ArchMsg.Warning.warning_software_step_set_while_enabled

TAGS UNPREDICTABLE DISPLAY MDSCR_EL1.SS set to 1 while software step debug exceptions are enabled.

ArchMsg.Warning.warning_thumb_instruction_wraps_around_memory

TAGS UNPREDICTABLE DISPLAY Thumb instruction wraps around memory.

ArchMsg.Warning.warning_ttbr_sbz_bits_are_not_zero

TAGS UNPREDICTABLE DISPLAY {%TTBR}[{%MSB:d}:0] should be zero, but are not. Fields:

MSB unsigned int

x-1.

TTBR enum

which TTBR register.

ArchMsg.Warning.warning_unaligned_address_dbgdtrrx_write

TAGS **UNPREDICTABLE** DISPLAY Unaligned address %{ADDR} in X0/R0 when DBGDTRRX written in memory access mode. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unaligned_address_dbgdtrtx_read

TAGS **UNPREDICTABLE** DISPLAY Unaligned address %{ADDR} in X0/R0 when DBGDTRTX read in memory access mode. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unaligned_to_device

TAGS **UNPREDICTABLE** DISPLAY Unaligned Access to Device Memory %{ADDR}. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unaligned_to_strongly_ordered

TAGS **UNPREDICTABLE** DISPLAY Unaligned Access to Strongly Ordered Memory %{ADDR}. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unknown_sau_rnr

TAGS **UNPREDICTABLE** DISPLAY SAU_RNR was set to an unsupported value.

ArchMsg.Warning.warning_unpredictable_AIRCR_PRIS_and_BFHFNMINs

TAGS **UNPREDICTABLE** DISPLAY The effect of setting both AIRCR.BFHFNMINs and AIRCR.PRIS to 1 is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_AIRCR_VECTCLRACTIVE_when_not_in_debug

TAGS **UNPREDICTABLE** DISPLAY The effect of writing a 1 to AIRCR.VECTCLRACTIVE if the processor is not halted in Debug state is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_AIRCR_VECTRESET_and_SYSRESETREQ

TAGS **UNPREDICTABLE** DISPLAY When the processor is halted in Debug state, if a write to the register writes a 1 to both VECTRESET and SYSRESETREQ, the behavior is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_AIRCR_VECTRESET_when_not_in_debug

TAGS **UNPREDICTABLE** DISPLAY The effect of writing a 1 to AIRCR.VECTRESET if the processor is not halted in Debug state is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_AIRCR_incorrect_VKEY

TAGS **UNPREDICTABLE** DISPLAY The value 0x05FA must be written to AIRCR.VKEY, otherwise the register write is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_EXC_RETURN_Reserved_Bit

EXC_RETURN[23:7] are reserved with the special condition that all bits should be written as one. Values other than all 1s are **UNPREDICTABLE**. TAGS **UNPREDICTABLE**. Fields:

EXC_RETURN unsigned int

Special PC value.

SBO_MASK unsigned int

Which bits Should Be One to not UNPRED.

ArchMsg.Warning.warning_unpredictable_PMEVTYPEPn_ELO_VS_reserved_value

TAGS **UNPREDICTABLE** DISPLAY Unpredictable reserved value of PMEVTYPEn<n>_ELO.VS.

ArchMsg.Warning.warning_unpredictable_change_to_DEMCR_MON_STEP_at_insufficient_priority

TAGS **UNPREDICTABLE** DISPLAY The effect of changing DEMCR.MON_STEP at an execution priority that is lower than the priority of the DebugMonitor exception is **CONSTRAINED UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_change_to_priority_of_active_exception

TAGS **UNPREDICTABLE** DISPLAY Changing the priority of an active exception is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_clear_lspact

TAGS **UNPREDICTABLE** DISPLAY **UNPREDICTABLE** state on context switch. Fields:

DRVEXC_TAKEN unsigned int

Derived exception taken, not pended.

REACHED_LAST_STORE unsigned int

Only the last faulted.

ArchMsg.Warning.warning_unpredictable_dbg_exit_unaligned_dlr

TAGS **UNPREDICTABLE** DISPLAY Debug exit to AArch32 with DLR[0] set to 1. PC[0] can be set to 0 or to DLR[0].

ArchMsg.Warning.warning_unpredictable_exception_catch

TAGS **UNPREDICTABLE** DISPLAY Generation of an exception catch event is unpredictable.

ArchMsg.Warning.warning_unpredictable_exception_return_inconsistent_state

TAGS **UNPREDICTABLE** DISPLAY state on exception return is unpredictable.

ArchMsg.Warning.warning_unpredictable_exception_return_instruction

TAGS **UNPREDICTABLE** DISPLAY exception return instruction is unpredictable.

ArchMsg.Warning.warning_unpredictable_in_debug_state

TAGS **UNPREDICTABLE** DISPLAY Instruction is **UNPREDICTABLE** when executed in debug state.

ArchMsg.Warning.warning_unpredictable_mair_encoding

TAGS **UNPREDICTABLE** DISPLAY Memory attribute %{ATTR} has invalid MAIR encoding; this may lead to unpredictable results. Fields:

ATTR unsigned int

Memory attribute.

ArchMsg.Warning.warning_unpredictable_pmu_counter_access

TAGS **UNPREDICTABLE** DISPLAY Access to PMU counters from non-secure EL0 or EL1 is **UNPREDICTABLE** with MDCR_EL2.HPMN set to 0.

ArchMsg.Warning.warning_unpredictable_prioritization_breakpoint_match_vector_catch

TAGS **UNPREDICTABLE** DISPLAY Two events with the same priority occurred at the same instruction: Address Matching Vector Catch debug event, and Breakpoint debug event. It is constrained unpredictable which is taken.

ArchMsg.Warning.warning_unpredictable_stack_selection

TAGS **UNPREDICTABLE** DISPLAY **UNPREDICTABLE** state on context switch.

ArchMsg.Warning.warning_unpredictable_tsize_out_of_range

TAGS **UNPREDICTABLE** DISPLAY %{TCR}.T%{TTBR:d}SZ (%{TSIZE:d} bits) is out of range. Must be between 25 and %{TSIZE_MAX:d}. Fields:

TCR enum

Which TCR.

TSIZE signed int

Value of TSize.

TSIZE_MAX signed int

The maximum allowed value.

TTBR unsigned int

Which TTBR region.

ArchMsg.Warning.warning_unpredictable_unaligned_pc_as_base_register

TAGS **UNPREDICTABLE** DISPLAY The PC value must be word-aligned, otherwise the behavior of the instruction is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_unaligned_pop_stack

TAGS **UNPREDICTABLE** DISPLAY **CONSTRAINED UNPREDICTABLE** unaligned stack frame %{FRAME_ADDR:x} in exception-return. Fields:

FRAME_ADDR unsigned int

unaligned stack frame address.

ArchMsg.Warning.warning_unpredictable_unaligned_to_non_writeback

TAGS **UNPREDICTABLE** DISPLAY Unaligned Access to Non-Writeback-Cacheable Memory.

ArchMsg.Warning.warning_unpredictable_vmsa_memattrib

TAGS **UNPREDICTABLE** DISPLAY Unpredictable vmsa memory attribute with texcb value programmed to %{TEXCB} when tex remap is %{TEX_REMAP}. Fields:

TEXCB unsigned int

texcb value.

TEX_REMAP bool

use tex remap.

ArchMsg.Warning.warning_unpredictable_vtcr_t0sz_sl0

TAGS **UNPREDICTABLE** DISPLAY The combination of VTCR.TOSZ=%{TOSZ:d} and VTCR.SLO=%{SLO:d} is unpredictable. Fields:

SLO signed int

VTCR.SLO, starting level for stage 2 translation table walks.

TOSZ signed int

VTCR.TOSZ, required input address range.

ArchMsg.Warning.warning_unpredictable_write_DHCSR

TAGS **UNPREDICTABLE** DISPLAY **UNPREDICTABLE** set of bit-changes in DHCSR. Fields:

NEW unsigned int

data attempted to be written.

OLD unsigned int

previous value.

ArchMsg.Warning.warning_unpredictable_write_SHCSR

TAGS **UNPREDICTABLE** DISPLAY **UNPREDICTABLE** setting a pending bit to 1 for an exception with priority >= execution priority.

ArchMsg.Warning.warning_unsupported_access_to_memory_mapped_register

TAGS **UNPREDICTABLE** DISPLAY Access to memory mapped register is unsupported.

ArchMsg.Warning.warning_user_jmcr_access

TAGS **UNPREDICTABLE** DISPLAY User %{WRITE:(write|read)} access to JMCR is unpredictable. Fields:

WRITE unsigned int

access is write.

ArchMsg.Warning.warning_wcr_mask_and_bas

TAGS **UNPREDICTABLE** DISPLAY DBGWCR_EL1.MASK is non-zero and BAS is not 0xff.

ArchMsg.Warning.warning_wcr_mask_reserved

TAGS **UNPREDICTABLE** DISPLAY DBGWCR_ELn.MASK is set to a reserved value.

ArchMsg.Warning.warning_wcr_non_configuous_bas

TAGS **UNPREDICTABLE** DISPLAY DBGWCR_EL1.BAS has non-contiguous set of ones.

ArchMsg.Warning.warning_wvr_masked_bit_not_zero

TAGS **UNPREDICTABLE** DISPLAY DBGWVRn_EL1.VA contains a bit masked by DBGWCRn_EL1.MASK that is not zero.

ArchMsg.Why.why_branch_target_exception

A branch target exception has occurred. DISPLAY branch target exception: target pc=%{PC}, target opcode=%{OPCODE}. PRIMARY KEY PC. Fields:

BTYPE unsigned int

pstate.BTYPE at time branch taken.

OPCODE unsigned int

opcode at branch target.

PC unsigned int

Virtual Address of branch target.

ArchMsg.Why.why_illegal_state

An illegal mode change has occurred. DISPLAY illegal mode change: %{MESSAGE}. Fields:

MESSAGE string

The message.

BRANCH_MISPREDICT

Simulating branch mispredict. Fields:

PC unsigned int

Origin address (or 0 if unavailable).

BRA_DIR

Direct branches, to immediate address. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISSET enum

The instruction set of the branch instruction.

IS_COND bool

Indicates if this is a conditional branch.

IS_HINTED bool

Whether the conditional instruction is hinted.

IS_ISB bool

Whether the instruction is ISB.

IS_LINK bool

Indicates if this is a branch with link.

PC unsigned int

The address of the branch instruction.

TARGET_ISET enum

The instruction set after the branch.

TARGET_PC unsigned int

The address the instruction branches to.

BRA_INDIR

Indirect branches, perhaps to a register. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISET enum

The instruction set of the branch instruction.

IS_COND bool

Indicates if this is a conditional branch.

IS_HINTED bool

Whether the conditional instruction is hinted.

IS_ISB bool

Whether the instruction is ISB.

IS_LINK bool

Indicates if this is a branch with link.

PC unsigned int

The address of the branch instruction.

TARGET_ISET enum

The instruction set after the branch.

TARGET_PC unsigned int

The address the instruction branches to.

BRA_INDIR_CCFAIL

Indirect branches fail, perhaps to a register.

BRA_RET_CCFAIL

Branch return fail, perhaps to a register.

CACHE_MAINTENANCE_OP

Cache Maintenance Operation. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

DATA unsigned int

Specified MVA or Set/way.

FUNCTION enum

Clean / Invalidate.

SCOPE enum

Affected region.

SIDE enum

Inst / Data.

CCFAIL

Conditional instruction condition check fail. Fields:

COND **enum**

The condition of the conditional instruction.

CORE_NUM **unsigned int**

Core number in a multi processor.

PC **unsigned int**

The address of the conditional instruction.

CCFAIL_UNC

Unconditional instruction condition check fail. Fields:

COND **enum**

The condition of the conditional instruction.

CORE_NUM **unsigned int**

Core number in a multi processor.

PC **unsigned int**

The address of the conditional instruction.

CCPASS

Conditional instruction condition check pass. Fields:

COND **enum**

The condition of the conditional instruction.

CORE_NUM **unsigned int**

Core number in a multi processor.

PC **unsigned int**

The address of the conditional instruction.

CCPASS_UNC

Unconditional instruction condition check pass. Fields:

COND **enum**

The condition of the conditional instruction.

CORE_NUM **unsigned int**

Core number in a multi processor.

PC **unsigned int**

The address of the conditional instruction.

CODE_CACHE_MAINT

Code cache maintenance. Fields:

END_INCL **unsigned int**

Inclusive end address for VA or PA requests.

NS bool

Non-secure world for PA.

START unsigned int

Start address for VA or PA requests.

TYPE enum

Request type.

COMPILE_BLOCK_END

Last instruction of basic block translated. Fields:

VADDR unsigned int

Address of next instruction after this basic block.

COMPILE_BLOCK_START

First instruction of basic block translated. Fields:

VADDR unsigned int

Address of first instruction in basic block.

COMPILE_INST

ARM instruction compiled. Fields:

DISASS string

Disassembly of the instruction.

ISSET enum

The instruction set of this instruction.

ITSTATE unsigned int

The ITSTATE current for the instruction.

OPCODE unsigned int

The opcode of the instruction.

PC unsigned int

The address of the instruction.

SIZE unsigned int

The size of the instruction in bytes.

CONTEXTIDR

Context ID Register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

NS bool

Secure or nonsecure banked register is accessed.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The new value written.

CONTEXT_SYNC

Called for every context synchronization event. Fields:

ARCH_CSE bool

Architectural Context Synchronization Event.

PC unsigned int

The address of the synchronization instruction.

SYSREG_SYNC bool

System register synchronization event.

CORE_ENDIAN

Core BE8 Big-Endian state changed. Fields:

BE8 bool

Core BE8 Big-Endian state.

CORE_INFO

Static processor attributes. Only triggered by a call to DumpState(). Fields:

ARCH_PROFILE enum

The architecture profile of the core.

CLUSTER_ID unsigned int

The cluster ID of this processor.

CORE_NUM unsigned int

The number of this core in an MP processor.

DETAILED_CORE_FEATURES enum

Detailed core features as a bitfield. Enum values are bitmasks to extract the info from the field.

FPU_VERSION enum

The VFP version implemented by the core.

MEM_ARCH enum

The memory architecture of the core.

MIDR_EL1 unsigned int

MIDR_EL1 value.

NUM_CORES unsigned int

The number of cores in this MP processor.

QUANTUM_SIZE unsigned int

The default quantum size of the core (in ticks).

SECURITY_FEATURES bool

Does the core have security features?.

CORE_LOADS

Processor load accesses. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ACQREL enum

Is this an acquire/release.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DATA unsigned int

The data read or written.

ELEMENT_SIZE unsigned int

Width of each element.

ENDIANNESS enum

Byte ordering of this access.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

LOCK enum

Normal, exclusive or locked access.

MECID unsigned int

Memory Encryption Context ID, if applicable.

MECID2 unsigned int

Memory Encryption Context ID of the second page, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PAGE_SIZE unsigned int

Page size as $\log_2(\text{size in bytes})$.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK, 2=Exclusive Failed, 3=Unsupported.

SIZE unsigned int

Width of the access in bytes. Only required if DATA is not traced.

TRANS bool

Is this a translated access.

VADDR unsigned int

The virtual address of the access.

CORE_REGS

Changes of the core registers R0 to R14. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number, 0 to 14.

MODE enum

Bank of the register accessed.

OLD_VALUE unsigned int

The old value overwritten.

PHYS_ID enum

The physical register accessed.

VALUE unsigned int

The new value written to the register.

CORE_REGS64

Changes of the core registers X0..X30, SP_ELn. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ID enum

The register number, 0-30 for X0-X30, >=32 for SP_ELn.

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

CORE_REGS64_READ

General purpose AArch64 register has been read X0..X30, SP_ELn. Fields:

ID enum

The register number, 0-30 for X0-X30, >=32 for SP_ELn.

VALUE unsigned int

The value read from the register.

CORE_STALLED_TRANSITION

Detect when a core is prevented from progressing PChannel states. Fields:

CURRENT_STATE enum

Current power mode of core.

REASON string

The reason a transition was stalled, and what must be done to unblock it.

TARGET_STATE enum

Desired power mode of core after transition.

CORE_STORES

Processor store accesses. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ACQREL enum

Is this an acquire/release.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DATA unsigned int

The data read or written.

ELEMENT_SIZE unsigned int

Width of each element.

ENDIANNESS enum

Byte ordering of this access.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

LOCK enum

Normal, exclusive or locked access.

MECID unsigned int

Memory Encryption Context ID, if applicable.

MECID2 unsigned int

Memory Encryption Context ID of the second page, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PAGE_SIZE unsigned int

Page size as $\log_2(\text{size in bytes})$.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK, 2=Exclusive Failed, 3=Unsupported.

SIZE unsigned int

Width of the access in bytes. Only required if DATA is not traced.

TRANS bool

Is this a translated access.

VADDR unsigned int

The virtual address of the access.

CP14_READ

System Coprocessor register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP14 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc1 unsigned int

opcode 1.

opc2 unsigned int

opcode 2.

CP14_WRITE

System Coprocessor register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP14 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc1 unsigned int

opcode 1.

opc2 unsigned int

opcode 2.

CP15_READ

System Coprocessor register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP15 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc1 unsigned int

opcode 1.

opc2 unsigned int

opcode 2.

CP15_READ64

System Coprocessor register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP15 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc unsigned int

opcode 1.

CP15_WRITE

System Control Coprocessor register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP15 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc1 unsigned int

opcode 1.

opc2 unsigned int

opcode 2.

CP15_WRITE64

System Coprocessor register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP15 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc unsigned int

opcode 1.

CPSR

CPSR change. (In AArch64 PSTATE represented in SPSR format). Fields:

CORE_NUM unsigned int

Core number in a multi processor.

OLD_VALUE unsigned int

The old CPSR value.

UNKNOWN unsigned int

Bits within the register that have unknown value.

VALUE unsigned int

The new CPSR value.

CRYPTO_SPEC

Every crypto instruction speculatively executed.

CorePowerStateChange

Core power-up state has changed. DISPLAY Core powered %{PU:(up|down)}. Fields:

PU bool

Whether the core powered up.

DATA_CACHE_SET_ALLOC_TAGS

Set allocation tags from DC GVA instruction. Fields:

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

NSDESC unsigned int

The security state of the access.

PADDR unsigned int

The physical address of the access.

PAS enum

The physical address space of the page.

RESPONSE enum

0=Aborted, 1=OK.

SIZE unsigned int

The size of memory in bytes.

TAG unsigned int

The value of the tag being set.

TAGGING_TYPE enum

Type of Memory tagging (Physical or Virtual).

TAG_VADDR unsigned int

The virtual address of the Virtual Tag if Virtual tagging is used; otherwise 0.

VADDR unsigned int

Virtual address used for setting allocation tags.

DATA_CACHE_ZERO

Zero and invalidate cache line from DC ZVA instruction. Fields:

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

NSDESC unsigned int

The security state of the access.

PADDR unsigned int

The physical address of the access.

PADDR_TAG unsigned int

If tag checked, the physical address tag for the access.

PAS enum

The physical address space of the page.

RESPONSE enum

0=Aborted, 1=OK.

SIZE unsigned int

The size of zero'd memory in bytes.

TAG_CHECKED bool

Is this access tag checked.

VADDR unsigned int

Virtual address of the zero'd cache line.

DEBUG_EVENT

Hardware debug support event. Fields:

EVENT enum

Description of event.

VALUE unsigned int

data value.

DEBUG_TARGET_ISET

Hardware debug target ISA. Fields:

TARGET_ISET enum

The target instruction set of the debug state.

DMI_ALLOCATE

DMI allocated. Fields:

ACCESS_DIRECTION enum

READ / WRITE.

PADDR unsigned int

The physical address of the DMI.

PTR unsigned int

DMI range pointer.

SIDE enum

DSIDE / ISIDE.

SIZE unsigned int

The size of DMI.

VADDR unsigned int

The virtual address of the DMI..

DMI_HIT

DMI hit, access used DMI (only fire for small DMI regions i.e. less than a full page). Fields:

ACCESS_DIRECTION enum

READ / WRITE.

PADDR unsigned int

The physical address of the DMI.

SIDE enum

DSIDE / ISIDE.

SIZE unsigned int

The size of the DMI.

VADDR unsigned int

The virtual address of the DMI..

DMI_REVOKE

DMI revoked. Fields:

ACCESS_DIRECTION enum

READ / WRITE.

PADDR unsigned int

The physical address of the DMI.

SIDE enum

DSIDE / ISIDE.

SIZE unsigned int

The size of DMI.

VADDR unsigned int

The virtual address of the DMI..

END_COMPILE

Compilation end. Fields:

END_OF_PAGE_COUNT unsigned int

Number of basic blocks exited due to page boundary since START_COMPILE.

FETCHFAIL_COUNT unsigned int

Number of basic blocks exited due to ifetch failure START_COMPILE.

INST_COUNT unsigned int

Number of instructions compiled since START_COMPILE.

NONSEQ_COUNT unsigned int

Number of basic blocks exited due to non-sequential instructions since START_COMPILE.

PAGE_STRADDLE_COUNT unsigned int

Number of basic blocks exited due to unaligned instructions crossing page since START_COMPILE.

ESB

Error Synchronization Barrier hit. Fields:

ASYNC bool

ESB is async.

IMPLICIT bool

ESB is implicit.

EXCEPTION

Exception taken. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ESR unsigned int

The value assigned to ESR when exception is taken.

ESR_EC unsigned int

The Value assigned to ESR.EC when exception is taken.

IS_PHYSICAL bool

Physical or Virtual exception.

IS_TRAP bool

Exception coming from a trap.

LR unsigned int

The value assigned to the link register.

PC unsigned int

The location where the exception occurred.

PREFERRED_RETURN unsigned int

The preferred return address for the exception.

TAKEN_LOCALLY bool

Whether exception is taken locally(same exception level).

TARGET_ISET enum

The instruction set of the exception handler code.

TARGET_PC unsigned int

The address the exception branches to.

VECTOR enum

The exception vector.

VECTOR_OFFSET unsigned int

The offset of an exception vector from the base address.

EXCEPTION_END

Every exception completed.

EXCEPTION_INFO

Trace human-readable reason of the current exception for data/instruction aborts, floating point exceptions and unknown exceptions. Fields:

DESCRIPTION string

Reason for current exception.

ESR_EC unsigned int

The value of ESR.EC when exception is taken.

EXCEPTION_RAISE

Every exception raised.

EXCEPTION_RETURN

Branches on leaving exception. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISET enum

The instruction set of the branch instruction.

IS_COND bool

Indicates if this is a conditional branch.

IS_HINTED bool

Whether the conditional instruction is hinted.

IS_ISB bool

Whether the instruction is ISB.

IS_LINK bool

Indicates if this is a branch with link.

PC unsigned int

The address of the branch instruction.

TARGET_ISET enum

The instruction set after the branch.

TARGET_PC unsigned int

The address the instruction branches to.

EXCEPTION_RETURN_PREBRANCH

Exception return event that must occur before PC and processor state have updated.

EXCEPTION_START

Every exception started.

FIQ_TAKEN

FIQ taken exception.

FP_EXCEPTIONS_SIGNALLED

Trace which FP exceptions were signalled as a part of the FP operations performed by an instruction. Fields:

FP_EXCEPTIONS_SIGNALLED_MASK unsigned int

Indicates which FP exceptions were signalled by an instruction. This uses the same encoding as the cumulative floating-point exception bitfields in FPSR.

FP_STATE

Floating point state. Fields:

FPSR unsigned int

FPSR register value.

MASK unsigned int

The mask indicating updated cumulative bits.

FREQ_CHANGED

Core frequency changed. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The instruction count of this core.

NEW_FREQ signed int

The new frequency of this core (expressed in Hz).

OLD_FREQ signed int

The old frequency of this core (expressed in Hz).

HLT

HLT instruction occurred. Fields:

IMM16 unsigned int

The 16-bit immediate value encoded in HLT instruction.

INST

Every instruction executed. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DEBUG_STATE bool

The instruction is executed in debug state.

DISASS string

Disassembly of instruction.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISSET enum

The current instruction set.

ITSTATE unsigned int

The current ITSTATE.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

MODE enum

The mode the core is in.

NS unsigned int

The core's non-secure bit.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

OPCODE unsigned int

The opcode of the instruction.

PADDR unsigned int

The physical address of the instruction.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the instruction.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the page.

PC unsigned int

The address of the instruction.

SECURITY_STATE enum

The core's security state.

SIZE unsigned int

The size of the instruction in bytes.

INST_END

Every instruction completed.

INST_START

Every instruction started. Fields:

DEBUG_STATE bool

The instruction is executed in debug state.

ISSET enum

The current instruction set.

MODE enum

The mode the core is in.

NS enum

The current Secure State.

PC unsigned int

The address of the conditional instruction.

SECURITY_STATE enum

The current Security State.

INST_STRADDLE

Instruction straddles boundary. Fields:

DEBUG_STATE bool

The instruction is executed in debug state.

ISSET enum

The current instruction set.

MODE enum

The mode the core is in.

NS enum

The current Secure State.

PC unsigned int

The address of the conditional instruction.

SECURITY_STATE enum

The current Security State.

IRQ_TAKEN

IRQ taken exception.

ISB

ISB instruction occurred. Fields:

PC unsigned int

The address of the ISB instruction.

LOAD_MULTIPLE_REGS_LIST

List of destination registers (not source) of a multiple registers load instruction. Fields:

REG string

Register to add into the list.

LOCAL_MONITOR

Local monitor activity. Fields:

PADDR unsigned int

Local Monitor Address.

State enum

State of the monitor (Open/Exclusive).

MAPPING_CREATE

A memory mapping has been created. This is an implementation detail of the model and so may change without notice. Fields:

ASID unsigned int

The ASID used during translation, or 0 if no ASID was used.

IS_VALID bool

Whether the mapping is valid or not.

PADDR unsigned int

Base physical address of region, or 0 if not known.

SIDE enum

I-side or D-side.

SIZE unsigned int

Size of the page, or 0 if not known.

VADDR unsigned int

Base virtual address of region.

VMID unsigned int

The VMID used during translation, or 0 if no VMID was used.

MAPPING_DESTROY

A memory mapping has been destroyed. This is an implementation detail of the model and so may change without notice. Fields:

ASID unsigned int

The ASID used during translation.

CAUSE enum

The cause of invalidation.

PADDR unsigned int

Base physical address of page.

SIDE enum

I-side or D-side.

SIZE unsigned int

Size of the page.

VADDR unsigned int

Base virtual address of page.

VMID unsigned int

The VMID used during translation.

MEMMAP_DEBUG_READ

Memory mapped reads to the debug registers. Fields:

ADDR unsigned int

address.

CORE_NUM unsigned int

Core number in a multi processor.

EXT bool

Whether access is from an external device (such as the DAP).

NS enum

Secure state of the access.

REG_NAME string

Name of the debug register accessed.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

MEMMAP_DEBUG_WRITE

Memory mapped writes to the debug registers. Fields:

ADDR **unsigned int**

address.

CORE_NUM **unsigned int**

Core number in a multi processor.

EXT **bool**

Whether access is from an external device (such as the DAP).

NS **enum**

Secure state of the access.

REG_NAME **string**

Name of the debug register accessed.

UNDEF **bool**

The register accessed is undefined.

VALUE **unsigned int**

The new value written.

MMU_D128_TTB_READ

This event is triggered by reads caused by a D128 translation table walk. Fields:

ABORTED **bool**

Was the walk successful.

DATA_HIGH **unsigned int**

The data upper bits read.

DATA_LOW **unsigned int**

The data lower bits read.

IPA **unsigned int**

For stage 1, the IPA of the read.

LEVEL **unsigned int**

Translation table level.

MECID **unsigned int**

Memory Encryption Context(MEC) ID, if applicable.

MPAM_PARTID **unsigned int**

MPAM Partition ID.

MPAM_PMG **unsigned int**

MPAM Performance Monitoring Group.

MPAM_SP **unsigned int**

MPAM Partition ID Space.

OUTADDR unsigned int

The output physical address.

PADDR unsigned int

The physical address of the read.

PASmem enum

Physical Address Space of the accessed memory.

PASreq enum

Physical Address Space of the requested lookup.

REGIME_EL enum

Entry matches in this translation regime.

REQUESTER enum

The requester of this table walk.

SIDE enum

Inst / Data.

STAGE unsigned int

Translation stage.

MMU_D128_TTB_WRITE

This event is triggered by writes caused by a D128 translation table walk. Fields:

ABORTED bool

Was the walk successful.

DATA_HIGH unsigned int

The data upper bits written.

DATA_LOW unsigned int

The data lower bits written.

IPA unsigned int

For stage 1, the IPA of the write.

IS_HACDBS_ACCESS bool

If the access was due to HACDBS_CLEAN.

LEVEL unsigned int

Translation table level.

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

OUTADDR unsigned int

The output physical address.

PADDR unsigned int

The physical address of the write.

PASmem enum

Physical Address Space of the accessed memory.

PASreq enum

Physical Address Space of the requested lookup.

REGIME_EL enum

Entry matches in this translation regime.

SIDE enum

Inst / Data.

STAGE unsigned int

Translation stage.

MMU_TRANS

Address translation information. Fields:

ASID unsigned int

Address space identifier.

CORE_NUM unsigned int

Core number in a multi processor.

Hyp bool

Entry matches in Hyp state only.

INNERCACHE_RA bool

Is the inner cache allocate on read.

INNERCACHE_TRANSIENT bool

Is the inner write-through transient.

INNERCACHE_TYPE enum

Inner Caching scheme (NC/MB/WA).

INNERCACHE_WA bool

Is the inner cache allocate on write.

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MEMTYPE enum

Memory type.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC enum

Is secure side supposed to access secure or nonsecure memory.

OUTERCACHE_RA bool

Is the outer cache allocate on read.

OUTERCACHE_TRANSIENT bool

Is the outer write-through transient.

OUTERCACHE_TYPE enum

Outer Caching scheme (NC/MB/WA).

OUTERCACHE_WA bool

Is the outer cache allocate on write.

PADDR unsigned int

Physical address of the access.

PAGESIZE unsigned int

Page size as log2(size).

PAS enum

PAS supposed to be accessed is secure or nonsecure memory or Root/Realm if RME is implemented.

SH enum

Shareability.

SIDE enum

Inst / Data.

STAGE1_PERM unsigned int

Stage 1 permissions, [5:3] Privileged access (XWR) [2:0] User access (XWR).

STAGE2_PERM unsigned int

Stage 2 permission mask [5:3] Privileged access (XWR) [2:0] User access (XWR).

VADDR unsigned int

Virtual address of the access.

VMID unsigned int

Virtual machine identifier.

XS bool

XS Attribute.

nG enum

Flag indicating whether ASID will be matched.

MMU_TTB_READ

This event is triggered by reads caused by a translation table walk. Fields:

ABORTED bool

Was the walk successful.

DATA unsigned int

The data read.

DESC_KIND enum

The encoding scheme of the descriptors used for this walk.

IPA unsigned int

For stage 1, the IPA of the read.

LEVEL unsigned int

Translation table level.

LPAE bool

Is this for an LPAE translation (*DEPRECATED*, use DESC_KIND).

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSmem bool

Non secure memory access.

NSreq bool

Non secure request.

OUTADDR unsigned int

The output physical address.

PADDR unsigned int

The physical address of the read.

PASmem enum

Physical Address Space of the accessed memory.

PASreq enum

Physical Address Space of the requested lookup.

REGIME_EL enum

Entry matches in this translation regime.

REQUESTER enum

The requester of this table walk.

SIDE enum

Inst / Data.

STAGE unsigned int

Translation stage.

MMU_TTB_WRITE

This event is triggered by writes caused by a translation table walk. Fields:

ABORTED bool

Was the walk successful.

DATA unsigned int

The data written.

DESC_KIND enum

The encoding scheme of the descriptors used for this walk.

IPA unsigned int

For stage 1, the IPA of the write.

IS_HACDBS_ACCESS bool

If the access was due to HACDBS_CLEAN.

LEVEL unsigned int

Translation table level.

LPAE bool

Is this for an LPAE translation (*DEPRECATED*, use DESC_KIND).

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSmem bool

Non secure memory access.

NSreq bool

Non secure request.

OUTADDR unsigned int

The output physical address.

PADDR unsigned int

The physical address of the write.

PASmem enum

Physical Address Space of the accessed memory.

PASreq enum

Physical Address Space of the requested lookup.

REGIME_EL enum

Entry matches in this translation regime.

SIDE enum

Inst / Data.

STAGE unsigned int

Translation stage.

MODE_CHANGE

Mode change. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

MODE enum

The new mode.

NON_SECURE enum

The core's new non-secure bit.

OLD_MODE enum

The old mode.

SECURITY_STATE enum

The new security state.

NOT_TAG_CHECKED_INST

MTE Tag not checked for current instruction.

PERIODIC

Called for every quantum. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The instruction count of this CPU.

PC unsigned int

The address of the next instruction to be executed on this CPU.

PMU_COUNTER_OVERFLOW

PMU counter overflow. Fields:

COUNTER_GROUP enum

Counter group.

COUNTER_RANGE enum

Counter range.

EVENT_ID unsigned int

Event Identifier, field is valid only for Event counter group.

INDEX unsigned int

Counter index (as selected by PMSELR), field is valid only for Event/Cycle counter groups.

INTERRUPT bool

Is interrupt enabled on overflow for this counter.

POTINDEX_UPDATE

Value of POT index for current pc changed. Fields:

NEW_POTINDEX unsigned int

New POTIndex.

OLD_POTINDEX unsigned int

Old POTIndex.

PREFETCH_MEMORY64

Prefetch from PRFM or PRFUM instructions. Fields:

PRFOP enum

Prefetch hint.

VADDR unsigned int

Virtual address of the location that should be prefetched.

PRELOAD64_REQUEST

Preload requests from PRFM or PRFUM instructions. Fields:

PRFOP enum

Prefetch hint.

VADDR unsigned int

Virtual address of the location that should be prefetched.

PRELOAD_DATA

Data preload from PLD instruction. Fields:

VADDR unsigned int

Virtual address of the data that should be preloaded.

PRELOAD_INST

Instruction preload from PLI instruction. Fields:

VADDR unsigned int

Virtual address of the instruction that should be preloaded.

PRE_CORE_LOAD

Trace just before a core load. Fields:

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

PRE_CORE_STORE

Trace just before a core store. Fields:

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

PRE_TTB_READ

This event is triggered before reads caused by a translation table walk. Fields:

ADDR unsigned int

The physical address of the read.

PSTATE

PSTATE fields update. Fields:

ALLINT unsigned int

All interrupt mask bit.

BTYPE unsigned int

Branch target identification bit.

CORE_NUM unsigned int

Core number in a multi processor.

DAIF unsigned int

Exception mask bits.

DIT unsigned int

Data Independent Timing bit.

EL unsigned int

Current Exception level field.

EXLOCK unsigned int

Exception return state lock bit.

IL unsigned int

Illegal Execution state bit.

NZCV unsigned int

Condition flags.

PACM unsigned int

PACM bit.

PAN unsigned int

Privileged Access Never state bit.

PM unsigned int

PMU exception mask bit.

PPEND unsigned int

PMU exception pending bit.

SM unsigned int

StreamingMode status bit.

SP unsigned int

Stack pointer register selection bit.

SS unsigned int

Software Step bit.

SSBS unsigned int

Speculative Store Bypass Safe bit.

TCO unsigned int

Tag Check Override bit.

TOUCHED unsigned int

Mask of the fields that have been touched since this trace has last been emitted. Note that touched does not necessarily mean the value changed. The index of the field is the bit index in the mask plus the index of this field plus one.

UAO unsigned int

User Access Override bit.

UINJ unsigned int

Undefined Instruction exceptions bit.

ZA unsigned int

ZA storage and ZT0 register enable bit.

nRW unsigned int

Current Execution state bit.

PSTATE_PACTIVE_OPMODE_TRANSITION

Trace pactive and Opmode mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Operating mode after the transition.

OLD_STATE enum

Operating mode before the transition.

PSTATE_PACTIVE_OPMODE_TRANSITION_DENIED

Trace the denied pactive and Opmode mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Operating mode after the transition.

OLD_STATE enum

Operating mode before the transition.

REASON string

Reason for the transition being denied.

PSTATE_PACTIVE_POWER_TRANSITION

Trace pactive and power mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Power mode after the transition.

OLD_STATE enum

Power mode before the transition.

PSTATE_PACTIVE_POWER_TRANSITION_DENIED

Trace the denied pactive and power mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Power mode after the transition.

OLD_STATE enum

Power mode before the transition.

REASON string

Reason for the transition being denied.

PSTATE_PACTIVE_THREAD_OPMODE_TRANSITION

Trace pactive and Opmode mode transitions for threads on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Operating mode after the transition.

OLD_STATE enum

Operating mode before the transition.

QUANTUM_START

Called at the start of every quantum.

RANGE_PRELOAD_REQUEST

Preload requests from RPRFM. Fields:

COUNT unsigned int

Number of blocks of data to access.

LENGTH unsigned int

Number of contiguous bytes to be accessed in each block.

OP unsigned int

Range prefetch operation .

REUSE unsigned int

Maximum number of bytes to be accessed before next RPRFM .

STRIDE signed int

Stride to advance after accessing LENGTH bytes.

VADDR unsigned int

Base virtual address of range prefetch request.

RUN_STATE

Run state transition. Fields:

INST_COUNT unsigned int

Ticks count at point of transition.

NEW enum

New run state.

OLD enum

Old run state.

SEMIHOSTING_CALL

Call of a semihost function occurred. Fields:

PC unsigned int

The program counter after the semihosting call.

SEMIHOSTING_PRECALL

About to call semihost. Fields:

EL2 bool

Use EL2 translation regime.

NS bool

Is Non-Secure.

REG_WIDTH unsigned int

The current register width in bytes.

SIGNAL

External signal state change. Fields:

SIGNAL enum

Signal that changed.

STATE bool

Signal asserted state.

SOFTWARE_STEP

Return the Debug Software Step state. Fields:

STATE enum

Software Step state.

SPSR

SPSR change. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

MODE enum

Which of the banked SPSR registers is written.

OLD_VALUE unsigned int

The old SPSR value.

VALUE unsigned int

The new SPSR value.

START_COMPILE

Compilation started. Fields:

VADDR unsigned int

Instruction where compilation begins.

STG_MULTIPLE_MICRO_OPS_ADDRESS_RANGE

Address range for MTE stores with multiple uops. Fields:

ADDR_END unsigned int

Final address.

ADDR_START unsigned int

Starting address.

STORE_MULTIPLE_REGS_ADDRESS_RANGE

Address range for stores with multiple registers. Fields:

ADDR_END unsigned int

Final address.

ADDR_START unsigned int

Starting address.

SUBOPTIMAL_LDST_RETIRED

suboptimal load/store. Fields:

ACCESS_SIZE unsigned int

Log2 of the access width in bytes used for alignment checking (1: half-word, 2: word ...).

OPTIMAL_ALIGN_SELECTOR unsigned int

Selector to decide the optimal alignment size (1: access width, 2: 4-bytes, 3: 8-bytes, ..., 12: 4-kbytes).

VADDR unsigned int

The virtual address of the access.

SYNC

Called for every synchronization, that is every quantum break. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The instruction count of this CPU.

LOCAL_QUANTUM unsigned int

The local quantum of this CPU.

LOCAL_TIME unsigned int

The local time of this CPU.

SYNC_REASON enum

Reason for sync.

SYSCALL

System call instruction executed. Fields:

IMM unsigned int

Immediate value of the system call instruction.

TYPE enum

System call type.

VADDR unsigned int

Instruction that caused the system call.

SYSREG_READ128

System Coprocessor 128 bit register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

REGNUM enum

Internal register number.

REGNUM_PHYS enum

Internal register number physically accessed (e.g. in presense of VHE redirection).

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_READ64

System Coprocessor register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REGNUM enum

Internal register number.

REGNUM_OPERAND enum

Register number corresponding to the instruction operand.

REGNUM_PHYS enum

Internal register number physically accessed (e.g. in presense of VHE redirection).

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_UPDATE128

Triggers when the system updates a register. Fields:

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG enum

Register number.

UNKNOWN unsigned int

Bits of the register which became unknown.

VALUE unsigned int

Value written to the register.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_UPDATE32

Triggers when the system updates a register. Fields:

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG enum

Register number.

UNKNOWN unsigned int

Bits of the register which became unknown.

VALUE unsigned int

Value written to the register.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_UPDATE64

Triggers when the system updates a register. Fields:

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG enum

Register number.

UNKNOWN unsigned int

Bits of the register which became unknown.

VALUE unsigned int

Value written to the register.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_WRITE128

System Coprocessor 128 bit register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

REGNUM enum

Internal register number.

REGNUM_PHYS enum

Internal register number physically accessed (e.g. in presense of VHE redirection).

UNDEF bool

The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_WRITE64

System Coprocessor register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REGNUM enum

Internal register number.

REGNUM_OPERAND enum

Register number corresponding to the instruction operand.

REGNUM_PHYS enum

Internal register number physically accessed (e.g. in presense of VHE redirection).

UNDEF bool

The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

UNALIGNED_LDST_RETIRED

Processor unaligned load/store. Fields:

ACCESS_SIZE unsigned int

Log2 of the access width in bytes used for alignment checking (1=>half-word, 2->word ...).

NUMBER_OF_BEATS unsigned int

Number of accesses (beats) in this burst (Total size in bytes is ACCESS_SIZE * NUMBER_OF_BEATS).

VADDR unsigned int

The virtual address of the access.

VFP_D_REGS

VFP/NEON D 64 bit register write. Fields:

ALIASING enum

Type of register aliasing used.

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a write to the corresponding bit in the register. For example, 0x1 would indicate a write to VALUE[0].

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

VFP_D_REGS_READ

VFP/NEON D 64 bit register read. Fields:

ID unsigned int

The register number.

VALUE unsigned int

The current value of the register.

VFP_Q_REGS

VFP/NEON Q 128 bit register write. Fields:

ALIASING enum

Type of register aliasing used.

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a write to the corresponding byte in the register. For example, 0x1 would indicate a write to VALUE[7:0].

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

VFP_Q_REGS_READ

VFP/NEON Q 128 bit register read. Fields:

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a read to the corresponding byte in the register. For example, 0x1 would indicate a read to VALUE[7:0].

VALUE unsigned int

The current value of the register.

VFP_SYS_REGS

Writes to the VFP/NEON units system registers. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ID enum

Which VFP system register is written.

MASK unsigned int

The mask indicating updated cumulative bits.

OLD_VALUE unsigned int

The register's old value overwritten.

VALUE unsigned int

The new value written to the VFP system register.

VFP_S_REGS

VFP/NEON S 32 bit register write. Fields:

ALIASING enum

Type of register aliasing used.

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a write to the corresponding bit in the register. For example, 0x1 would indicate a write to VALUE[0].

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

VFP_S_REGS_READ

VFP/NEON S 32 bit register read. Fields:

ID unsigned int

The register number.

VALUE unsigned int

The current value of the register.

VIRTUAL_TAG_ALLOCATED

Virtual tag information when allocated. Fields:

TAG_VADDR unsigned int

Virtual tag virtual address.

VADDR unsigned int

Data Virtual address.

VIRTUAL_TAG unsigned int

Virtual tag value.

WAYPOINT

Signals end of basic block execution. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ISET enum

Origin instruction set.

IS_COND bool

Indicates if this is a conditional waypoint.

PC unsigned int

Origin address (or 0 if unavailable).

TAKEN bool

Indicates if this waypoint was taken.

TARGET unsigned int

Destination address.

TARGET_ISET enum

Destination instruction set.

WFE_END

WFE ended. Fields:

INST_COUNT unsigned int

Ticks count when leaving WFE.

WFE_EVENT_REGISTER

WFE event register status: set/clear, reason. Fields:

INST_COUNT unsigned int

Ticks count.

REASON enum

Reason for set/clear. Only REASON==0 (Cleared by WFE) clears the bit, all other reasons set it.

WFE_IGNORED

WFE ignored. Fields:

EVENT bool

This WFE was ignored because the event register was set.

INST_COUNT unsigned int

Ticks count when ignoring WFE.

TRAPPED bool

This WFE was trapped.

WFE_START

WFE entered. Fields:

INST_COUNT unsigned int

Ticks count when entering WFE.

WFI_END

WFI ended. Fields:

INST_COUNT unsigned int

Ticks count when leaving WFI.

WFI_IGNORED

WFI ignored. Fields:

DISABLED bool

This WFI was ignored because WFI is disabled.

INST_COUNT unsigned int

Ticks count when ignoring WFI.

REASON enum

specifies reason why WFI trace was ignored.

TRAPPED bool

This WFI was trapped.

WFI_START

WFI entered. Fields:

INST_COUNT unsigned int

Ticks count when entering WFI.

WFI_WAKEUP

WFI wakeup. Fields:

INST_COUNT unsigned int

Ticks count when WFI wakeup occurred.

REASON enum

Reason for wakeup.

2.13 ARM_Cortex-A320

This section describes the trace sources.

AA64_ASE_SVE_REGS

Modification of AArch64 Advanced SIMD or SVE Vector Registers (via SIMD or Scalar operations), SVE Predicate Registers and ZT0. Fields:

ID **enum**

The register identifier.

MASK **unsigned int**

Bitmask of the register to signal the modified bits in the VALUE field.

SM **bool**

Whether the PE is in Streaming Mode.

VALUE **unsigned int**

Value of AArch64 vector register.

ASYNC_MEMORY_FAULT

Context ID Register write. Fields:

FAULT **unsigned int**

Fault status present in ESR.ISS encoding for Serror in ARCH64 and present in DFSR encoding for external abort format in ARCH32.

PADDR **unsigned int**

Physical Address (or 0 if unavailable).

VADDR **signed int**

Virtual Address (or 0 if unavailable).

ATOMIC_END_ACCESS

Bus trace access for atomics. Fields:

ACCESS_FAIL **bool**

Memory access failed.

ADDR **unsigned int**

The virtual address of the access.

ATTR **unsigned int**

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

COMPARE_VALUE **unsigned int**

Compare value for CAS.

LOAD_VALUE unsigned int

Loaded values.

NSDESC unsigned int

The security state of the access.

OPERAND_VALUE unsigned int

Operation's operand.

OPERATION enum

Operation type.

PADDR unsigned int

The physical address of the access.

PRIV bool

Is this a privileged access?.

ATOMIC_START_ACCESS

Bus trace access for atomics. Fields:

ADDR unsigned int

The virtual address of the access.

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

COMPARE_VALUE unsigned int

Compare value for CAS.

NSDESC unsigned int

The security state of the access.

OPERAND_VALUE unsigned int

Operation's operand.

OPERATION enum

Operation type.

PADDR unsigned int

The physical address of the access.

PRIV bool

Is this a privileged access?.

ArchMsg.Error.exit_code

DISPLAY exit code: %{EXITCODE} component kind: %{KIND} reason: %{REASON}. Fields:

EXITCODE signed int

Exit code to be returned.

KIND string

Component kind that invoked the exit code trace.

REASON string

Reason why the exit code trace was invoked (optional).

ArchMsg.Warning.IgnoredWarmResetRequest

Event fires when software attempts to request warm reset through write to deprecated DBGPRCR.CWRR.

ArchMsg.Warning.RAS_ErrselrUnpredictableState

DISPLAY ERRSELR is in unpredictable state due to value greater than number_of_error_records written to it.

ArchMsg.Warning.RAS_UnpredictableStateWrite

DISPLAY %{REG_NAME} write ignored due to unpredictable ERRSELR state. Fields:

REG_NAME string

Name of register being written.

ArchMsg.Warning.ReportedRevisionRegisterRead

DISPLAY Reported revision and patch do not match that of the cpu. Fields:

ORIGINAL_VALUE unsigned int

The value the register would have based on the modelled revision.

REGISTER string

The name of the register that has ben read.

REPORTED_VALUE unsigned int

The value the register has been overridden to.

ArchMsg.Warning.Unpredictably Indexed PM Event Register

TAGS UNPREDICTABLE DISPLAY Event %{IsDirect:(Selector value|Register)} %{SEL} is >= %{N}. Fields:

IsDirect unsigned int

Direct Access rather than indirect.

N unsigned int

Number of Accessible Registers.

SEL unsigned int

Selector.

ArchMsg.Warning.cache_contents_unknown

A cache has been activated without being invalidated and may contain **UNKNOWN** content. DISPLAY %{SIDE:(D|I)}-Cache was enabled in %{REGIME:(SECURE|NON_SECURE|HYP)} regime but was not invalidated since power-on: cache lines could contain **UNKNOWN** content %{SIDE:(Note: D-side caches must invalidate SECURE lines even if the cache is only used by NON_SECURE code in order to ensure that unexpected dirty lines tagged SECURE are not naturally evicted)}. Fields:

REGIME unsigned int

Current translation regime.

SIDE unsigned int

0 d-side cache or unified cache, 1 i-side.

ArchMsg.Warning.dcimvac_matches_watchpoint

TAGS **UNPREDICTABLE** DISPLAY Watchpoints matching an AArch32 DCIMVA are implementation defined.

ArchMsg.Warning.decode_fieldmismatch

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with different fields %{FIELD1} and %{FIELD2}. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_fracbitsnegative

TAGS **UNPREDICTABLE** DISPLAY VCVT Instruction is unpredictable with a negative number of fraction bits.

ArchMsg.Warning.decode_initblock

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable in an IT block.

ArchMsg.Warning.decode_initblocknotlast

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable in an IT block when not the last.

ArchMsg.Warning.decode_invalid_condition

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable when the condition is not AL.

ArchMsg.Warning.decode_invalidfieldcombination

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with the values in %{FIELD1} and %{FIELD2}. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_invalidvalue

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with the value in %{FIELD}. Fields:

FIELD enum

Field name.

ArchMsg.Warning.decode_mem_hint_unallocated

TAGS **UNPREDICTABLE** DISPLAY There is no memory hint allocated to this bit pattern.

ArchMsg.Warning.decode_registermatch

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with %{FIELD1} and %{FIELD2} as the same register. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_registermismatch

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with %{FIELD1} and %{FIELD2} as different registers. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_registeroutofrange

TAGS **UNPREDICTABLE** DISPLAY Floating-point instruction is unpredictable indexing beyond the end of the register bank.

ArchMsg.Warning.decode_sbzsbo

TAGS **UNPREDICTABLE** DISPLAY Reserved bits in the instruction are not canonical.

ArchMsg.Warning.decode_transactiontoobig

TAGS **UNPREDICTABLE** DISPLAY Attempt to perform a memory transaction of more than 128 bytes.

ArchMsg.Warning.decode_unpred_other

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable.

ArchMsg.Warning.decode_unpreduseofpc

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable when %{FIELD} is PC. Fields:

FIELD enum

Field name.

ArchMsg.Warning.decode_unpreduseofr13

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable when %{FIELD} is R13. Fields:

FIELD enum

Field name.

ArchMsg.Warning.decode_unpreduseofr13inreglist

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with R13 in register-list.

ArchMsg.Warning.decode_useofd16orhigher

TAGS **UNPREDICTABLE** DISPLAY Attempt to use register D16 or higher where it is not allowed.

ArchMsg.Warning.decode_writebackandbaseinlist

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with write back when the base register is in the transfer list.

ArchMsg.Warning.decode_zeroregistersinlist

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable when no registers are to be transferred.

ArchMsg.Warning.misaligned_ptrs

DISPLAY TRBPTR_EL1(%{TRBPTR_EL1}) or TRBTRG_EL1(%{TRBTRG_EL1}) are not aligned as per TRBIDR_EL1.Align(%{TRBIDR_EL1}). Fields:

TRBIDR_EL1 unsigned int

TRBIDR_EL1.Align value.

TRBPTR_EL1 unsigned int

Current write pointer.

TRBTRG_EL1 unsigned int

Trigger pointer.

ArchMsg.Warning.out_of_range_ptrs

DISPLAY TRBPTR_EL1(%{TRBPTR_EL1}) is out of range as per TRBBASER_EL1(%{TRBBASER_EL1}) or TRBLIMITR_EL1.LIMIT(%{TRBLIMITR_EL1}). Fields:

TRBBASER_EL1 unsigned int

Base pointer.

TRBLIMITR_EL1 unsigned int

Limit pointer.

TRBPTR_EL1 unsigned int

Current write pointer.

ArchMsg.Warning.recursive_branch

An instruction is performing a branch that targets the same instruction. If this was intended then a WFI might be more effective DISPLAY recursive branch detected at PC=%{PC}.

PRIMARY KEY PC. Fields:

PC unsigned int

Address of instruction causing the branch.

ArchMsg.Warning.recursive_exception

An instruction has generated an exception that targets the same instruction. DISPLAY recursive %{TYPE} exception detected at PC=%{PC} (CPSR=%{CPSR}). PRIMARY KEY PC. Fields:

CPSR unsigned int

Processor state.

PC unsigned int

Address of instruction causing the exception.

TYPE enum

Type of exception.

ArchMsg.Warning.reentrant_vector_catch

TAGS **UNPREDICTABLE** DISPLAY Vector catch debug event on Prefetch or Data abort vector %{ADDR}. Fields:

ADDR unsigned int

vector address.

ArchMsg.Warning.reserved_it_state

TAGS **UNPREDICTABLE** DISPLAY Execution with reserved IT state %{ITSTATE}. Fields:

ITSTATE unsigned int

reserved IT state.

ArchMsg.Warning.tlb_contents_unknown

Translation has been enabled without invalidating the TLB. DISPLAY Translation was enabled in the %{PAS:(Secure|Non-secure|Root|Realm)} EL%{EL:d} translation regime (vmid=%{VMID:x}) without invalidating %{INVALIDITY:(|d-side |i-side |)}entries since power-on: %{INVALIDITY:(|D|I|)}TLB could contain **UNKNOWN** content. Fields:

EL unsigned int

Exception level of translation regime.

INVALIDITY unsigned int

Which TLB may be invalid (if TLBs are separate).

NON_SECURE bool

Whether the translation regime is non-secure.

PAS unsigned int

Physical address space of translation regime.

VMID unsigned int

Current VMID, if applicable.

ArchMsg.Warning.unknown_DLR_DSPSR

Exiting debug state with unknown DLR or DSPSR.

ArchMsg.Warning.unknown_ELR_SPSR

Returning from debug exception with unknown ELR or SPSR.

ArchMsg.Warning.unpred_FAR_on_LRCPC3_non_atomic_mem_access

TAGS **UNPREDICTABLE** Unpredictable FAR/FSTATUS on LRCPC3 pair instructions resulting in two separate memory accessesDISPLAY Unpredictable FAR on LRCPC3 non atomic memory access at %{ADDR1} with FSTATUS %{FSTATUS1} or %{ADDR2} with FSTATUS %{FSTATUS2}. Fields:

ADDR1 unsigned int

Possible FAR address 1.

ADDR2 unsigned int

Possible FAR address 2.

FSTATUS1 unsigned int

Possible fault status value 1.

FSTATUS2 unsigned int

Possible fault status value 2.

ArchMsg.Warning.unpred_async_tag_check_fail_on_sve_speculative_load_do_update_ffr

Unpredictable set to FALSE of the FFR element associated to SVE speculative load with an async tag check fail TAGS **UNPREDICTABLE**.

ArchMsg.Warning.unpredictable_a32_breakpoint

TAGS **UNPREDICTABLE** DISPLAY Unpredictable breakpoint on a A32 instruction at %{ADDR1} Offset %{OFFSET} BAS %{BAS} for a %{IS_ADDRESS_MISMATCH_BREAKPOINT:(Mismatch|Match)} Breakpoint number %{N:d} Breakpoint address %{ADDR2}. Fields:

ADDR1 unsigned int

Address.

ADDR2 unsigned int

Breakpoint address.

BAS unsigned int

BAS field.

IS_ADDRESS_MISMATCH_BREAKPOINT bool

Is a Mismatch breakpoint, not a Match breakpoint.

N unsigned int

Breakpoint number.

OFFSET unsigned int

Always has a value of zero.

ArchMsg.Warning.unpredictable_banked_register_access

TAGS **UNPREDICTABLE** DISPLAY Access to banked register (R=%{R:d} SYSm=%{SYSM:2x}) is unpredictable. Fields:

R unsigned int

R bit.

SYSM unsigned int

SYSm.

ArchMsg.Warning.unpredictable_ccfail_undef

TAGS **UNPREDICTABLE** DISPLAY An instruction %{OPCODE} at VA=%{VADDR} failed its condition codes check but would otherwise have caused a non-data-dependent trap or undefined exception. Behaviour is implementation-defined choice of exception or nop. Fields:

OPCODE unsigned int

opcode of the instruction.

VADDR unsigned int

Virtual address of the instruction being executed.

ArchMsg.Warning.unpredictable_ici_bits

TAGS **UNPREDICTABLE** DISPLAY ICI bits do not apply or are architecturally unpredictable for instruction.

ArchMsg.Warning.unpredictable_t32_breakpoint

TAGS **UNPREDICTABLE** DISPLAY Unpredictable breakpoint on a T32 instruction at %{ADDR1} Offset %{OFFSET} BAS %{BAS} for a %{IS_ADDRESS_MISMATCH_BREAKPOINT:(Mismatch|Match)} Breakpoint number %{N:d} Breakpoint address %{ADDR2}. Fields:

ADDR1 unsigned int

Address.

ADDR2 unsigned int

Breakpoint address.

BAS unsigned int

BAS field.

IS_ADDRESS_MISMATCH_BREAKPOINT bool

Is a Mismatch breakpoint, not a Match breakpoint.

N unsigned int

Breakpoint number.

OFFSET unsigned int

Indicates which halfword of a 32-bit Thumb instruction the breakpoint is set on. 0 for the first halfword or if the instruction is 16 bits wide, or 1 for the second halfword.

ArchMsg.Warning.unpredictable_t32_vector_catch

TAGS **UNPREDICTABLE** DISPLAY Unpredictable Vector catch exception on a T32 instruction at %{ADDR} Offset %{OFFSET}. Fields:

ADDR unsigned int

Address.

OFFSET unsigned int

Indicates which halfword of a 32-bit Thumb instruction the vector catch exception is set on. 0 for the first halfword or if the instruction is 16 bits wide, or 1 for the second halfword.

ArchMsg.Warning.unpredictable_watchpoint_far

TAGS **UNPREDICTABLE** DISPLAY The value for FAR/EDWAR for hitting this watchpoint may be anywhere in the range %{LOWER_BOUND} to %{UPPER_BOUND}. Fields:

LOWER_BOUND unsigned int

the lowest address accessed by the instruction that triggered the watchpoint.

UPPER_BOUND unsigned int

the highest watchpointed address accessed by that instruction.

ArchMsg.Warning.unpredictable_watchpoint_far_isscmp

DISPLAY The value for FAR/EDWAR for hitting this watchpoint may be anywhere in the range %{LOWER_BOUND} to %{UPPER_BOUND}. Fields:

LOWER_BOUND unsigned int

the lowest address accessed by the instruction that triggered the watchpoint.

UPPER_BOUND unsigned int

the highest watchpointed address accessed by that instruction.

ArchMsg.Warning.warning_AdvSIMDEExpandImmUnexpectedZero

TAGS **UNPREDICTABLE** DISPLAY AdvSIMDEExpandImm may treat this immediate value as **UNPREDICTABLE**.

ArchMsg.Warning.warning_ConditionalSMC

TAGS **UNPREDICTABLE** DISPLAY SMC instruction has **UNPREDICTABLE** effects when conditional and when combined with traps.

ArchMsg.Warning.warning_TLBID_out_of_range

TLBID out of range in TLBI according to the configuration of NIS, NVIS, NOS and NVOS. Fields:

NIS unsigned int

NIS.

NOS unsigned int

NOS.

NVIS unsigned int

NVIS.

NVOS unsigned int

NVOS.

TLBID unsigned int

The TLBID value used in the TLBI instruction.

ArchMsg.Warning.warning_TTL_or_TG_mismatches_current_translation_granule

TTL or TG mismatches current translation granule in TLBI VA/RVA/IPA/RIPA. Fields:

CURRENT_TG unsigned int

Current TG as indicated by TCR_ELx or VTCR_EL2.

TLBI_TG unsigned int

TG (or TTL[3:2]) used in TLBI instruction.

ArchMsg.Warning.warning_access_crosses_page_boundary

TAGS **UNPREDICTABLE** DISPLAY Access crosses page boundary.

ArchMsg.Warning.warning_access_crosses_page_boundary_has_fault

TAGS **UNPREDICTABLE** DISPLAY Access that crosses page boundary has fault(s).

ArchMsg.Warning.warning_access_crosses_page_boundary_has_fault_on_both_pages

TAGS **UNPREDICTABLE** DISPLAY Access that crosses page boundary has a fault on each page.

Fields:

ADDR1 unsigned int

FAR address on first page.

ADDR2 unsigned int

FAR address on second page.

FSTATUS1 unsigned int

Fault status on first page.

FSTATUS2 unsigned int

Fault status on second page.

ArchMsg.Warning.warning_access_crosses_page_boundary_spanning_different_memory

TAGS **UNPREDICTABLE** DISPLAY Access at address %{ADDR} crosses page from MEM_TYPE %{MEMTYPE_PAGE1} to %{MEMTYPE_PAGE2}. Fields:

ADDR unsigned int

address of access crossing page.

MEMTYPE_PAGE1 signed int

First Page's memory type (0-Normal, 1-Non-Normal).

MEMTYPE_PAGE2 signed int

Second Page's memory type (0-Normal, 1-Non-Normal).

ArchMsg.Warning.warning_access_wraps_around_memory

TAGS **UNPREDICTABLE** DISPLAY Access wraps around memory in %{WIDTH:d}bit mode.

Fields:

WIDTH unsigned int

Address width.

ArchMsg.Warning.warning_bcr_linking_status

TAGS **UNPREDICTABLE** DISPLAY DBGBCR.{BT2,BT} combination gives unpredictable behaviour.

ArchMsg.Warning.warning_bcr_mask_reserved

TAGS **UNPREDICTABLE** DISPLAY DBGBCR_ELn.MASK is set to a reserved value.

ArchMsg.Warning.warning_bvr_masked_bit_not_zero

TAGS **UNPREDICTABLE** DISPLAY DBGBVRn_EL1.VA contains a bit masked by DBGBCRn_EL1.MASK that is not zero.

ArchMsg.Warning.warning_ccsidr_unimplemented_level

TAGS **UNPREDICTABLE** DISPLAY CCSIDR read while CSSELR %{CSSELR:x} points to an unimplemented cache level. Fields:

CSSELR unsigned int

current effective value of CSSELR.

ArchMsg.Warning.warning_change_to_ns_when_tge_set

TAGS **UNPREDICTABLE** DISPLAY Attempting to change NS to 1 when HCR.TGE = 1.

ArchMsg.Warning.warning_contiguous_bit_check_abort

TAGS **UNPREDICTABLE** DISPLAY An abort occurred whilst attempting to check TLB entry is contiguous at %{ENTRY_ADDR}. Fields:

ENTRY_ADDR unsigned int

address of conflicting TLB entry.

ArchMsg.Warning.warning_contiguous_bit_error

TAGS **UNPREDICTABLE** DISPLAY TLB entry at %{ENTRY_ADDR} was not contiguous with entry at %{CONTIG_ADDR} Contents are %{ENTRY_DATA} but expected %{CONTIG_DATA}. Fields:

CONTIG_ADDR unsigned int

address of the TLB first read, with which this entry is expected to be contiguous.

CONTIG_DATA unsigned int

expected contents based on the entry at CONTIG_ADDR.

ENTRY_ADDR unsigned int

address of conflicting TLB entry.

ENTRY_DATA unsigned int

contents of conflicting TLB entry.

ArchMsg.Warning.warning_contiguous_bit_error_D128

TAGS **UNPREDICTABLE** DISPLAY TLB entry at %{ENTRY_ADDR} was not contiguous with entry at %{CONTIG_ADDR} Contents are %{ENTRY_DATA_HI} %{ENTRY_DATA_LO} but expected %{CONTIG_DATA}. Fields:

CONTIG_ADDR unsigned int

address of the TLB first read, with which this entry is expected to be contiguous.

CONTIG_DATA unsigned int

expected contents based on the entry at CONTIG_ADDR.

ENTRY_ADDR unsigned int

address of conflicting TLB entry.

ENTRY_DATA_HI unsigned int

Higher bit contents of conflicting TLB entry.

ENTRY_DATA_LO unsigned int

Lower bit contents of conflicting TLB entry.

ArchMsg.Warning.warning_cp10_cp11_mismatch

TAGS **UNPREDICTABLE** DISPLAY Unpredictable configuration of CP10 and CP11 controls at EL %{EL:d}. Fields:

EL unsigned int

EL owning the control.

ArchMsg.Warning.warning_cp10_cp11_reserved_value

TAGS **UNPREDICTABLE** DISPLAY Unpredictable configuration of CPACR CP10 ({CP10}) or CP11 ({CP11}). Fields:

CP10 unsigned int

CP10 access.

CP11 unsigned int

CP11 access.

ArchMsg.Warning.warning_csselr_level_out_of_range

TAGS **UNPREDICTABLE** DISPLAY CSSELR.Level out of range, written value {LEVEL} with only {IMPLEMENTED:d} levels implemented. Fields:

IMPLEMENTED unsigned int

number of cache levels implemented.

LEVEL unsigned int

written value of CSSELR.Level.

TYPE enum

The type of cache selected by CSSELR.InD and TnD.

ArchMsg.Warning.warning_debug_flow_control_bits_not_obeyed

TAGS **UNPREDICTABLE** DISPLAY Flow control bits for {REGISTER} were not obeyed, causing {ERROR}. Fields:

ERROR enum

type of error resulting.

REGISTER enum

register under consideration.

ArchMsg.Warning.warning_debug_register_access_during_reset

TAGS **UNPREDICTABLE** DISPLAY Unpredictable {IS_WRITE:(read|write)} access to debug register offset {OFFSET} during reset. Fields:

IS_WRITE unsigned int

Write Not Read.

OFFSET unsigned int

Register Offset.

ArchMsg.Warning.warning_decode_cps_inconsistent_fields

TAGS **UNPREDICTABLE** DISPLAY The mode setting fields in the CPS instruction are inconsistent.

ArchMsg.Warning.warning_decode_invalid_state

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable in current exception-level/security state.

ArchMsg.Warning.warning_default_cacheable_mmu_on

TAGS **UNPREDICTABLE** DISPLAY Default cacheable is enabled (HCR.DC = 1) while MMU is enabled.

ArchMsg.Warning.warning_default_cacheable_vmmu_off

TAGS **UNPREDICTABLE** DISPLAY Default cacheable is enabled (HCR.DC = 1) while HCR.VM = 0.

ArchMsg.Warning.warning_deprecated_wvr_bit_2_set

TAGS **UNPREDICTABLE** DISPLAY DBGWVR%{N:d}_EL1=%{WVR:h} has bit 2 set, deprecated in ARMv8. Fields:

N unsigned int

index.

WVR unsigned int

DBGWVR.

ArchMsg.Warning.warning_eret_while_software_step_active_pending

Missing ISB between setting MDSCR_EL1.SS and ERET TAGS **UNPREDICTABLE**.

ArchMsg.Warning.warning_exclusive_to_non_normal

TAGS **UNPREDICTABLE** DISPLAY Exclusive Access to Non-Normal Memory %{ADDR}. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_exclusive_to_non_writeback

TAGS **UNPREDICTABLE** DISPLAY Exclusive Access to Non-Writeback-Cacheable Memory %{ADDR}. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_execute_from_device_memory

TAGS **UNPREDICTABLE** DISPLAY An attempt was made to execute from Device memory.

ArchMsg.Warning.warning_illegal_cpsr_mode

TAGS **UNPREDICTABLE** DISPLAY Writing an illegal or unimplement mode (%{NEW_MODE:h}) to CPSR was unpredictable. Fields:

NEW_MODE unsigned int

New value of CPSR.M.

ArchMsg.Warning.warning_illegal_srs_mode

TAGS **UNPREDICTABLE** DISPLAY Illegal or **UNPREDICTABLE** mode (%{MODE:h}) used for SRS instruction. Fields:

MODE unsigned int

mode for Banked SP.

ArchMsg.Warning.warning_implementation_defined_read_debug_register_in_SCS

Read Access to Debug registers from the processor is IMPLEMENTATION DEFINED TAGS IMP_DEF. Fields:

PPB_OFFSET unsigned int
debug register offset.

ArchMsg.Warning.warning_implementation_defined_sequential_security_transitions_supported

The behavior of sequential instruction fetches that cross from non-secure to secure memory and contain an SG instruction is CONSTRAINED UNPREDICTABLE TAGS UNPREDICTABLE.

ArchMsg.Warning.warning_implementation_defined_stack_limit_check_supported

It is IMPLEMENTATION DEFINED whether stack pointer limit checking is performed for this instruction TAGS IMP_DEF. Fields:

ADDRESS unsigned int
Address of instruction.

OPCODE unsigned int
instruction opcode.

ArchMsg.Warning.warning_implementation_defined_taggednonsharedmemorytype

TAGS IMP_DEF DISPLAY Memory attribute has Tagged, non shared memory type; it is IMPLEMENTATION DEFINED whether the memory location is treated as Tagged or non-Tagged.

ArchMsg.Warning.warning_implementation_defined_write_debug_register_in_SCS

Write Access to Debug registers from the processor is IMPLEMENTATION DEFINED TAGS IMP_DEF. Fields:

DATA unsigned int
data attempted to be written.

PPB_OFFSET unsigned int
debug register offset.

ArchMsg.Warning.warning_invalid_tcr_granule

TAGS UNPREDICTABLE DISPLAY TCR.TG%{TG_ID:(0|1)} bits have been set to a granule size %{{REQUEST:d}}K not implemented - using %{{SUBSTITUTE:d}}K. Fields:

REQUEST signed int
page size requested (or 0 for reserved).

SUBSTITUTE signed int
best guess available page size.

TG_ID bool
bits TG0 or TG1.

ArchMsg.Warning.warning_load_multiple_user_registers_from_user_mode

TAGS UNPREDICTABLE DISPLAY An LDM(user registers) instruction executed from user mode.

ArchMsg.Warning.warning_mte_in_mem_attr_not_supported

TAGS UNPREDICTABLE DISPLAY Memory attribute %{ATTR} has Tagged memory type but Armv8.5-MTE is not implemented. Fields:

ATTR unsigned int

Memory attribute.

ArchMsg.Warning.warning_reserved_breakpoint_state_match

TAGS UNPREDICTABLE DISPLAY %{IS_BREAKPOINT:(Watchpoint|Breakpoint)} programmed with a reserved combination of HMC, SSC and %{IS_BREAKPOINT:(PAC|PMC)}. Fields:

IS_BREAKPOINT unsigned int

Is a breakpoint, not a watchpoint.

ArchMsg.Warning.warning_shareability

TAGS UNPREDICTABLE DISPLAY Unpredictable: combination of the 1st and 2nd stages of translation is Normal Inner Non-Cacheable, Outer Non-Cacheable.

ArchMsg.Warning.warning_software_step_set_while_enabled

TAGS UNPREDICTABLE DISPLAY MDSCR_EL1.SS set to 1 while software step debug exceptions are enabled.

ArchMsg.Warning.warning_thumb_instruction_wraps_around_memory

TAGS UNPREDICTABLE DISPLAY Thumb instruction wraps around memory.

ArchMsg.Warning.warning_ttbr_sbz_bits_are_not_zero

TAGS UNPREDICTABLE DISPLAY %{TTBR}[%{MSB:d}:0] should be zero, but are not. Fields:

MSB unsigned int

x-1.

TTBR enum

which TTBR register.

ArchMsg.Warning.warning_unaligned_address_dbgdtrrx_write

TAGS UNPREDICTABLE DISPLAY Unaligned address %{ADDR} in X0/R0 when DBGDTRRX written in memory access mode. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unaligned_address_dbgdtrtx_read

TAGS UNPREDICTABLE DISPLAY Unaligned address %{ADDR} in X0/R0 when DBGDTRTX read in memory access mode. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unaligned_to_device

TAGS UNPREDICTABLE DISPLAY Unaligned Access to Device Memory %{ADDR}. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unaligned_to_strongly_ordered

TAGS **UNPREDICTABLE** DISPLAY Unaligned Access to Strongly Ordered Memory %{ADDR}.

Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unknown_sau_rnr

TAGS **UNPREDICTABLE** DISPLAY SAU_RNR was set to an unsupported value.

ArchMsg.Warning.warning_unpredictable_AIRCR_PRIS_and_BFHFNMINs

TAGS **UNPREDICTABLE** DISPLAY The effect of setting both AIRCR.BFHFNMINs and AIRCR.PRIS to 1 is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_AIRCR_VECTCLRACTIVE_when_not_in_debug

TAGS **UNPREDICTABLE** DISPLAY The effect of writing a 1 to AIRCR.VECTCLRACTIVE if the processor is not halted in Debug state is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_AIRCR_VECTRESET_and_SYSRESETREQ

TAGS **UNPREDICTABLE** DISPLAY When the processor is halted in Debug state, if a write to the register writes a 1 to both VECTRESET and SYSRESETREQ, the behavior is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_AIRCR_VECTRESET_when_not_in_debug

TAGS **UNPREDICTABLE** DISPLAY The effect of writing a 1 to AIRCR.VECTRESET if the processor is not halted in Debug state is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_AIRCR_incorrect_VKEY

TAGS **UNPREDICTABLE** DISPLAY The value 0x05FA must be written to AIRCR.VKEY, otherwise the register write is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_EXC_RETURN_Reserved_Bit

EXC_RETURN[23:7] are reserved with the special condition that all bits should be written as one. Values other than all 1s are **UNPREDICTABLE**. TAGS **UNPREDICTABLE**. Fields:

EXC_RETURN unsigned int

Special PC value.

SBO_MASK unsigned int

Which bits Should Be One to not UNPRED.

ArchMsg.Warning.warning_unpredictable_PMEVTYPEPERn_ELO_VS_reserved_value

TAGS **UNPREDICTABLE** DISPLAY Unpredictable reserved value of PMEVTYPER<n>_ELO.VS.

ArchMsg.Warning.warning_unpredictable_change_to_DEMCR_MON_STEP_at_insufficient_priority

TAGS **UNPREDICTABLE** DISPLAY The effect of changing DEMCR.MON_STEP at an execution priority that is lower than the priority of the DebugMonitor exception is **CONSTRAINED UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_change_to_priority_of_active_exception

TAGS UNPREDICTABLE DISPLAY Changing the priority of an active exception is UNPREDICTABLE.

ArchMsg.Warning.warning_unpredictable_clear_lspact

TAGS UNPREDICTABLE DISPLAY UNPREDICTABLE state on context switch. Fields:

DRVEXC_TAKEN unsigned int

Derived exception taken, not pended.

REACHED_LAST_STORE unsigned int

Only the last faulted.

ArchMsg.Warning.warning_unpredictable_dbg_exit_unaligned_dlr

TAGS UNPREDICTABLE DISPLAY Debug exit to AArch32 with DLR[0] set to 1. PC[0] can be set to 0 or to DLR[0].

ArchMsg.Warning.warning_unpredictable_exception_catch

TAGS UNPREDICTABLE DISPLAY Generation of an exception catch event is unpredictable.

ArchMsg.Warning.warning_unpredictable_exception_return_inconsistent_state

TAGS UNPREDICTABLE DISPLAY state on exception return is unpredictable.

ArchMsg.Warning.warning_unpredictable_exception_return_instruction

TAGS UNPREDICTABLE DISPLAY exception return instruction is unpredictable.

ArchMsg.Warning.warning_unpredictable_in_debug_state

TAGS UNPREDICTABLE DISPLAY Instruction is UNPREDICTABLE when executed in debug state.

ArchMsg.Warning.warning_unpredictable_mair_encoding

TAGS UNPREDICTABLE DISPLAY Memory attribute %{ATTR} has invalid MAIR encoding; this may lead to unpredictable results. Fields:

ATTR unsigned int

Memory attribute.

ArchMsg.Warning.warning_unpredictable_pmu_counter_access

TAGS UNPREDICTABLE DISPLAY Access to PMU counters from non-secure EL0 or EL1 is UNPREDICTABLE with MDCR_EL2.HPMN set to 0.

ArchMsg.Warning.warning_unpredictable_prioritization_breakpoint_match_vector_catch

TAGS UNPREDICTABLE DISPLAY Two events with the same priority occurred at the same instruction: Address Matching Vector Catch debug event, and Breakpoint debug event. It is constrained unpredictable which is taken.

ArchMsg.Warning.warning_unpredictable_stack_selection

TAGS UNPREDICTABLE DISPLAY UNPREDICTABLE state on context switch.

ArchMsg.Warning.warning_unpredictable_tsize_out_of_range

TAGS UNPREDICTABLE DISPLAY %{TCR}.T%{TTBR:d}SZ (%{TSIZE:d} bits) is out of range. Must be between 25 and %{TSIZE_MAX:d}. Fields:

TCR enum

Which TCR.

TSize signed int

Value of TSize.

TSize_MAX signed int

The maximum allowed value.

TTBR unsigned int

Which TTBR region.

ArchMsg.Warning.warning_unpredictable_unaligned_pc_as_base_register

TAGS **UNPREDICTABLE** DISPLAY The PC value must be word-aligned, otherwise the behavior of the instruction is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_unaligned_pop_stack

TAGS **UNPREDICTABLE** DISPLAY **CONSTRAINED UNPREDICTABLE** unaligned stack frame `%{FRAME_ADDR:x}` in exception-return. Fields:

FRAME_ADDR unsigned int

unaligned stack frame address.

ArchMsg.Warning.warning_unpredictable_unaligned_to_non_writeback

TAGS **UNPREDICTABLE** DISPLAY Unaligned Access to Non-Writeback-Cacheable Memory.

ArchMsg.Warning.warning_unpredictable_vmsa_memattrib

TAGS **UNPREDICTABLE** DISPLAY Unpredictable vmsa memory attribute with texcb value programmed to `%{TEXCB}` when tex remap is `%{TEX_REMAP}`. Fields:

TEXCB unsigned int

texcb value.

TEX_REMAP bool

use tex remap.

ArchMsg.Warning.warning_unpredictable_vtcr_t0sz_sl0

TAGS **UNPREDICTABLE** DISPLAY The combination of VTCR.TOSZ=`%{TOSZ:d}` and VTCR.SL0=`%{SL0:d}` is unpredictable. Fields:

SL0 signed int

VTCR.SL0, starting level for stage 2 translation table walks.

TOSZ signed int

VTCR.TOSZ, required input address range.

ArchMsg.Warning.warning_unpredictable_write_DHCSR

TAGS **UNPREDICTABLE** DISPLAY **UNPREDICTABLE** set of bit-changes in DHCSR. Fields:

NEW unsigned int

data attempted to be written.

OLD unsigned int

previous value.

ArchMsg.Warning.warning_unpredictable_write_SHCSR

TAGS UNPREDICTABLE DISPLAY UNPREDICTABLE setting a pending bit to 1 for an exception with priority \geq execution priority.

ArchMsg.Warning.warning_unsupported_access_to_memory_mapped_register

TAGS UNPREDICTABLE DISPLAY Access to memory mapped register is unsupported.

ArchMsg.Warning.warning_user_jmcr_access

TAGS UNPREDICTABLE DISPLAY User %{WRITE:(write|read)} access to JMCR is unpredictable. Fields:

WRITE unsigned int
access is write.

ArchMsg.Warning.warning_wcr_mask_and_bas

TAGS UNPREDICTABLE DISPLAY DBGWCR_EL1.MASK is non-zero and BAS is not 0xff.

ArchMsg.Warning.warning_wcr_mask_reserved

TAGS UNPREDICTABLE DISPLAY DBGWCR_ELn.MASK is set to a reserved value.

ArchMsg.Warning.warning_wcr_non_configuous_bas

TAGS UNPREDICTABLE DISPLAY DBGWCR_EL1.BAS has non-contiguous set of ones.

ArchMsg.Warning.warning_wvr_masked_bit_not_zero

TAGS UNPREDICTABLE DISPLAY DBGWVRn_EL1.VA contains a bit masked by DBGWCRn_EL1.MASK that is not zero.

ArchMsg.Why.why_branch_target_exception

A branch target exception has occurred. DISPLAY branch target exception: target pc=%{PC}, target opcode=%{OPCODE}. PRIMARY KEY PC. Fields:

BTYPE unsigned int
pstate.BTYPE at time branch taken.

OPCODE unsigned int
opcode at branch target.

PC unsigned int
Virtual Address of branch target.

ArchMsg.Why.why_illegal_state

An illegal mode change has occurred. DISPLAY illegal mode change: %{MESSAGE}. Fields:

MESSAGE string
The message.

BRANCH_MISPREDICT

Simulating branch mispredict. Fields:

PC unsigned int
Origin address (or 0 if unavailable).

BRA_DIR

Direct branches, to immediate address. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISET enum

The instruction set of the branch instruction.

IS_COND bool

Indicates if this is a conditional branch.

IS_HINTED bool

Whether the conditional instruction is hinted.

IS_ISB bool

Whether the instruction is ISB.

IS_LINK bool

Indicates if this is a branch with link.

PC unsigned int

The address of the branch instruction.

TARGET_ISET enum

The instruction set after the branch.

TARGET_PC unsigned int

The address the instruction branches to.

BRA_INDIR

Indirect branches, perhaps to a register. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISET enum

The instruction set of the branch instruction.

IS_COND bool

Indicates if this is a conditional branch.

IS_HINTED bool

Whether the conditional instruction is hinted.

IS_ISB bool

Whether the instruction is ISB.

IS_LINK bool

Indicates if this is a branch with link.

PC unsigned int

The address of the branch instruction.

TARGET_ISET enum

The instruction set after the branch.

TARGET_PC unsigned int

The address the instruction branches to.

BRA_INDIR_CCFAIL

Indirect branches fail, perhaps to a register.

BRA_RET_CCFAIL

Branch return fail, perhaps to a register.

CACHE_MAINTENANCE_OP

Cache Maintenance Operation. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

DATA unsigned int

Specified MVA or Set/way.

FUNCTION enum

Clean / Invalidate.

SCOPE enum

Affected region.

SIDE enum

Inst / Data.

CCFAIL

Conditional instruction condition check fail. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CCFAIL_UNC

Unconditional instruction condition check fail. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CCPASS

Conditional instruction condition check pass. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CCPASS_UNC

Unconditional instruction condition check pass. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CODE_CACHE_MAINT

Code cache maintenance. Fields:

END_INCL unsigned int

Inclusive end address for VA or PA requests.

NS bool

Non-secure world for PA.

START unsigned int

Start address for VA or PA requests.

TYPE enum

Request type.

COMPILE_BLOCK_END

Last instruction of basic block translated. Fields:

VADDR unsigned int

Address of next instruction after this basic block.

COMPILE_BLOCK_START

First instruction of basic block translated. Fields:

VADDR unsigned int

Address of first instruction in basic block.

COMPILE_INST

ARM instruction compiled. Fields:

DISASS string

Disassembly of the instruction.

ISET enum

The instruction set of this instruction.

ITSTATE unsigned int

The ITSTATE current for the instruction.

OPCODE unsigned int

The opcode of the instruction.

PC unsigned int

The address of the instruction.

SIZE unsigned int

The size of the instruction in bytes.

CONTEXTIDR

Context ID Register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

NS bool

Secure or nonsecure banked register is accessed.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The new value written.

CONTEXT_SYNC

Called for every context synchronization event. Fields:

ARCH_CSE bool

Architectural Context Synchronization Event.

PC unsigned int

The address of the synchronization instruction.

SYSREG_SYNC bool

System register synchronization event.

CORE_ENDIAN

Core BE8 Big-Endian state changed. Fields:

BE8 bool

Core BE8 Big-Endian state.

CORE_INFO

Static processor attributes. Only triggered by a call to DumpState(). Fields:

ARCH_PROFILE enum

The architecture profile of the core.

CLUSTER_ID unsigned int

The cluster ID of this processor.

CORE_NUM unsigned int

The number of this core in an MP processor.

DETAILED_CORE_FEATURES enum

Detailed core features as a bitfield. Enum values are bitmasks to extract the info from the field.

FPU_VERSION enum

The VFP version implemented by the core.

MEM_ARCH enum

The memory architecture of the core.

MIDR_EL1 unsigned int

MIDR_EL1 value.

NUM_CORES unsigned int

The number of cores in this MP processor.

QUANTUM_SIZE unsigned int

The default quantum size of the core (in ticks).

SECURITY_FEATURES bool

Does the core have security features?.

CORE_LOADS

Processor load accesses. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ACQREL enum

Is this an acquire/release.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DATA unsigned int

The data read or written.

ELEMENT_SIZE unsigned int

Width of each element.

ENDIANNESS enum

Byte ordering of this access.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

LOCK enum

Normal, exclusive or locked access.

MECID unsigned int

Memory Encryption Context ID, if applicable.

MECID2 unsigned int

Memory Encryption Context ID of the second page, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PADDR_TAG unsigned int

If tag checked, the physical address tag for the access.

PAGE_SIZE unsigned int

Page size as $\log_2(\text{size in bytes})$.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK, 2=Exclusive Failed, 3=Unsupported.

SIZE unsigned int

Width of the access in bytes. Only required if DATA is not traced.

TAG_CHECKED bool

Is this access tag checked.

TAG_CHECKED2 bool

Is the second page access tag checked.

TRANS bool

Is this a translated access.

VADDR unsigned int

The virtual address of the access.

CORE_REGS

Changes of the core registers R0 to R14. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number, 0 to 14.

MODE enum

Bank of the register accessed.

OLD_VALUE unsigned int

The old value overwritten.

PHYS_ID enum

The physical register accessed.

VALUE unsigned int

The new value written to the register.

CORE_REGS64

Changes of the core registers X0..X30, SP_ELn. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ID enum

The register number, 0-30 for X0-X30, >=32 for SP_ELn.

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

CORE_REGS64_READ

General purpose AArch64 register has been read X0..X30, SP_ELn. Fields:

ID enum

The register number, 0-30 for X0-X30, >=32 for SP_ELn.

VALUE unsigned int

The value read from the register.

CORE_STALLED_TRANSITION

Detect when a core is prevented from progressing PChannel states. Fields:

CURRENT_STATE enum

Current power mode of core.

REASON string

The reason a transition was stalled, and what must be done to unblock it.

TARGET_STATE enum

Desired power mode of core after transition.

CORE_STORES

Processor store accesses. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ACQREL enum

Is this an acquire/release.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DATA unsigned int

The data read or written.

ELEMENT_SIZE unsigned int

Width of each element.

ENDIANNESS enum

Byte ordering of this access.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

LOCK enum

Normal, exclusive or locked access.

MECID unsigned int

Memory Encryption Context ID, if applicable.

MECID2 unsigned int

Memory Encryption Context ID of the second page, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PADDR_TAG unsigned int

If tag checked, the physical address tag for the access.

PAGE_SIZE unsigned int

Page size as $\log_2(\text{size in bytes})$.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK, 2=Exclusive Failed, 3=Unsupported.

SIZE unsigned int

Width of the access in bytes. Only required if DATA is not traced.

TAG_CHECKED bool

Is this access tag checked.

TAG_CHECKED2 bool

Is the second page access tag checked.

TRANS bool

Is this a translated access.

VADDR unsigned int

The virtual address of the access.

CP14_READ

System Coprocessor register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP14 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc1 unsigned int

opcode 1.

opc2 unsigned int

opcode 2.

CP14_WRITE

System Coprocessor register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP14 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc1 unsigned int

opcode 1.

opc2 unsigned int

opcode 2.

CP15_READ

System Coprocessor register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP15 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc1 unsigned int

opcode 1.

opc2 unsigned int

opcode 2.

CP15_READ64

System Coprocessor register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP15 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc unsigned int

opcode 1.

CP15_WRITE

System Control Coprocessor register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP15 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc1 unsigned int

opcode 1.

opc2 unsigned int

opcode 2.

CP15_WRITE64

System Coprocessor register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP15 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc unsigned int

opcode 1.

CPSR

CPSR change. (In AArch64 PSTATE represented in SPSR format). Fields:

CORE_NUM unsigned int

Core number in a multi processor.

OLD_VALUE unsigned int

The old CPSR value.

UNKNOWN unsigned int

Bits within the register that have unknown value.

VALUE unsigned int

The new CPSR value.

CRYPTO_SPEC

Every crypto instruction speculatively executed.

CorePowerStateChange

Core power-up state has changed. DISPLAY Core powered %{PU:(up|down)}. Fields:

PU bool

Whether the core powered up.

DATA_CACHE_SET_ALLOC_TAGS

Set allocation tags from DC GVA instruction. Fields:

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

NSDESC unsigned int

The security state of the access.

PADDR unsigned int

The physical address of the access.

PAS enum

The physical address space of the page.

RESPONSE enum

0=Aborted, 1=OK.

SIZE unsigned int

The size of memory in bytes.

TAG unsigned int

The value of the tag being set.

TAGGING_TYPE enum

Type of Memory tagging (Physical or Virtual).

TAG_VADDR unsigned int

The virtual address of the Virtual Tag if Virtual tagging is used; otherwise 0.

VADDR unsigned int

Virtual address used for setting allocation tags.

DATA_CACHE_ZERO

Zero and invalidate cache line from DC ZVA instruction. Fields:

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

NSDESC unsigned int

The security state of the access.

PADDR unsigned int

The physical address of the access.

PADDR_TAG unsigned int

If tag checked, the physical address tag for the access.

PAS enum

The physical address space of the page.

RESPONSE enum

0=Aborted, 1=OK.

SIZE unsigned int

The size of zero'd memory in bytes.

TAG_CHECKED bool

Is this access tag checked.

VADDR unsigned int

Virtual address of the zero'd cache line.

DEBUG_EVENT

Hardware debug support event. Fields:

EVENT enum

Description of event.

VALUE unsigned int

data value.

DEBUG_TARGET_ISET

Hardware debug target ISA. Fields:

TARGET_ISET enum

The target instruction set of the debug state.

DMI_ALLOCATE

DMI allocated. Fields:

ACCESS_DIRECTION enum

READ / WRITE.

PADDR unsigned int

The physical address of the DMI.

PTR unsigned int

DMI range pointer.

SIDE enum

DSIDE / ISIDE.

SIZE unsigned int

The size of DMI.

VADDR unsigned int

The virtual address of the DMI..

DMI_HIT

DMI hit, access used DMI (only fire for small DMI regions i.e. less than a full page). Fields:

ACCESS_DIRECTION enum

READ / WRITE.

PADDR unsigned int

The physical address of the DMI.

SIDE enum

DSIDE / ISIDE.

SIZE unsigned int

The size of the DMI.

VADDR unsigned int

The virtual address of the DMI..

DMI_REVOKE

DMI revoked. Fields:

ACCESS_DIRECTION enum

READ / WRITE.

PADDR unsigned int

The physical address of the DMI.

SIDE enum

DSIDE / ISIDE.

SIZE unsigned int

The size of DMI.

VADDR unsigned int

The virtual address of the DMI..

END_COMPILE

Compilation end. Fields:

END_OF_PAGE_COUNT unsigned int

Number of basic blocks exited due to page boundary since START_COMPILE.

FETCHFAIL_COUNT unsigned int

Number of basic blocks exited due to ifetch failure START_COMPILE.

INST_COUNT unsigned int

Number of instructions compiled since START_COMPILE.

NONSEQ_COUNT unsigned int

Number of basic blocks exited due to non-sequential instructions since START_COMPILE.

PAGE_STRADDLE_COUNT unsigned int

Number of basic blocks exited due to unaligned instructions crossing page since START_COMPILE.

ESB

Error Synchronization Barrier hit. Fields:

ASYNC bool

ESB is async.

IMPLICIT bool

ESB is implicit.

EXCEPTION

Exception taken. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ESR unsigned int

The value assigned to ESR when exception is taken.

ESR_EC unsigned int

The Value assigned to ESR.EC when exception is taken.

IS_PHYSICAL bool

Physical or Virtual exception.

IS_TRAP bool

Exception coming from a trap.

LR unsigned int

The value assigned to the link register.

PC unsigned int

The location where the exception occurred.

PREFERRED_RETURN unsigned int

The preferred return address for the exception.

TAKEN_LOCALLY bool

Whether exception is taken locally(same exception level).

TARGET_ISET enum

The instruction set of the exception handler code.

TARGET_PC unsigned int

The address the exception branches to.

VECTOR enum

The exception vector.

VECTOR_OFFSET unsigned int

The offset of an exception vector from the base address.

EXCEPTION_END

Every exception completed.

EXCEPTION_INFO

Trace human-readable reason of the current exception for data/instruction aborts, floating point exceptions and unknown exceptions. Fields:

DESCRIPTION string

Reason for current exception.

ESR_EC unsigned int

The value of ESR.EC when exception is taken.

EXCEPTION_RAISE

Every exception raised.

EXCEPTION_RETURN

Branches on leaving exception. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISET enum

The instruction set of the branch instruction.

IS_COND bool

Indicates if this is a conditional branch.

IS_HINTED bool

Whether the conditional instruction is hinted.

IS_ISB bool

Whether the instruction is ISB.

IS_LINK bool

Indicates if this is a branch with link.

PC unsigned int

The address of the branch instruction.

TARGET_ISET enum

The instruction set after the branch.

TARGET_PC unsigned int

The address the instruction branches to.

EXCEPTION_RETURN_PREBRANCH

Exception return event that must occur before PC and processor state have updated.

EXCEPTION_START

Every exception started.

FIQ_TAKEN

FIQ taken exception.

FP_EXCEPTIONS_SIGNALLED

Trace which FP exceptions were signalled as a part of the FP operations performed by an instruction. Fields:

FP_EXCEPTIONS_SIGNALLED_MASK unsigned int

Indicates which FP exceptions were signalled by an instruction. This uses the same encoding as the cumulative floating-point exception bitfields in FPSR.

FP_STATE

Floating point state. Fields:

FPSR unsigned int

FPSR register value.

MASK unsigned int

The mask indicating updated cumulative bits.

FREQ_CHANGED

Core frequency changed. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The instruction count of this core.

NEW_FREQ signed int

The new frequency of this core (expressed in Hz).

OLD_FREQ signed int

The old frequency of this core (expressed in Hz).

HLT

HLT instruction occurred. Fields:

IMM16 unsigned int

The 16-bit immediate value encoded in HLT instruction.

INST

Every instruction executed. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DEBUG_STATE bool

The instruction is executed in debug state.

DISASS string

Disassembly of instruction.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISSET enum

The current instruction set.

ITSTATE unsigned int

The current ITSTATE.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

MODE enum

The mode the core is in.

NS unsigned int

The core's non-secure bit.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

OPCODE unsigned int

The opcode of the instruction.

PADDR unsigned int

The physical address of the instruction.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the instruction.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the page.

PC unsigned int

The address of the instruction.

SECURITY_STATE enum

The core's security state.

SIZE unsigned int

The size of the instruction in bytes.

INST_END

Every instruction completed.

INST_START

Every instruction started. Fields:

DEBUG_STATE bool

The instruction is executed in debug state.

ISET enum

The current instruction set.

MODE enum

The mode the core is in.

NS enum

The current Secure State.

PC unsigned int

The address of the conditional instruction.

SECURITY_STATE enum

The current Security State.

INST_STRADDLE

Instruction straddles boundary. Fields:

DEBUG_STATE bool

The instruction is executed in debug state.

ISET enum

The current instruction set.

MODE enum

The mode the core is in.

NS enum

The current Secure State.

PC unsigned int

The address of the conditional instruction.

SECURITY_STATE enum

The current Security State.

IRQ_TAKEN

IRQ taken exception.

ISB

ISB instruction occurred. Fields:

PC unsigned int

The address of the ISB instruction.

LOAD_MULTIPLE_REGS_LIST

List of destination registers (not source) of a multiple registers load instruction. Fields:

REG string

Register to add into the list.

LOCAL_MONITOR

Local monitor activity. Fields:

PADDR unsigned int

Local Monitor Address.

State enum

State of the monitor (Open/Exclusive).

MAPPING_CREATE

A memory mapping has been created. This is an implementation detail of the model and so may change without notice. Fields:

ASID unsigned int

The ASID used during translation, or 0 if no ASID was used.

IS_VALID bool

Whether the mapping is valid or not.

PADDR unsigned int

Base physical address of region, or 0 if not known.

SIDE enum

I-side or D-side.

SIZE unsigned int

Size of the page, or 0 if not known.

VADDR unsigned int

Base virtual address of region.

VMID unsigned int

The VMID used during translation, or 0 if no VMID was used.

MAPPING_DESTROY

A memory mapping has been destroyed. This is an implementation detail of the model and so may change without notice. Fields:

ASID unsigned int

The ASID used during translation.

CAUSE enum

The cause of invalidation.

PADDR unsigned int

Base physical address of page.

SIDE enum

I-side or D-side.

SIZE unsigned int

Size of the page.

VADDR unsigned int

Base virtual address of page.

VMID unsigned int

The VMID used during translation.

MEMMAP_DEBUG_READ

Memory mapped reads to the debug registers. Fields:

ADDR unsigned int

address.

CORE_NUM unsigned int

Core number in a multi processor.

EXT bool

Whether access is from an external device (such as the DAP).

NS enum

Secure state of the access.

REG_NAME string

Name of the debug register accessed.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

MEMMAP_DEBUG_WRITE

Memory mapped writes to the debug registers. Fields:

ADDR **unsigned int**

address.

CORE_NUM **unsigned int**

Core number in a multi processor.

EXT **bool**

Whether access is from an external device (such as the DAP).

NS **enum**

Secure state of the access.

REG_NAME **string**

Name of the debug register accessed.

UNDEF **bool**

The register accessed is undefined.

VALUE **unsigned int**

The new value written.

MEMTAG_LOADS

Allocation tag memory read; when attributes and system configuration allow tag accesses.
Fields:

ACCESS_TYPE **enum**

The type of instruction performing the access.

ATTR **unsigned int**

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 **unsigned int**

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

NSDESC **unsigned int**

The physical address non-secure bit.

NSDESC2 **unsigned int**

The second page physical address non-secure bit.

NUM_GRANULES **unsigned int**

Number of tags to be Loaded/stored.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK.

TAGGING_TYPE enum

Type of Memory tagging (Physical or Virtual).

TAGS unsigned int

The tags read or written. Tags for subsequent granules would be found at subsequent bytes in this field (One byte for each tag).

TAG_VADDR unsigned int

The virtual address of the Virtual Tag if Virtual tagging is used; otherwise 0.

VADDR unsigned int

The virtual address of the access.

MEMTAG_LOAD_INST

Allocation tag load instruction; regardless of tags being actually accessed. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

NUM_GRANULES unsigned int

Number of tags to be Loaded/stored.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK.

TAGGING_TYPE enum

Type of Memory tagging (Physical or Virtual).

TAGS unsigned int

The tags read or written. Tags for subsequent granules would be found at subsequent bytes in this field (One byte for each tag).

TAG_VADDR unsigned int

The virtual address of the Virtual Tag if Virtual tagging is used; otherwise 0.

VADDR unsigned int

The virtual address of the access.

MEMTAG_STORES

Allocation tag memory write; when attributes and system configuration allow tag accesses.

Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

NUM_GRANULES unsigned int

Number of tags to be Loaded/stored.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK.

TAGGING_TYPE enum

Type of Memory tagging (Physical or Virtual).

TAGS unsigned int

The tags read or written. Tags for subsequent granules would be found at subsequent bytes in this field (One byte for each tag).

TAG_VADDR unsigned int

The virtual address of the Virtual Tag if Virtual tagging is used; otherwise 0.

VADDR unsigned int

The virtual address of the access.

MEMTAG_STORE_INST

Allocation tag store instruction; regardless of tags being actually accessed. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

NUM_GRANULES unsigned int

Number of tags to be Loaded/stored.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK.

TAGGING_TYPE enum

Type of Memory tagging (Physical or Virtual).

TAGS unsigned int

The tags read or written. Tags for subsequent granules would be found at subsequent bytes in this field (One byte for each tag).

TAG_VADDR unsigned int

The virtual address of the Virtual Tag if Virtual tagging is used; otherwise 0.

VADDR unsigned int

The virtual address of the access.

MMU_D128_TTB_READ

This event is triggered by reads caused by a D128 translation table walk. Fields:

ABORTED bool

Was the walk successful.

DATA_HIGH unsigned int

The data upper bits read.

DATA_LOW unsigned int

The data lower bits read.

IPA unsigned int

For stage 1, the IPA of the read.

LEVEL unsigned int

Translation table level.

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

OUTADDR unsigned int

The output physical address.

PADDR unsigned int

The physical address of the read.

PASmem enum

Physical Address Space of the accessed memory.

PASreq enum

Physical Address Space of the requested lookup.

REGIME_EL enum

Entry matches in this translation regime.

REQUESTER enum

The requester of this table walk.

SIDE enum

Inst / Data.

STAGE unsigned int

Translation stage.

MMU_D128_TTB_WRITE

This event is triggered by writes caused by a D128 translation table walk. Fields:

ABORTED bool

Was the walk successful.

DATA_HIGH unsigned int

The data upper bits written.

DATA_LOW unsigned int

The data lower bits written.

IPA unsigned int

For stage 1, the IPA of the write.

IS_HACDBS_ACCESS bool

If the access was due to HACDBS_CLEAN.

LEVEL unsigned int

Translation table level.

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

OUTADDR unsigned int

The output physical address.

PADDR unsigned int

The physical address of the write.

PASmem enum

Physical Address Space of the accessed memory.

PASreq enum

Physical Address Space of the requested lookup.

REGIME_EL enum

Entry matches in this translation regime.

SIDE enum

Inst / Data.

STAGE unsigned int

Translation stage.

MMU_TRANS

Address translation information. Fields:

ASID unsigned int

Address space identifier.

CORE_NUM unsigned int

Core number in a multi processor.

Hyp bool

Entry matches in Hyp state only.

INNERCACHE_RA bool

Is the inner cache allocate on read.

INNERCACHE_TRANSIENT bool

Is the inner write-through transient.

INNERCACHE_TYPE enum

Inner Caching scheme (NC/MB/WA).

INNERCACHE_WA bool

Is the inner cache allocate on write.

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MEMTYPE enum

Memory type.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC enum

Is secure side supposed to access secure or nonsecure memory.

OUTERCACHE_RA bool

Is the outer cache allocate on read.

OUTERCACHE_TRANSIENT bool

Is the outer write-through transient.

OUTERCACHE_TYPE enum

Outer Caching scheme (NC/MB/WA).

OUTERCACHE_WA bool

Is the outer cache allocate on write.

PADDR unsigned int

Physical address of the access.

PAGESIZE unsigned int

Page size as $\log_2(\text{size})$.

PAS enum

PAS supposed to be accessed is secure or nonsecure memory or Root/Realm if RME is implemented.

SH enum

Shareability.

SIDE enum

Inst / Data.

STAGE1_PERM unsigned int

Stage 1 permissions, [5:3] Privileged access (XWR) [2:0] User access (XWR).

STAGE2_PERM unsigned int

Stage 2 permission mask [5:3] Privileged access (XWR) [2:0] User access (XWR).

VADDR unsigned int

Virtual address of the access.

VMID unsigned int

Virtual machine identifier.

XS bool

XS Attribute.

nG enum

Flag indicating whether ASID will be matched.

MMU_TTB_READ

This event is triggered by reads caused by a translation table walk. Fields:

ABORTED bool

Was the walk successful.

DATA unsigned int

The data read.

DESC_KIND enum

The encoding scheme of the descriptors used for this walk.

IPA unsigned int

For stage 1, the IPA of the read.

LEVEL unsigned int

Translation table level.

LPAE bool

Is this for an LPAE translation (*DEPRECATED*, use DESC_KIND).

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSmem bool

Non secure memory access.

NSreq bool

Non secure request.

OUTADDR **unsigned int**

The output physical address.

PADDR **unsigned int**

The physical address of the read.

PASmem **enum**

Physical Address Space of the accessed memory.

PASreq **enum**

Physical Address Space of the requested lookup.

REGIME_EL **enum**

Entry matches in this translation regime.

REQUESTER **enum**

The requester of this table walk.

SIDE **enum**

Inst / Data.

STAGE **unsigned int**

Translation stage.

MMU_TTB_WRITE

This event is triggered by writes caused by a translation table walk. Fields:

ABORTED **bool**

Was the walk successful.

DATA **unsigned int**

The data written.

DESC_KIND **enum**

The encoding scheme of the descriptors used for this walk.

IPA **unsigned int**

For stage 1, the IPA of the write.

IS_HACDBS_ACCESS **bool**

If the access was due to HACDBS_CLEAN.

LEVEL **unsigned int**

Translation table level.

LPAAE **bool**

Is this for an LPAE translation (*DEPRECATED*, use DESC_KIND).

MECID **unsigned int**

Memory Encryption Context(MEC) ID, if applicable.

MPAM_PARTID **unsigned int**

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSmem bool

Non secure memory access.

NSreq bool

Non secure request.

OUTADDR unsigned int

The output physical address.

PADDR unsigned int

The physical address of the write.

PASmem enum

Physical Address Space of the accessed memory.

PASreq enum

Physical Address Space of the requested lookup.

REGIME_EL enum

Entry matches in this translation regime.

SIDE enum

Inst / Data.

STAGE unsigned int

Translation stage.

MODE_CHANGE

Mode change. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

MODE enum

The new mode.

NON_SECURE enum

The core's new non-secure bit.

OLD_MODE enum

The old mode.

SECURITY_STATE enum

The new security state.

NOT_TAG_CHECKED_INST

MTE Tag not checked for current instruction.

PERIODIC

Called for every quantum. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The instruction count of this CPU.

PC unsigned int

The address of the next instruction to be executed on this CPU.

PMU_COUNTER_OVERFLOW

PMU counter overflow. Fields:

COUNTER_GROUP enum

Counter group.

COUNTER_RANGE enum

Counter range.

EVENT_ID unsigned int

Event Identifier, field is valid only for Event counter group.

INDEX unsigned int

Counter index (as selected by PMSELR), field is valid only for Event/Cycle counter groups.

INTERRUPT bool

Is interrupt enabled on overflow for this counter.

POINTER_AUTH

Called when execution hits a pointer authentication check. Fields:

AUTH_STATE bool

Authentication result.

KEY_USED enum

Pointer authentication key.

POTINDEX_UPDATE

Value of POT index for current pc changed. Fields:

NEW_POTINDEX unsigned int

New POTIndex.

OLD_POTINDEX unsigned int

Old POTIndex.

PREFETCH_MEMORY64

Prefetch from PRFM or PRFUM instructions. Fields:

PRFOP enum

Prefetch hint.

VADDR unsigned int

Virtual address of the location that should be prefetched.

PRELOAD64_REQUEST

Preload requests from PRFM or PRFUM instructions. Fields:

PRFOP enum

Prefetch hint.

VADDR unsigned int

Virtual address of the location that should be prefetched.

PRELOAD_DATA

Data preload from PLD instruction. Fields:

VADDR unsigned int

Virtual address of the data that should be preloaded.

PRELOAD_INST

Instruction preload from PLI instruction. Fields:

VADDR unsigned int

Virtual address of the instruction that should be preloaded.

PRE_CORE_LOAD

Trace just before a core load. Fields:

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

PRE_CORE_STORE

Trace just before a core store. Fields:

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

PRE_TTB_READ

This event is triggered before reads caused by a translation table walk. Fields:

ADDR unsigned int

The physical address of the read.

PSTATE

PSTATE fields update. Fields:

ALLINT unsigned int

All interrupt mask bit.

BTYPE unsigned int

Branch target identification bit.

CORE_NUM unsigned int

Core number in a multi processor.

DAIF unsigned int

Exception mask bits.

DIT unsigned int

Data Independent Timing bit.

EL unsigned int

Current Exception level field.

EXLOCK unsigned int

Exception return state lock bit.

IL unsigned int

Illegal Execution state bit.

NZCV unsigned int

Condition flags.

PACM unsigned int

PACM bit.

PAN unsigned int

Privileged Access Never state bit.

PM unsigned int

PMU exception mask bit.

PPEND unsigned int

PMU exception pending bit.

SM unsigned int

StreamingMode status bit.

SP unsigned int

Stack pointer register selection bit.

SS unsigned int

Software Step bit.

SSBS unsigned int

Speculative Store Bypass Safe bit.

TCO unsigned int

Tag Check Override bit.

TOUCHED unsigned int

Mask of the fields that have been touched since this trace has last been emitted. Note that touched does not necessarily mean the value changed. The index of the field is the bit index in the mask plus the index of this field plus one.

UAO unsigned int

User Access Override bit.

UINJ unsigned int

Undefined Instruction exceptions bit.

ZA unsigned int

ZA storage and ZT0 register enable bit.

nRW unsigned int

Current Execution state bit.

PSTATE_PACTIVE_OPMODE_TRANSITION

Trace pactive and Opmode mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Operating mode after the transition.

OLD_STATE enum

Operating mode before the transition.

PSTATE_PACTIVE_OPMODE_TRANSITION_DENIED

Trace the denied pactive and Opmode mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Operating mode after the transition.

OLD_STATE enum

Operating mode before the transition.

REASON string

Reason for the transition being denied.

PSTATE_PACTIVE_POWER_TRANSITION

Trace pactive and power mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Power mode after the transition.

OLD_STATE enum

Power mode before the transition.

PSTATE_PACTIVE_POWER_TRANSITION_DENIED

Trace the denied pactive and power mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Power mode after the transition.

OLD_STATE enum

Power mode before the transition.

REASON string

Reason for the transition being denied.

PSTATE_PACTIVE_THREAD_OPMODE_TRANSITION

Trace pactive and Opmode mode transitions for threads on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Operating mode after the transition.

OLD_STATE enum

Operating mode before the transition.

QUANTUM_START

Called at the start of every quantum.

RANGE_PRELOAD_REQUEST

Preload requests from RPRFM. Fields:

COUNT unsigned int

Number of blocks of data to access.

LENGTH unsigned int

Number of contiguous bytes to be accessed in each block.

OP unsigned int

Range prefetch operation .

REUSE unsigned int

Maximum number of bytes to be accessed before next RPRFM .

STRIDE signed int

Stride to advance after accessing LENGTH bytes.

VADDR unsigned int

Base virtual address of range prefetch request.

RUN_STATE

Run state transition. Fields:

INST_COUNT **unsigned int**

Ticks count at point of transition.

NEW **enum**

New run state.

OLD **enum**

Old run state.

SEMIHOSTING_CALL

Call of a semihost function occurred. Fields:

PC **unsigned int**

The program counter after the semihosting call.

SEMIHOSTING_PRECALL

About to call semihost. Fields:

EL2 **bool**

Use EL2 translation regime.

NS **bool**

Is Non-Secure.

REG_WIDTH **unsigned int**

The current register width in bytes.

SIGNAL

External signal state change. Fields:

SIGNAL **enum**

Signal that changed.

STATE **bool**

Signal asserted state.

SOFTWARE_STEP

Return the Debug Software Step state. Fields:

STATE **enum**

Software Step state.

SPSR

SPSR change. Fields:

CORE_NUM **unsigned int**

Core number in a multi processor.

MODE **enum**

Which of the banked SPSR registers is written.

OLD_VALUE unsigned int

The old SPSR value.

VALUE unsigned int

The new SPSR value.

START_COMPILE

Compilation started. Fields:

VADDR unsigned int

Instruction where compilation begins.

STG_MULTIPLE_MICRO_OPS_ADDRESS_RANGE

Address range for MTE stores with multiple uops. Fields:

ADDR_END unsigned int

Final address.

ADDR_START unsigned int

Starting address.

STORE_MULTIPLE_REGS_ADDRESS_RANGE

Address range for stores with multiple registers. Fields:

ADDR_END unsigned int

Final address.

ADDR_START unsigned int

Starting address.

SUBOPTIMAL_LDST_RETIRED

suboptimal load/store. Fields:

ACCESS_SIZE unsigned int

Log2 of the access width in bytes used for alignment checking (1: half-word, 2: word ...).

OPTIMAL_ALIGN_SELECTOR unsigned int

Selector to decide the optimal alignment size (1: access width, 2: 4-bytes, 3: 8-bytes, ..., 12: 4-kbytes).

VADDR unsigned int

The virtual address of the access.

SVE_INST_SPEC

SVE operations speculatively executed.

SVE_LD_RETIRED

SVE load instructions architecturally executed.

SVE_LOADS

SVE load accesses. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ACQREL enum

Is this an acquire/release.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DATA unsigned int

The data read or written.

ELEMENT_SIZE unsigned int

Width of each element.

ENDIANNESS enum

Byte ordering of this access.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

LOCK enum

Normal, exclusive or locked access.

MECID unsigned int

Memory Encryption Context ID, if applicable.

MECID2 unsigned int

Memory Encryption Context ID of the second page, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PADDR_TAG unsigned int

If tag checked, the physical address tag for the access.

PAGE_SIZE unsigned int

Page size as $\log_2(\text{size in bytes})$.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK, 2=Exclusive Failed, 3=Unsupported.

SIZE unsigned int

Width of the access in bytes. Only required if DATA is not traced.

TAG_CHECKED bool

Is this access tag checked.

TAG_CHECKED2 bool

Is the second page access tag checked.

TRANS bool

Is this a translated access.

VADDR unsigned int

The virtual address of the access.

SVE_MEM_PREFETCH

Scalable Vector Extension memory prefetch. Fields:

CACHE_LEVEL enum

The preferred cache level.

HINT enum

What the memory is expected to be used for.

SIZE unsigned int

The prefetch size in bytes.

TEMPORAL bool

Whether the data is temporal.

VADDR unsigned int

The virtual address of the prefetch.

SVE_REG_READ

Scalable Vector Extension register read. Fields:

ID enum

The register identifier.

SM bool

Whether the PE is in Streaming Mode.

VALUE unsigned int

The read data.

SVE_REG_UPDATE

Scalable Vector Extension register update. Fields:

ID enum

The register identifier.

SM bool

Whether the PE is in Streaming Mode.

VALUE unsigned int

The written data.

SVE_REG_WRITE

Scalable Vector Extension register write. Fields:

ID enum

The register identifier.

SM bool

Whether the PE is in Streaming Mode.

VALUE unsigned int

The written data.

SVE_STORES

SVE store accesses. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ACQREL enum

Is this an acquire/release.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DATA unsigned int

The data read or written.

ELEMENT_SIZE unsigned int

Width of each element.

ENDIANNESS enum

Byte ordering of this access.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

LOCK enum

Normal, exclusive or locked access.

MECID unsigned int

Memory Encryption Context ID, if applicable.

MECID2 unsigned int

Memory Encryption Context ID of the second page, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PADDR_TAG unsigned int

If tag checked, the physical address tag for the access.

PAGE_SIZE unsigned int

Page size as $\log_2(\text{size in bytes})$.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK, 2=Exclusive Failed, 3=Unsupported.

SIZE unsigned int

Width of the access in bytes. Only required if DATA is not traced.

TAG_CHECKED bool

Is this access tag checked.

TAG_CHECKED2 bool

Is the second page access tag checked.

TRANS bool

Is this a translated access.

VADDR unsigned int

The virtual address of the access.

SVE_ST_RETIRED

SVE store instructions architecturally executed.

SVE_UNALIGNED_LDST_RETIRED

SVE load or store instructions architecturally executed that would cause alignment fault if A bit is set.

SVE_Z_PRED_MEM_ACCESS_INFO

Information about SVE/SME predicated accesses to Z registers (non Gather/Scatter). Fields:

MEM_ELEM_SIZE unsigned int

Width of elements accessed in memory (in bytes).

RnW bool

Whether this is a Read not a Write.

VA unsigned int

Virtual base address computed for accesses.

SYNC

Called for every synchronization, that is every quantum break. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The instruction count of this CPU.

LOCAL_QUANTUM unsigned int

The local quantum of this CPU.

LOCAL_TIME unsigned int

The local time of this CPU.

SYNC_REASON enum

Reason for sync.

SYSCALL

System call instruction executed. Fields:

IMM unsigned int

Immediate value of the system call instruction.

TYPE enum

System call type.

VADDR unsigned int

Instruction that caused the system call.

SYSREG_READ128

System Coprocessor 128 bit register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

REGNUM enum

Internal register number.

REGNUM_PHYS enum

Internal register number physically accessed (e.g. in presense of VHE redirection).

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_READ64

System Coprocessor register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REGNUM enum

Internal register number.

REGNUM_OPERAND enum

Register number corresponding to the instruction operand.

REGNUM_PHYS enum

Internal register number physically accessed (e.g. in presense of VHE redirection).

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_UPDATE128

Triggers when the system updates a register. Fields:

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG enum

Register number.

UNKNOWN unsigned int

Bits of the register which became unknown.

VALUE unsigned int

Value written to the register.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_UPDATE32

Triggers when the system updates a register. Fields:

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG enum

Register number.

UNKNOWN unsigned int

Bits of the register which became unknown.

VALUE unsigned int

Value written to the register.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_UPDATE64

Triggers when the system updates a register. Fields:

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG enum

Register number.

UNKNOWN unsigned int

Bits of the register which became unknown.

VALUE unsigned int

Value written to the register.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_WRITE128

System Coprocessor 128 bit register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

REGNUM enum

Internal register number.

REGNUM_PHYS enum

Internal register number physically accessed (e.g. in presense of VHE redirection).

UNDEF bool

The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int
opcode 2.

SYSREG_WRITE64

System Coprocessor register write. Fields:

CORE_NUM unsigned int
Core number in a multi processor.

CRm unsigned int
CRm.

CRn unsigned int
CRn.

NS bool
Secure or nonsecure banked register is accessed.

REGNUM enum
Internal register number.

REGNUM_OPERAND enum
Register number corresponding to the instruction operand.

REGNUM_PHYS enum
Internal register number physically accessed (e.g. in presense of VHE redirection).

UNDEF bool
The register accessed is undefined.

UPDATED_VALUE unsigned int
Updated value of the register now it has been written.

VALUE unsigned int
The new value written.

opc unsigned int
opcode 1.

opc0 unsigned int
opcode 0.

opc2 unsigned int
opcode 2.

TRBU_BUFFER_WRAP

DISPLAY Trace buffer current write pointer wrapped.

TRBU_TRIGGER_EVENT

DISPLAY Trace buffer Trigger Event.

TRBU_WRITE

DISPLAY PETU trace data has been written to the TRBU memory buffer at
%{BUFFER_ADDRESS} with size %{SIZE} and data %{DATA}. Fields:

BUFFER_ADDRESS unsigned int

Virtual Address of start of buffer write.

DATA unsigned int

Data written to the buffer.

SIZE unsigned int

Size of sample written to buffer, in bytes.

UNALIGNED_LDST_RETIRED

Processor unaligned load/store. Fields:

ACCESS_SIZE unsigned int

Log2 of the access width in bytes used for alignment checking (1=>half-word, 2->word ...).

NUMBER_OF_BEATS unsigned int

Number of accesses (beats) in this burst (Total size in bytes is ACCESS_SIZE * NUMBER_OF_BEATS).

VADDR unsigned int

The virtual address of the access.

VFP_D_REGS

VFP/NEON D 64 bit register write. Fields:

ALIASING enum

Type of register aliasing used.

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a write to the corresponding bit in the register. For example, 0x1 would indicate a write to VALUE[0].

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

VFP_D_REGS_READ

VFP/NEON D 64 bit register read. Fields:

ID unsigned int

The register number.

VALUE unsigned int

The current value of the register.

VFP_Q_REGS

VFP/NEON Q 128 bit register write. Fields:

ALIASING enum

Type of register aliasing used.

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a write to the corresponding byte in the register. For example, 0x1 would indicate a write to VALUE[7:0].

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

VFP_Q_REGS_READ

VFP/NEON Q 128 bit register read. Fields:

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a read to the corresponding byte in the register. For example, 0x1 would indicate a read to VALUE[7:0].

VALUE unsigned int

The current value of the register.

VFP_SYS_REGS

Writes to the VFP/NEON units system registers. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ID enum

Which VFP system register is written.

MASK unsigned int

The mask indicating updated cumulative bits.

OLD_VALUE unsigned int

The register's old value overwritten.

VALUE unsigned int

The new value written to the VFP system register.

VFP_S_REGS

VFP/NEON S 32 bit register write. Fields:

ALIASING enum

Type of register aliasing used.

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a write to the corresponding bit in the register. For example, 0x1 would indicate a write to VALUE[0].

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

VFP_S_REGS_READ

VFP/NEON S 32 bit register read. Fields:

ID unsigned int

The register number.

VALUE unsigned int

The current value of the register.

VIRTUAL_TAG_ALLOCATED

Virtual tag information when allocated. Fields:

TAG_VADDR unsigned int

Virtual tag virtual address.

VADDR unsigned int

Data Virtual address.

VIRTUAL_TAG unsigned int

Virtual tag value.

WAYPOINT

Signals end of basic block execution. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ISSET enum

Origin instruction set.

IS_COND bool

Indicates if this is a conditional waypoint.

PC unsigned int

Origin address (or 0 if unavailable).

TAKEN bool

Indicates if this waypoint was taken.

TARGET unsigned int

Destination address.

TARGET_ISET enum

Destination instruction set.

WFE_END

WFE ended. Fields:

INST_COUNT unsigned int

Ticks count when leaving WFE.

WFE_EVENT_REGISTER

WFE event register status: set/clear, reason. Fields:

INST_COUNT unsigned int

Ticks count.

REASON enum

Reason for set/clear. Only REASON==0 (Cleared by WFE) clears the bit, all other reasons set it.

WFE_IGNORED

WFE ignored. Fields:

EVENT bool

This WFE was ignored because the event register was set.

INST_COUNT unsigned int

Ticks count when ignoring WFE.

TRAPPED bool

This WFE was trapped.

WFE_START

WFE entered. Fields:

INST_COUNT unsigned int

Ticks count when entering WFE.

WFI_END

WFI ended. Fields:

INST_COUNT unsigned int

Ticks count when leaving WFI.

WFI_IGNORED

WFI ignored. Fields:

DISABLED bool

This WFI was ignored because WFI is disabled.

INST_COUNT unsigned int

Ticks count when ignoring WFI.

REASON enum

specifies reason why WFI trace was ignored.

TRAPPED bool

This WFI was trapped.

WFI_START

WFI entered. Fields:

INST_COUNT unsigned int

Ticks count when entering WFI.

WFI_WAKEUP

WFI wakeup. Fields:

INST_COUNT unsigned int

Ticks count when WFI wakeup occurred.

REASON enum

Reason for wakeup.

2.14 ARM_Cortex-A34

This section describes the trace sources.

AA64_ASE_SVE_REGS

Modification of AArch64 Advanced SIMD or SVE Vector Registers (via SIMD or Scalar operations), SVE Predicate Registers and ZTO. Fields:

ID enum

The register identifier.

MASK unsigned int

Bitmask of the register to signal the modified bits in the VALUE field.

SM bool

Whether the PE is in Streaming Mode.

VALUE unsigned int

Value of AArch64 vector register.

ASYNC_MEMORY_FAULT

Context ID Register write. Fields:

FAULT unsigned int

Fault status present in ESR.ISS encoding for Serror in ARCH64 and present in DFSR encoding for external abort format in ARCH32.

PADDR unsigned int

Physical Address (or 0 if unavailable).

VADDR signed int

Virtual Address (or 0 if unavailable).

ATOMIC_END_ACCESS

Bus trace access for atomics. Fields:

ACCESS_FAIL bool

Memory access failed.

ADDR unsigned int

The virtual address of the access.

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

COMPARE_VALUE unsigned int

Compare value for CAS.

LOAD_VALUE unsigned int

Loaded values.

NSDESC unsigned int

The security state of the access.

OPERAND_VALUE unsigned int

Operation's operand.

OPERATION enum

Operation type.

PADDR unsigned int

The physical address of the access.

PRIV bool

Is this a privileged access?.

ATOMIC_START_ACCESS

Bus trace access for atomics. Fields:

ADDR unsigned int

The virtual address of the access.

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

COMPARE_VALUE unsigned int

Compare value for CAS.

NSDESC unsigned int

The security state of the access.

OPERAND_VALUE unsigned int

Operation's operand.

OPERATION enum

Operation type.

PADDR unsigned int

The physical address of the access.

PRIV bool

Is this a privileged access?.

ArchMsg.Error.exit_code

DISPLAY exit code: %{EXITCODE} component kind: %{KIND} reason: %{REASON}. Fields:

EXITCODE signed int

Exit code to be returned.

KIND string

Component kind that invoked the exit code trace.

REASON string

Reason why the exit code trace was invoked (optional).

ArchMsg.Warning.IgnoredWarmResetRequest

Event fires when software attempts to request warm reset through write to deprecated DBGPRCR.CWRR.

ArchMsg.Warning.ReportedRevisionRegisterRead

DISPLAY Reported revision and patch do not match that of the cpu. Fields:

ORIGINAL_VALUE unsigned int

The value the register would have based on the modelled revision.

REGISTER string

The name of the register that has ben read.

REPORTED_VALUE unsigned int

The value the register has been overridden to.

ArchMsg.Warning.Unpredictably Indexed PM Event Register

TAGS **UNPREDICTABLE** DISPLAY Event %{IsDirect:(Selector value|Register)} %{SEL} is >= %{N}.

Fields:

IsDirect unsigned int

Direct Access rather than indirect.

N unsigned int

Number of Accessible Registers.

SEL unsigned int

Selector.

ArchMsg.Warning.cache_contents_unknown

A cache has been activated without being invalidated and may contain **UNKNOWN** content. DISPLAY %{SIDE:(D|I)}-Cache was enabled in %{REGIME:(SECURE|NON_SECURE|HYP)} regime but was not invalidated since power-on: cache lines could contain **UNKNOWN** content %{SIDE:(Note: D-side caches must invalidate SECURE lines even if the cache is only used by NON_SECURE code in order to ensure that unexpected dirty lines tagged SECURE are not naturally evicted)}. Fields:

REGIME unsigned int

Current translation regime.

SIDE unsigned int

0 d-side cache or unified cache, 1 i-side.

ArchMsg.Warning.dcimvac_matches_watchpoint

TAGS **UNPREDICTABLE** DISPLAY Watchpoints matching an AArch32 DCIMVA are implementation defined.

ArchMsg.Warning.decode_fieldmismatch

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with different fields %{FIELD1} and %{FIELD2}. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_frachitsnegative

TAGS **UNPREDICTABLE** DISPLAY VCVT Instruction is unpredictable with a negative number of fraction bits.

ArchMsg.Warning.decode_initblock

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable in an IT block.

ArchMsg.Warning.decode_initblocknotlast

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable in an IT block when not the last.

ArchMsg.Warning.decode_invalid_condition

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable when the condition is not AL.

ArchMsg.Warning.decode_invalidfieldcombination

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with the values in %{FIELD1} and %{FIELD2}. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_invalidvalue

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with the value in %{FIELD}. Fields:

FIELD enum

Field name.

ArchMsg.Warning.decode_mem_hint_unallocated

TAGS **UNPREDICTABLE** DISPLAY There is no memory hint allocated to this bit pattern.

ArchMsg.Warning.decode_registermatch

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with %{FIELD1} and %{FIELD2} as the same register. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_registermismatch

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with %{FIELD1} and %{FIELD2} as different registers. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_registeroutofrange

TAGS **UNPREDICTABLE** DISPLAY Floating-point instruction is unpredictable indexing beyond the end of the register bank.

ArchMsg.Warning.decode_sbzsbo

TAGS **UNPREDICTABLE** DISPLAY Reserved bits in the instruction are not canonical.

ArchMsg.Warning.decode_transactiontoobig

TAGS UNPREDICTABLE DISPLAY Attempt to perform a memory transaction of more than 128 bytes.

ArchMsg.Warning.decode_unpred_other

TAGS UNPREDICTABLE DISPLAY Instruction is unpredictable.

ArchMsg.Warning.decode_unpreduseofpc

TAGS UNPREDICTABLE DISPLAY Instruction is unpredictable when %{FIELD} is PC. Fields:

FIELD enum

Field name.

ArchMsg.Warning.decode_unpreduseofr13

TAGS UNPREDICTABLE DISPLAY Instruction is unpredictable when %{FIELD} is R13. Fields:

FIELD enum

Field name.

ArchMsg.Warning.decode_unpreduseofr13inreglist

TAGS UNPREDICTABLE DISPLAY Instruction is unpredictable with R13 in register-list.

ArchMsg.Warning.decode_useofd16orhigher

TAGS UNPREDICTABLE DISPLAY Attempt to use register D16 or higher where it is not allowed.

ArchMsg.Warning.decode_writebackandbaseinlist

TAGS UNPREDICTABLE DISPLAY Instruction is unpredictable with write back when the base register is in the transfer list.

ArchMsg.Warning.decode_zeroregistersinlist

TAGS UNPREDICTABLE DISPLAY Instruction is unpredictable when no registers are to be transferred.

ArchMsg.Warning.recursive_branch

An instruction is performing a branch that targets the same instruction. If this was intended then a WFI might be more effective DISPLAY recursive branch detected at PC=%{PC}. PRIMARY KEY PC. Fields:

PC unsigned int

Address of instruction causing the branch.

ArchMsg.Warning.recursive_exception

An instruction has generated an exception that targets the same instruction. DISPLAY recursive %{TYPE} exception detected at PC=%{PC} (CPSR=%{CPSR}). PRIMARY KEY PC. Fields:

CPSR unsigned int

Processor state.

PC unsigned int

Address of instruction causing the exception.

TYPE enum

Type of exception.

ArchMsg.Warning.reentrant_vector_catch

TAGS **UNPREDICTABLE** DISPLAY Vector catch debug event on Prefetch or Data abort vector %{ADDR}. Fields:

ADDR unsigned int

vector address.

ArchMsg.Warning.reserved_it_state

TAGS **UNPREDICTABLE** DISPLAY Execution with reserved IT state %{ITSTATE}. Fields:

ITSTATE unsigned int

reserved IT state.

ArchMsg.Warning.tlb_contents_unknown

Translation has been enabled without invalidating the TLB. DISPLAY Translation was enabled in the %{PAS:(Secure|Non-secure|Root|Realm)} EL%{EL:d} translation regime (vmid=%{VMID:x}) without invalidating %{INVALIDITY:(|d-side |i-side |)}entries since power-on: %{INVALIDITY:(|D|I|)}TLB could contain **UNKNOWN** content. Fields:

EL unsigned int

Exception level of translation regime.

INVALIDITY unsigned int

Which TLB may be invalid (if TLBs are separate).

NON_SECURE bool

Whether the translation regime is non-secure.

PAS unsigned int

Physical address space of translation regime.

VMID unsigned int

Current VMID, if applicable.

ArchMsg.Warning.unknown_DLR_DSPSR

Exiting debug state with unknown DLR or DSPSR.

ArchMsg.Warning.unknown_ELR_SPSR

Returning from debug exception with unknown ELR or SPSR.

ArchMsg.Warning.unpred_FAR_on_LRCPC3_non_atomic_mem_access

TAGS **UNPREDICTABLE** Unpredictable FAR/FSTATUS on LRCPC3 pair instructions resulting in two separate memory accessesDISPLAY Unpredictable FAR on LRCPC3 non atomic memory access at %{ADDR1} with FSTATUS %{FSTATUS1} or %{ADDR2} with FSTATUS %{FSTATUS2}. Fields:

ADDR1 unsigned int

Possible FAR address 1.

ADDR2 unsigned int

Possible FAR address 2.

FSTATUS1 unsigned int

Possible fault status value 1.

FSTATUS2 unsigned int

Possible fault status value 2.

ArchMsg.Warning.unpred_async_tag_check_fail_on_sve_speculative_load_do_update_ffr

Unpredictable set to FALSE of the FFR element associated to SVE speculative load with an async tag check fail TAGS **UNPREDICTABLE**.

ArchMsg.Warning.unpredictable_a32_breakpoint

TAGS **UNPREDICTABLE** DISPLAY Unpredictable breakpoint on a A32 instruction at %{ADDR1} Offset %{OFFSET} BAS %{BAS} for a %{IS_ADDRESS_MISMATCH_BREAKPOINT:(Mismatch|Match)} Breakpoint number %{N:d} Breakpoint address %{ADDR2}. Fields:

ADDR1 unsigned int

Address.

ADDR2 unsigned int

Breakpoint address.

BAS unsigned int

BAS field.

IS_ADDRESS_MISMATCH_BREAKPOINT bool

Is a Mismatch breakpoint, not a Match breakpoint.

N unsigned int

Breakpoint number.

OFFSET unsigned int

Always has a value of zero.

ArchMsg.Warning.unpredictable_banked_register_access

TAGS **UNPREDICTABLE** DISPLAY Access to banked register (R=%{R:d} SYSm=%{SYSM:2x}) is unpredictable. Fields:

R unsigned int

R bit.

SYSM unsigned int

SYSm.

ArchMsg.Warning.unpredictable_ccfail_undef

TAGS **UNPREDICTABLE** DISPLAY An instruction %{OPCODE} at VA=%{VADDR} failed its condition codes check but would otherwise have caused a non-data-dependent trap or undefined exception. Behaviour is implementation-defined choice of exception or nop. Fields:

OPCODE unsigned int

opcode of the instruction.

VADDR unsigned int

Virtual address of the instruction being executed.

ArchMsg.Warning.unpredictable_ici_bits

TAGS **UNPREDICTABLE** DISPLAY ICI bits do not apply or are architecturally unpredictable for instruction.

ArchMsg.Warning.unpredictable_t32_breakpoint

TAGS **UNPREDICTABLE** DISPLAY Unpredictable breakpoint on a T32 instruction at %{ADDR1} Offset %{OFFSET} BAS %{BAS} for a %{IS_ADDRESS_MISMATCH_BREAKPOINT:(Mismatch|Match)} Breakpoint number %{N:d} Breakpoint address %{ADDR2}. Fields:

ADDR1 unsigned int

Address.

ADDR2 unsigned int

Breakpoint address.

BAS unsigned int

BAS field.

IS_ADDRESS_MISMATCH_BREAKPOINT bool

Is a Mismatch breakpoint, not a Match breakpoint.

N unsigned int

Breakpoint number.

OFFSET unsigned int

Indicates which halfword of a 32-bit Thumb instruction the breakpoint is set on. 0 for the first halfword or if the instruction is 16 bits wide, or 1 for the second halfword.

ArchMsg.Warning.unpredictable_t32_vector_catch

TAGS **UNPREDICTABLE** DISPLAY Unpredictable Vector catch exception on a T32 instruction at %{ADDR} Offset %{OFFSET}. Fields:

ADDR unsigned int

Address.

OFFSET unsigned int

Indicates which halfword of a 32-bit Thumb instruction the vector catch exception is set on. 0 for the first halfword or if the instruction is 16 bits wide, or 1 for the second halfword.

ArchMsg.Warning.unpredictable_watchpoint_far

TAGS **UNPREDICTABLE** DISPLAY The value for FAR/EDWAR for hitting this watchpoint may be anywhere in the range %{LOWER_BOUND} to %{UPPER_BOUND}. Fields:

LOWER_BOUND unsigned int

the lowest address accessed by the instruction that triggered the watchpoint.

UPPER_BOUND unsigned int

the highest watchpointed address accessed by that instruction.

ArchMsg.Warning.unpredictable_watchpoint_far_isscmp

DISPLAY The value for FAR/EDWAR for hitting this watchpoint may be anywhere in the range %{LOWER_BOUND} to %{UPPER_BOUND}. Fields:

LOWER_BOUND unsigned int

the lowest address accessed by the instruction that triggered the watchpoint.

UPPER_BOUND unsigned int

the highest watchpointed address accessed by that instruction.

ArchMsg.Warning.warning_AdvSIMDExpandImmUnexpectedZero

TAGS **UNPREDICTABLE** DISPLAY AdvSIMDExpandImm may treat this immediate value as **UNPREDICTABLE**.

ArchMsg.Warning.warning_ConditionalSMC

TAGS **UNPREDICTABLE** DISPLAY SMC instruction has **UNPREDICTABLE** effects when conditional and when combined with traps.

ArchMsg.Warning.warning_TLBID_out_of_range

TLBID out of range in TLBI according to the configuration of NIS, NVIS, NOS and NVOS. Fields:

NIS unsigned int

NIS.

NOS unsigned int

NOS.

NVIS unsigned int

NVIS.

NVOS unsigned int

NVOS.

TLBID unsigned int

The TLBID value used in the TLBI instruction.

ArchMsg.Warning.warning_TTL_or_TG_mismatches_current_translation_granule

TTL or TG mismatches current translation granule in TLBI VA/RVA/IPA/RIPA. Fields:

CURRENT_TG unsigned int

Current TG as indicated by TCR_ELx or VTCR_EL2.

TLBI_TG unsigned int

TG (or TTL[3:2]) used in TLBI instruction.

ArchMsg.Warning.warning_access_crosses_page_boundary

TAGS **UNPREDICTABLE** DISPLAY Access crosses page boundary.

ArchMsg.Warning.warning_access_crosses_page_boundary_has_fault

TAGS **UNPREDICTABLE** DISPLAY Access that crosses page boundary has fault(s).

ArchMsg.Warning.warning_access_crosses_page_boundary_has_fault_on_both_pages

TAGS **UNPREDICTABLE** DISPLAY Access that crosses page boundary has a fault on each page.

Fields:

ADDR1 unsigned int

FAR address on first page.

ADDR2 unsigned int

FAR address on second page.

FSTATUS1 unsigned int

Fault status on first page.

FSTATUS2 unsigned int

Fault status on second page.

ArchMsg.Warning.warning_access_crosses_page_boundary_spanning_different_memory

TAGS **UNPREDICTABLE** DISPLAY Access at address %{ADDR} crosses page from MEM_TYPE %{MEMTYPE_PAGE1} to %{MEMTYPE_PAGE2}. Fields:

ADDR unsigned int

address of access crossing page.

MEMTYPE_PAGE1 signed int

First Page's memory type (0-Normal, 1-Non-Normal).

MEMTYPE_PAGE2 signed int

Second Page's memory type (0-Normal, 1-Non-Normal).

ArchMsg.Warning.warning_access_wraps_around_memory

TAGS **UNPREDICTABLE** DISPLAY Access wraps around memory in %{WIDTH:d}bit mode.

Fields:

WIDTH unsigned int

Address width.

ArchMsg.Warning.warning_bcr_linking_status

TAGS **UNPREDICTABLE** DISPLAY DBGBCR.{BT2,BT} combination gives unpredictable behaviour.

ArchMsg.Warning.warning_bcr_mask_reserved

TAGS **UNPREDICTABLE** DISPLAY DBGBCR_ELn.MASK is set to a reserved value.

ArchMsg.Warning.warning_bvr_masked_bit_not_zero

TAGS **UNPREDICTABLE** DISPLAY DBGBVRn_EL1.VA contains a bit masked by DBGBCRn_EL1.MASK that is not zero.

ArchMsg.Warning.warning_ccsidr_unimplemented_level

TAGS **UNPREDICTABLE** DISPLAY CCSIDR read while CSSELR %{CSSELR:x} points to an unimplemented cache level. Fields:

CSSELR unsigned int

current effective value of CSSELR.

ArchMsg.Warning.warning_change_to_ns_when_tge_set

TAGS **UNPREDICTABLE** DISPLAY Attempting to change NS to 1 when HCR.TGE = 1.

ArchMsg.Warning.warning_contiguous_bit_check_abort

TAGS **UNPREDICTABLE** DISPLAY An abort occurred whilst attempting to check TLB entry is contiguous at %{ENTRY_ADDR}. Fields:

ENTRY_ADDR unsigned int

address of conflicting TLB entry.

ArchMsg.Warning.warning_contiguous_bit_error

TAGS **UNPREDICTABLE** DISPLAY TLB entry at %{ENTRY_ADDR} was not contiguous with entry at %{CONTIG_ADDR} Contents are %{ENTRY_DATA} but expected %{CONTIG_DATA}. Fields:

CONTIG_ADDR unsigned int

address of the TLB first read, with which this entry is expected to be contiguous.

CONTIG_DATA unsigned int

expected contents based on the entry at CONTIG_ADDR.

ENTRY_ADDR unsigned int

address of conflicting TLB entry.

ENTRY_DATA unsigned int

contents of conflicting TLB entry.

ArchMsg.Warning.warning_contiguous_bit_error_D128

TAGS **UNPREDICTABLE** DISPLAY TLB entry at %{ENTRY_ADDR} was not contiguous with entry at %{CONTIG_ADDR} Contents are %{ENTRY_DATA_HI} %{ENTRY_DATA_LO} but expected %{CONTIG_DATA}. Fields:

CONTIG_ADDR unsigned int

address of the TLB first read, with which this entry is expected to be contiguous.

CONTIG_DATA unsigned int

expected contents based on the entry at CONTIG_ADDR.

ENTRY_ADDR unsigned int

address of conflicting TLB entry.

ENTRY_DATA_HI unsigned int

Higher bit contents of conflicting TLB entry.

ENTRY_DATA_LO unsigned int

Lower bit contents of conflicting TLB entry.

ArchMsg.Warning.warning_cp10_cp11_mismatch

TAGS **UNPREDICTABLE** DISPLAY Unpredictable configuration of CP10 and CP11 controls at EL %{EL:d}. Fields:

EL unsigned int

EL owning the control.

ArchMsg.Warning.warning_cp10_cp11_reserved_value

TAGS **UNPREDICTABLE** DISPLAY Unpredictable configuration of CPACR CP10 ({CP10}) or CP11 ({CP11}). Fields:

CP10 unsigned int

CP10 access.

CP11 unsigned int

CP11 access.

ArchMsg.Warning.warning_csselr_level_out_of_range

TAGS **UNPREDICTABLE** DISPLAY CSSELR.Level out of range, written value {LEVEL} with only {IMPLEMENTED:d} levels implemented. Fields:

IMPLEMENTED unsigned int

number of cache levels implemented.

LEVEL unsigned int

written value of CSSELR.Level.

TYPE enum

The type of cache selected by CSSELR.InD and TnD.

ArchMsg.Warning.warning_debug_flow_control_bits_not_obeyed

TAGS **UNPREDICTABLE** DISPLAY Flow control bits for {REGISTER} were not obeyed, causing {ERROR}. Fields:

ERROR enum

type of error resulting.

REGISTER enum

register under consideration.

ArchMsg.Warning.warning_debug_register_access_during_reset

TAGS **UNPREDICTABLE** DISPLAY Unpredictable {IS_WRITE:(read|write)} access to debug register offset {OFFSET} during reset. Fields:

IS_WRITE unsigned int

Write Not Read.

OFFSET unsigned int

Register Offset.

ArchMsg.Warning.warning_decode_cps_inconsistent_fields

TAGS **UNPREDICTABLE** DISPLAY The mode setting fields in the CPS instruction are inconsistent.

ArchMsg.Warning.warning_decode_invalid_state

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable in current exception-level/security state.

ArchMsg.Warning.warning_default_cacheable_mmu_on

TAGS **UNPREDICTABLE** DISPLAY Default cacheable is enabled (HCR.DC = 1) while MMU is enabled.

ArchMsg.Warning.warning_default_cacheable_vmmu_off

TAGS **UNPREDICTABLE** DISPLAY Default cacheable is enabled (HCR.DC = 1) while HCR.VM = 0.

ArchMsg.Warning.warning_deprecated_wvr_bit_2_set

TAGS **UNPREDICTABLE** DISPLAY DBGWVR%{N:d}_EL1=%{WVR:h} has bit 2 set, deprecated in ARMv8. Fields:

N unsigned int

index.

WVR unsigned int

DBGWVR.

ArchMsg.Warning.warning_eret_while_software_step_active_pending

Missing ISB between setting MDSCR_EL1.SS and ERET TAGS **UNPREDICTABLE**.

ArchMsg.Warning.warning_exclusive_to_non_normal

TAGS **UNPREDICTABLE** DISPLAY Exclusive Access to Non-Normal Memory %{ADDR}. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_exclusive_to_non_writeback

TAGS **UNPREDICTABLE** DISPLAY Exclusive Access to Non-Writeback-Cacheable Memory %{ADDR}. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_execute_from_device_memory

TAGS **UNPREDICTABLE** DISPLAY An attempt was made to execute from Device memory.

ArchMsg.Warning.warning_illegal_cpsr_mode

TAGS **UNPREDICTABLE** DISPLAY Writing an illegal or unimplement mode (%{NEW_MODE:h}) to CPSR was unpredictable. Fields:

NEW_MODE unsigned int

New value of CPSR.M.

ArchMsg.Warning.warning_illegal_srs_mode

TAGS **UNPREDICTABLE** DISPLAY Illegal or **UNPREDICTABLE** mode (%{MODE:h}) used for SRS instruction. Fields:

MODE unsigned int

mode for Banked SP.

ArchMsg.Warning.warning_implementation_defined_read_debug_register_in_SCS

Read Access to Debug registers from the processor is IMPLEMENTATION DEFINED TAGS IMP_DEF. Fields:

PPB_OFFSET unsigned int
debug register offset.

ArchMsg.Warning.warning_implementation_defined_sequential_security_transitions_supported

The behavior of sequential instruction fetches that cross from non-secure to secure memory and contain an SG instruction is CONSTRAINED UNPREDICTABLE TAGS UNPREDICTABLE.

ArchMsg.Warning.warning_implementation_defined_stack_limit_check_supported

It is IMPLEMENTATION DEFINED whether stack pointer limit checking is performed for this instruction TAGS IMP_DEF. Fields:

ADDRESS unsigned int
Address of instruction.

OPCODE unsigned int
instruction opcode.

ArchMsg.Warning.warning_implementation_defined_taggednonsharedmemorytype

TAGS IMP_DEF DISPLAY Memory attribute has Tagged, non shared memory type; it is IMPLEMENTATION DEFINED whether the memory location is treated as Tagged or non-Tagged.

ArchMsg.Warning.warning_implementation_defined_write_debug_register_in_SCS

Write Access to Debug registers from the processor is IMPLEMENTATION DEFINED TAGS IMP_DEF. Fields:

DATA unsigned int
data attempted to be written.

PPB_OFFSET unsigned int
debug register offset.

ArchMsg.Warning.warning_invalid_tcr_granule

TAGS UNPREDICTABLE DISPLAY TCR.TG%{TG_ID:(0|1)} bits have been set to a granule size %{{REQUEST:d}}K not implemented - using %{{SUBSTITUTE:d}}K. Fields:

REQUEST signed int
page size requested (or 0 for reserved).

SUBSTITUTE signed int
best guess available page size.

TG_ID bool
bits TG0 or TG1.

ArchMsg.Warning.warning_load_multiple_user_registers_from_user_mode

TAGS UNPREDICTABLE DISPLAY An LDM(user registers) instruction executed from user mode.

ArchMsg.Warning.warning_mte_in_mem_attr_not_supported

TAGS UNPREDICTABLE DISPLAY Memory attribute %{ATTR} has Tagged memory type but Armv8.5-MTE is not implemented. Fields:

ATTR unsigned int

Memory attribute.

ArchMsg.Warning.warning_reserved_breakpoint_state_match

TAGS UNPREDICTABLE DISPLAY %{IS_BREAKPOINT:(Watchpoint|Breakpoint)} programmed with a reserved combination of HMC, SSC and %{IS_BREAKPOINT:(PAC|PMC)}. Fields:

IS_BREAKPOINT unsigned int

Is a breakpoint, not a watchpoint.

ArchMsg.Warning.warning_shareability

TAGS UNPREDICTABLE DISPLAY Unpredictable: combination of the 1st and 2nd stages of translation is Normal Inner Non-Cacheable, Outer Non-Cacheable.

ArchMsg.Warning.warning_software_step_set_while_enabled

TAGS UNPREDICTABLE DISPLAY MDSCR_EL1.SS set to 1 while software step debug exceptions are enabled.

ArchMsg.Warning.warning_thumb_instruction_wraps_around_memory

TAGS UNPREDICTABLE DISPLAY Thumb instruction wraps around memory.

ArchMsg.Warning.warning_ttbr_sbz_bits_are_not_zero

TAGS UNPREDICTABLE DISPLAY %{TTBR}[%{MSB:d}:0] should be zero, but are not. Fields:

MSB unsigned int

x-1.

TTBR enum

which TTBR register.

ArchMsg.Warning.warning_unaligned_address_dbgdtrrx_write

TAGS UNPREDICTABLE DISPLAY Unaligned address %{ADDR} in X0/R0 when DBGDTRRX written in memory access mode. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unaligned_address_dbgdtrtx_read

TAGS UNPREDICTABLE DISPLAY Unaligned address %{ADDR} in X0/R0 when DBGDTRTX read in memory access mode. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unaligned_to_device

TAGS UNPREDICTABLE DISPLAY Unaligned Access to Device Memory %{ADDR}. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unaligned_to_strongly_ordered

TAGS **UNPREDICTABLE** DISPLAY Unaligned Access to Strongly Ordered Memory %{ADDR}.

Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unknown_sau_rnr

TAGS **UNPREDICTABLE** DISPLAY SAU_RNR was set to an unsupported value.

ArchMsg.Warning.warning_unpredictable_AIRCR_PRIS_and_BFHFNMINs

TAGS **UNPREDICTABLE** DISPLAY The effect of setting both AIRCR.BFHFNMINs and AIRCR.PRIS to 1 is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_AIRCR_VECTCLRACTIVE_when_not_in_debug

TAGS **UNPREDICTABLE** DISPLAY The effect of writing a 1 to AIRCR.VECTCLRACTIVE if the processor is not halted in Debug state is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_AIRCR_VECTRESET_and_SYSRESETREQ

TAGS **UNPREDICTABLE** DISPLAY When the processor is halted in Debug state, if a write to the register writes a 1 to both VECTRESET and SYSRESETREQ, the behavior is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_AIRCR_VECTRESET_when_not_in_debug

TAGS **UNPREDICTABLE** DISPLAY The effect of writing a 1 to AIRCR.VECTRESET if the processor is not halted in Debug state is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_AIRCR_incorrect_VKEY

TAGS **UNPREDICTABLE** DISPLAY The value 0x05FA must be written to AIRCR.VKEY, otherwise the register write is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_EXC_RETURN_Reserved_Bit

EXC_RETURN[23:7] are reserved with the special condition that all bits should be written as one. Values other than all 1s are **UNPREDICTABLE**. TAGS **UNPREDICTABLE**. Fields:

EXC_RETURN unsigned int

Special PC value.

SBO_MASK unsigned int

Which bits Should Be One to not UNPRED.

ArchMsg.Warning.warning_unpredictable_PMEVTYPEPERn_ELO_VS_reserved_value

TAGS **UNPREDICTABLE** DISPLAY Unpredictable reserved value of PMEVTYPER<n>_ELO.VS.

ArchMsg.Warning.warning_unpredictable_change_to_DEMCR_MON_STEP_at_insufficient_priority

TAGS **UNPREDICTABLE** DISPLAY The effect of changing DEMCR.MON_STEP at an execution priority that is lower than the priority of the DebugMonitor exception is **CONSTRAINED UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_change_to_priority_of_active_exception

TAGS UNPREDICTABLE DISPLAY Changing the priority of an active exception is UNPREDICTABLE.

ArchMsg.Warning.warning_unpredictable_clear_lspact

TAGS UNPREDICTABLE DISPLAY UNPREDICTABLE state on context switch. Fields:

DRVEXC_TAKEN unsigned int

Derived exception taken, not pended.

REACHED_LAST_STORE unsigned int

Only the last faulted.

ArchMsg.Warning.warning_unpredictable_dbg_exit_unaligned_dlr

TAGS UNPREDICTABLE DISPLAY Debug exit to AArch32 with DLR[0] set to 1. PC[0] can be set to 0 or to DLR[0].

ArchMsg.Warning.warning_unpredictable_exception_catch

TAGS UNPREDICTABLE DISPLAY Generation of an exception catch event is unpredictable.

ArchMsg.Warning.warning_unpredictable_exception_return_inconsistent_state

TAGS UNPREDICTABLE DISPLAY state on exception return is unpredictable.

ArchMsg.Warning.warning_unpredictable_exception_return_instruction

TAGS UNPREDICTABLE DISPLAY exception return instruction is unpredictable.

ArchMsg.Warning.warning_unpredictable_in_debug_state

TAGS UNPREDICTABLE DISPLAY Instruction is UNPREDICTABLE when executed in debug state.

ArchMsg.Warning.warning_unpredictable_mair_encoding

TAGS UNPREDICTABLE DISPLAY Memory attribute %{ATTR} has invalid MAIR encoding; this may lead to unpredictable results. Fields:

ATTR unsigned int

Memory attribute.

ArchMsg.Warning.warning_unpredictable_pmu_counter_access

TAGS UNPREDICTABLE DISPLAY Access to PMU counters from non-secure EL0 or EL1 is UNPREDICTABLE with MDCR_EL2.HPMN set to 0.

ArchMsg.Warning.warning_unpredictable_prioritization_breakpoint_match_vector_catch

TAGS UNPREDICTABLE DISPLAY Two events with the same priority occurred at the same instruction: Address Matching Vector Catch debug event, and Breakpoint debug event. It is constrained unpredictable which is taken.

ArchMsg.Warning.warning_unpredictable_stack_selection

TAGS UNPREDICTABLE DISPLAY UNPREDICTABLE state on context switch.

ArchMsg.Warning.warning_unpredictable_tsize_out_of_range

TAGS UNPREDICTABLE DISPLAY %{TCR}.T%{TTBR:d}SZ (%{TSIZE:d} bits) is out of range. Must be between 25 and %{TSIZE_MAX:d}. Fields:

TCR enum

Which TCR.

TSize signed int

Value of TSize.

TSize_MAX signed int

The maximum allowed value.

TTBR unsigned int

Which TTBR region.

ArchMsg.Warning.warning_unpredictable_unaligned_pc_as_base_register

TAGS **UNPREDICTABLE** DISPLAY The PC value must be word-aligned, otherwise the behavior of the instruction is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_unaligned_pop_stack

TAGS **UNPREDICTABLE** DISPLAY **CONSTRAINED UNPREDICTABLE** unaligned stack frame `%{FRAME_ADDR:x}` in exception-return. Fields:

FRAME_ADDR unsigned int

unaligned stack frame address.

ArchMsg.Warning.warning_unpredictable_unaligned_to_non_writeback

TAGS **UNPREDICTABLE** DISPLAY Unaligned Access to Non-Writeback-Cacheable Memory.

ArchMsg.Warning.warning_unpredictable_vmsa_memattrib

TAGS **UNPREDICTABLE** DISPLAY Unpredictable vmsa memory attribute with texcb value programmed to `%{TEXCB}` when tex remap is `%{TEX_REMAP}`. Fields:

TEXCB unsigned int

texcb value.

TEX_REMAP bool

use tex remap.

ArchMsg.Warning.warning_unpredictable_vtcr_t0sz_sl0

TAGS **UNPREDICTABLE** DISPLAY The combination of VTCR.TOSZ=`%{TOSZ:d}` and VTCR.SL0=`%{SL0:d}` is unpredictable. Fields:

SL0 signed int

VTCR.SL0, starting level for stage 2 translation table walks.

TOSZ signed int

VTCR.TOSZ, required input address range.

ArchMsg.Warning.warning_unpredictable_write_DHCSR

TAGS **UNPREDICTABLE** DISPLAY **UNPREDICTABLE** set of bit-changes in DHCSR. Fields:

NEW unsigned int

data attempted to be written.

OLD unsigned int

previous value.

ArchMsg.Warning.warning_unpredictable_write_SHCSR

TAGS UNPREDICTABLE DISPLAY UNPREDICTABLE setting a pending bit to 1 for an exception with priority \geq execution priority.

ArchMsg.Warning.warning_unsupported_access_to_memory_mapped_register

TAGS UNPREDICTABLE DISPLAY Access to memory mapped register is unsupported.

ArchMsg.Warning.warning_user_jmcr_access

TAGS UNPREDICTABLE DISPLAY User %{WRITE:(write|read)} access to JMCR is unpredictable. Fields:

WRITE unsigned int
access is write.

ArchMsg.Warning.warning_wcr_mask_and_bas

TAGS UNPREDICTABLE DISPLAY DBGWCR_EL1.MASK is non-zero and BAS is not 0xff.

ArchMsg.Warning.warning_wcr_mask_reserved

TAGS UNPREDICTABLE DISPLAY DBGWCR_ELn.MASK is set to a reserved value.

ArchMsg.Warning.warning_wcr_non_configuous_bas

TAGS UNPREDICTABLE DISPLAY DBGWCR_EL1.BAS has non-contiguous set of ones.

ArchMsg.Warning.warning_wvr_masked_bit_not_zero

TAGS UNPREDICTABLE DISPLAY DBGWVRn_EL1.VA contains a bit masked by DBGWCRn_EL1.MASK that is not zero.

ArchMsg.Why.why_branch_target_exception

A branch target exception has occurred. DISPLAY branch target exception: target pc=%{PC}, target opcode=%{OPCODE}. PRIMARY KEY PC. Fields:

BTYPE unsigned int
pstate.BTYPE at time branch taken.

OPCODE unsigned int
opcode at branch target.

PC unsigned int
Virtual Address of branch target.

ArchMsg.Why.why_illegal_state

An illegal mode change has occurred. DISPLAY illegal mode change: %{MESSAGE}. Fields:

MESSAGE string
The message.

BRANCH_MISPREDICT

Simulating branch mispredict. Fields:

PC unsigned int
Origin address (or 0 if unavailable).

BRA_DIR

Direct branches, to immediate address. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISET enum

The instruction set of the branch instruction.

IS_COND bool

Indicates if this is a conditional branch.

IS_HINTED bool

Whether the conditional instruction is hinted.

IS_ISB bool

Whether the instruction is ISB.

IS_LINK bool

Indicates if this is a branch with link.

PC unsigned int

The address of the branch instruction.

TARGET_ISET enum

The instruction set after the branch.

TARGET_PC unsigned int

The address the instruction branches to.

BRA_INDIR

Indirect branches, perhaps to a register. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISET enum

The instruction set of the branch instruction.

IS_COND bool

Indicates if this is a conditional branch.

IS_HINTED bool

Whether the conditional instruction is hinted.

IS_ISB bool

Whether the instruction is ISB.

IS_LINK bool

Indicates if this is a branch with link.

PC unsigned int

The address of the branch instruction.

TARGET_ISET enum

The instruction set after the branch.

TARGET_PC unsigned int

The address the instruction branches to.

BRA_INDIR_CCFAIL

Indirect branches fail, perhaps to a register.

BRA_RET_CCFAIL

Branch return fail, perhaps to a register.

CACHE_MAINTENANCE_OP

Cache Maintenance Operation. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

DATA unsigned int

Specified MVA or Set/way.

FUNCTION enum

Clean / Invalidate.

SCOPE enum

Affected region.

SIDE enum

Inst / Data.

CCFAIL

Conditional instruction condition check fail. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CCFAIL_UNC

Unconditional instruction condition check fail. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CCPASS

Conditional instruction condition check pass. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CCPASS_UNC

Unconditional instruction condition check pass. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CODE_CACHE_MAINT

Code cache maintenance. Fields:

END_INCL unsigned int

Inclusive end address for VA or PA requests.

NS bool

Non-secure world for PA.

START unsigned int

Start address for VA or PA requests.

TYPE enum

Request type.

COMPILE_BLOCK_END

Last instruction of basic block translated. Fields:

VADDR unsigned int

Address of next instruction after this basic block.

COMPILE_BLOCK_START

First instruction of basic block translated. Fields:

VADDR unsigned int

Address of first instruction in basic block.

COMPILE_INST

ARM instruction compiled. Fields:

DISASS string

Disassembly of the instruction.

ISET enum

The instruction set of this instruction.

ITSTATE unsigned int

The ITSTATE current for the instruction.

OPCODE unsigned int

The opcode of the instruction.

PC unsigned int

The address of the instruction.

SIZE unsigned int

The size of the instruction in bytes.

CONTEXTIDR

Context ID Register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

NS bool

Secure or nonsecure banked register is accessed.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The new value written.

CONTEXT_SYNC

Called for every context synchronization event. Fields:

ARCH_CSE bool

Architectural Context Synchronization Event.

PC unsigned int

The address of the synchronization instruction.

SYSREG_SYNC bool

System register synchronization event.

CORE_ENDIAN

Core BE8 Big-Endian state changed. Fields:

BE8 bool

Core BE8 Big-Endian state.

CORE_INFO

Static processor attributes. Only triggered by a call to DumpState(). Fields:

ARCH_PROFILE enum

The architecture profile of the core.

CLUSTER_ID unsigned int

The cluster ID of this processor.

CORE_NUM unsigned int

The number of this core in an MP processor.

DETAILED_CORE_FEATURES enum

Detailed core features as a bitfield. Enum values are bitmasks to extract the info from the field.

FPU_VERSION enum

The VFP version implemented by the core.

MEM_ARCH enum

The memory architecture of the core.

MIDR_EL1 unsigned int

MIDR_EL1 value.

NUM_CORES unsigned int

The number of cores in this MP processor.

QUANTUM_SIZE unsigned int

The default quantum size of the core (in ticks).

SECURITY_FEATURES bool

Does the core have security features?.

CORE_LOADS

Processor load accesses. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ACQREL enum

Is this an acquire/release.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DATA unsigned int

The data read or written.

ELEMENT_SIZE unsigned int

Width of each element.

ENDIANNESS enum

Byte ordering of this access.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

LOCK enum

Normal, exclusive or locked access.

MECID unsigned int

Memory Encryption Context ID, if applicable.

MECID2 unsigned int

Memory Encryption Context ID of the second page, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PAGE_SIZE unsigned int

Page size as $\log_2(\text{size in bytes})$.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK, 2=Exclusive Failed, 3=Unsupported.

SIZE unsigned int

Width of the access in bytes. Only required if DATA is not traced.

TRANS bool

Is this a translated access.

VADDR unsigned int

The virtual address of the access.

CORE_REGS

Changes of the core registers R0 to R14. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number, 0 to 14.

MODE enum

Bank of the register accessed.

OLD_VALUE unsigned int

The old value overwritten.

PHYS_ID enum

The physical register accessed.

VALUE unsigned int

The new value written to the register.

CORE_REGS64

Changes of the core registers X0..X30, SP_ELn. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ID enum

The register number, 0-30 for X0-X30, >=32 for SP_ELn.

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

CORE_REGS64_READ

General purpose AArch64 register has been read X0..X30, SP_ELn. Fields:

ID enum

The register number, 0-30 for X0-X30, >=32 for SP_ELn.

VALUE unsigned int

The value read from the register.

CORE_STALLED_TRANSITION

Detect when a core is prevented from progressing PChannel states. Fields:

CURRENT_STATE enum

Current power mode of core.

REASON string

The reason a transition was stalled, and what must be done to unblock it.

TARGET_STATE enum

Desired power mode of core after transition.

CORE_STORES

Processor store accesses. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ACQREL enum

Is this an acquire/release.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DATA unsigned int

The data read or written.

ELEMENT_SIZE unsigned int

Width of each element.

ENDIANNESS enum

Byte ordering of this access.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

LOCK enum

Normal, exclusive or locked access.

MECID unsigned int

Memory Encryption Context ID, if applicable.

MECID2 unsigned int

Memory Encryption Context ID of the second page, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PAGE_SIZE unsigned int

Page size as $\log_2(\text{size in bytes})$.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK, 2=Exclusive Failed, 3=Unsupported.

SIZE unsigned int

Width of the access in bytes. Only required if DATA is not traced.

TRANS bool

Is this a translated access.

VADDR unsigned int

The virtual address of the access.

CP14_READ

System Coprocessor register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP14 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc1 unsigned int

opcode 1.

opc2 unsigned int

opcode 2.

CP14_WRITE

System Coprocessor register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP14 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc1 unsigned int

opcode 1.

opc2 unsigned int

opcode 2.

CP15_READ

System Coprocessor register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP15 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc1 unsigned int

opcode 1.

opc2 unsigned int

opcode 2.

CP15_READ64

System Coprocessor register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP15 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc unsigned int

opcode 1.

CP15_WRITE

System Control Coprocessor register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP15 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc1 unsigned int

opcode 1.

opc2 unsigned int
opcode 2.

CP15_WRITE64

System Coprocessor register write. Fields:

CORE_NUM unsigned int
Core number in a multi processor.

CRm unsigned int
CRm.

NS bool
Secure or nonsecure banked register is accessed.

REG_NAME string
Name of the CP15 register accessed.

SECURITY_STATE enum
The current security state.

UNDEF bool
The register accessed is undefined.

UPDATED_VALUE unsigned int
Updated value of the register now it has been written.

VALUE unsigned int
The new value written.

opc unsigned int
opcode 1.

CPSR

CPSR change. (In AArch64 PSTATE represented in SPSR format). Fields:

CORE_NUM unsigned int
Core number in a multi processor.

OLD_VALUE unsigned int
The old CPSR value.

UNKNOWN unsigned int
Bits within the register that have unknown value.

VALUE unsigned int
The new CPSR value.

CRYPTO_SPEC

Every crypto instruction speculatively executed.

CorePowerStateChange

Core power-up state has changed. DISPLAY Core powered %{PU:(up|down)}. Fields:

PU bool

Whether the core powered up.

DATA_CACHE_SET_ALLOC_TAGS

Set allocation tags from DC GVA instruction. Fields:

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

NSDESC unsigned int

The security state of the access.

PADDR unsigned int

The physical address of the access.

PAS enum

The physical address space of the page.

RESPONSE enum

0=Aborted, 1=OK.

SIZE unsigned int

The size of memory in bytes.

TAG unsigned int

The value of the tag being set.

TAGGING_TYPE enum

Type of Memory tagging (Physical or Virtual).

TAG_VADDR unsigned int

The virtual address of the Virtual Tag if Virtual tagging is used; otherwise 0.

VADDR unsigned int

Virtual address used for setting allocation tags.

DATA_CACHE_ZERO

Zero and invalidate cache line from DC ZVA instruction. Fields:

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

NSDESC unsigned int

The security state of the access.

PADDR **unsigned int**

The physical address of the access.

PADDR_TAG **unsigned int**

If tag checked, the physical address tag for the access.

PAS **enum**

The physical address space of the page.

RESPONSE **enum**

0=Aborted, 1=OK.

SIZE **unsigned int**

The size of zero'd memory in bytes.

TAG_CHECKED **bool**

Is this access tag checked.

VADDR **unsigned int**

Virtual address of the zero'd cache line.

DEBUG_EVENT

Hardware debug support event. Fields:

EVENT **enum**

Description of event.

VALUE **unsigned int**

data value.

DEBUG_TARGET_ISET

Hardware debug target ISA. Fields:

TARGET_ISET **enum**

The target instruction set of the debug state.

DMI_ALLOCATE

DMI allocated. Fields:

ACCESS_DIRECTION **enum**

READ / WRITE.

PADDR **unsigned int**

The physical address of the DMI.

PTR **unsigned int**

DMI range pointer.

SIDE **enum**

DSIDE / ISIDE.

SIZE **unsigned int**

The size of DMI.

VADDR unsigned int

The virtual address of the DMI..

DMI_HIT

DMI hit, access used DMI (only fire for small DMI regions i.e. less than a full page). Fields:

ACCESS_DIRECTION enum

READ / WRITE.

PADDR unsigned int

The physical address of the DMI.

SIDE enum

DSIDE / ISIDE.

SIZE unsigned int

The size of the DMI.

VADDR unsigned int

The virtual address of the DMI..

DMI_REVOKE

DMI revoked. Fields:

ACCESS_DIRECTION enum

READ / WRITE.

PADDR unsigned int

The physical address of the DMI.

SIDE enum

DSIDE / ISIDE.

SIZE unsigned int

The size of DMI.

VADDR unsigned int

The virtual address of the DMI..

END_COMPILE

Compilation end. Fields:

END_OF_PAGE_COUNT unsigned int

Number of basic blocks exited due to page boundary since START_COMPILE.

FETCHFAIL_COUNT unsigned int

Number of basic blocks exited due to ifetch failure START_COMPILE.

INST_COUNT unsigned int

Number of instructions compiled since START_COMPILE.

NONSEQ_COUNT unsigned int

Number of basic blocks exited due to non-sequential instructions since START_COMPILE.

PAGE_STRADDLE_COUNT unsigned int

Number of basic blocks exited due to unaligned instructions crossing page since START_COMPILE.

ESB

Error Synchronization Barrier hit. Fields:

ASYNC bool

ESB is async.

IMPLICIT bool

ESB is implicit.

EXCEPTION

Exception taken. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ESR unsigned int

The value assigned to ESR when exception is taken.

ESR_EC unsigned int

The Value assigned to ESR.EC when exception is taken.

IS_PHYSICAL bool

Physical or Virtual exception.

IS_TRAP bool

Exception coming from a trap.

LR unsigned int

The value assigned to the link register.

PC unsigned int

The location where the exception occurred.

PREFERRED_RETURN unsigned int

The preferred return address for the exception.

TAKEN_LOCALLY bool

Whether exception is taken locally(same exception level).

TARGET_ISET enum

The instruction set of the exception handler code.

TARGET_PC unsigned int

The address the exception branches to.

VECTOR enum

The exception vector.

VECTOR_OFFSET unsigned int

The offset of an exception vector from the base address.

EXCEPTION_END

Every exception completed.

EXCEPTION_INFO

Trace human-readable reason of the current exception for data/instruction aborts, floating point exceptions and unknown exceptions. Fields:

DESCRIPTION string

Reason for current exception.

ESR_EC unsigned int

The value of ESR.EC when exception is taken.

EXCEPTION_RAISE

Every exception raised.

EXCEPTION_RETURN

Branches on leaving exception. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISSET enum

The instruction set of the branch instruction.

IS_COND bool

Indicates if this is a conditional branch.

IS_HINTED bool

Whether the conditional instruction is hinted.

IS_ISB bool

Whether the instruction is ISB.

IS_LINK bool

Indicates if this is a branch with link.

PC unsigned int

The address of the branch instruction.

TARGET_ISET enum

The instruction set after the branch.

TARGET_PC unsigned int

The address the instruction branches to.

EXCEPTION_RETURN_PREBRANCH

Exception return event that must occur before PC and processor state have updated.

EXCEPTION_START

Every exception started.

FIQ_TAKEN

FIQ taken exception.

FP_EXCEPTIONS_SIGNALLED

Trace which FP exceptions were signalled as a part of the FP operations performed by an instruction. Fields:

FP_EXCEPTIONS_SIGNALLED_MASK unsigned int

Indicates which FP exceptions were signalled by an instruction. This uses the same encoding as the cumulative floating-point exception bitfields in FPSR.

FP_STATE

Floating point state. Fields:

FPSR unsigned int

FPSR register value.

MASK unsigned int

The mask indicating updated cumulative bits.

FREQ_CHANGED

Core frequency changed. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The instruction count of this core.

NEW_FREQ signed int

The new frequency of this core (expressed in Hz).

OLD_FREQ signed int

The old frequency of this core (expressed in Hz).

HLT

HLT instruction occurred. Fields:

IMM16 unsigned int

The 16-bit immediate value encoded in HLT instruction.

INST

Every instruction executed. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DEBUG_STATE bool

The instruction is executed in debug state.

DISASS string

Disassembly of instruction.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISET enum

The current instruction set.

ITSTATE unsigned int

The current ITSTATE.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

MODE enum

The mode the core is in.

NS unsigned int

The core's non-secure bit.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

OPCODE unsigned int

The opcode of the instruction.

PADDR unsigned int

The physical address of the instruction.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the instruction.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the page.

PC unsigned int

The address of the instruction.

SECURITY_STATE enum

The core's security state.

SIZE unsigned int

The size of the instruction in bytes.

INST_END

Every instruction completed.

INST_START

Every instruction started. Fields:

DEBUG_STATE **bool**

The instruction is executed in debug state.

ISET **enum**

The current instruction set.

MODE **enum**

The mode the core is in.

NS **enum**

The current Secure State.

PC **unsigned int**

The address of the conditional instruction.

SECURITY_STATE **enum**

The current Security State.

INST_STRADDLE

Instruction straddles boundary. Fields:

DEBUG_STATE **bool**

The instruction is executed in debug state.

ISET **enum**

The current instruction set.

MODE **enum**

The mode the core is in.

NS **enum**

The current Secure State.

PC **unsigned int**

The address of the conditional instruction.

SECURITY_STATE **enum**

The current Security State.

IRQ_TAKEN

IRQ taken exception.

ISB

ISB instruction occurred. Fields:

PC **unsigned int**

The address of the ISB instruction.

LOAD_MULTIPLE_REGS_LIST

List of destination registers (not source) of a multiple registers load instruction. Fields:

REG string

Register to add into the list.

LOCAL_MONITOR

Local monitor activity. Fields:

PADDR unsigned int

Local Monitor Address.

State enum

State of the monitor (Open/Exclusive).

MAPPING_CREATE

A memory mapping has been created. This is an implementation detail of the model and so may change without notice. Fields:

ASID unsigned int

The ASID used during translation, or 0 if no ASID was used.

IS_VALID bool

Whether the mapping is valid or not.

PADDR unsigned int

Base physical address of region, or 0 if not known.

SIDE enum

I-side or D-side.

SIZE unsigned int

Size of the page, or 0 if not known.

VADDR unsigned int

Base virtual address of region.

VMID unsigned int

The VMID used during translation, or 0 if no VMID was used.

MAPPING_DESTROY

A memory mapping has been destroyed. This is an implementation detail of the model and so may change without notice. Fields:

ASID unsigned int

The ASID used during translation.

CAUSE enum

The cause of invalidation.

PADDR unsigned int

Base physical address of page.

SIDE enum

I-side or D-side.

SIZE unsigned int

Size of the page.

VADDR unsigned int

Base virtual address of page.

VMID unsigned int

The VMID used during translation.

MEMMAP_DEBUG_READ

Memory mapped reads to the debug registers. Fields:

ADDR unsigned int

address.

CORE_NUM unsigned int

Core number in a multi processor.

EXT bool

Whether access is from an external device (such as the DAP).

NS enum

Secure state of the access.

REG_NAME string

Name of the debug register accessed.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

MEMMAP_DEBUG_WRITE

Memory mapped writes to the debug registers. Fields:

ADDR unsigned int

address.

CORE_NUM unsigned int

Core number in a multi processor.

EXT bool

Whether access is from an external device (such as the DAP).

NS enum

Secure state of the access.

REG_NAME string

Name of the debug register accessed.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The new value written.

MMU_D128_TTB_READ

This event is triggered by reads caused by a D128 translation table walk. Fields:

ABORTED bool

Was the walk successful.

DATA_HIGH unsigned int

The data upper bits read.

DATA_LOW unsigned int

The data lower bits read.

IPA unsigned int

For stage 1, the IPA of the read.

LEVEL unsigned int

Translation table level.

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

OUTADDR unsigned int

The output physical address.

PADDR unsigned int

The physical address of the read.

PASmem enum

Physical Address Space of the accessed memory.

PASreq enum

Physical Address Space of the requested lookup.

REGIME_EL enum

Entry matches in this translation regime.

REQUESTER enum

The requester of this table walk.

SIDE enum

Inst / Data.

STAGE unsigned int

Translation stage.

MMU_D128_TTB_WRITE

This event is triggered by writes caused by a D128 translation table walk. Fields:

ABORTED bool

Was the walk successful.

DATA_HIGH unsigned int

The data upper bits written.

DATA_LOW unsigned int

The data lower bits written.

IPA unsigned int

For stage 1, the IPA of the write.

IS_HACDBS_ACCESS bool

If the access was due to HACDBS_CLEAN.

LEVEL unsigned int

Translation table level.

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

OUTADDR unsigned int

The output physical address.

PADDR unsigned int

The physical address of the write.

PASmem enum

Physical Address Space of the accessed memory.

PASreq enum

Physical Address Space of the requested lookup.

REGIME_EL enum

Entry matches in this translation regime.

SIDE enum

Inst / Data.

STAGE unsigned int

Translation stage.

MMU_TRANS

Address translation information. Fields:

ASID unsigned int

Address space identifier.

CORE_NUM unsigned int

Core number in a multi processor.

Hyp bool

Entry matches in Hyp state only.

INNERCACHE_RA bool

Is the inner cache allocate on read.

INNERCACHE_TRANSIENT bool

Is the inner write-through transient.

INNERCACHE_TYPE enum

Inner Caching scheme (NC/MB/WA).

INNERCACHE_WA bool

Is the inner cache allocate on write.

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MEMTYPE enum

Memory type.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC enum

Is secure side supposed to access secure or nonsecure memory.

OUTERCACHE_RA bool

Is the outer cache allocate on read.

OUTERCACHE_TRANSIENT bool

Is the outer write-through transient.

OUTERCACHE_TYPE enum

Outer Caching scheme (NC/MB/WA).

OUTERCACHE_WA bool

Is the outer cache allocate on write.

PADDR unsigned int

Physical address of the access.

PAGESIZE unsigned int

Page size as log2(size).

PAS enum

PAS supposed to be accessed is secure or nonsecure memory or Root/Realm if RME is implemented.

SH enum

Shareability.

SIDE enum

Inst / Data.

STAGE1_PERM unsigned int

Stage 1 permissions, [5:3] Privileged access (XWR) [2:0] User access (XWR).

STAGE2_PERM unsigned int

Stage 2 permission mask [5:3] Privileged access (XWR) [2:0] User access (XWR).

VADDR unsigned int

Virtual address of the access.

VMID unsigned int

Virtual machine identifier.

XS bool

XS Attribute.

nG enum

Flag indicating whether ASID will be matched.

MMU_TTB_READ

This event is triggered by reads caused by a translation table walk. Fields:

ABORTED bool

Was the walk successful.

DATA unsigned int

The data read.

DESC_KIND enum

The encoding scheme of the descriptors used for this walk.

IPA unsigned int

For stage 1, the IPA of the read.

LEVEL unsigned int

Translation table level.

LPAE bool

Is this for an LPAE translation (*DEPRECATED*, use DESC_KIND).

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSmem bool

Non secure memory access.

NSreq bool

Non secure request.

OUTADDR unsigned int

The output physical address.

PADDR unsigned int

The physical address of the read.

PASmem enum

Physical Address Space of the accessed memory.

PASreq enum

Physical Address Space of the requested lookup.

REGIME_EL enum

Entry matches in this translation regime.

REQUESTER enum

The requester of this table walk.

SIDE enum

Inst / Data.

STAGE unsigned int

Translation stage.

MMU_TTB_WRITE

This event is triggered by writes caused by a translation table walk. Fields:

ABORTED bool

Was the walk successful.

DATA unsigned int

The data written.

DESC_KIND enum

The encoding scheme of the descriptors used for this walk.

IPA unsigned int

For stage 1, the IPA of the write.

IS_HACDBS_ACCESS bool

If the access was due to HACDBS_CLEAN.

LEVEL unsigned int

Translation table level.

LPAE bool

Is this for an LPAE translation (*DEPRECATED*, use DESC_KIND).

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSmem bool

Non secure memory access.

NSreq bool

Non secure request.

OUTADDR unsigned int

The output physical address.

PADDR unsigned int

The physical address of the write.

PASmem enum

Physical Address Space of the accessed memory.

PASreq enum

Physical Address Space of the requested lookup.

REGIME_EL enum

Entry matches in this translation regime.

SIDE enum

Inst / Data.

STAGE unsigned int

Translation stage.

MODE_CHANGE

Mode change. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

MODE enum

The new mode.

NON_SECURE enum

The core's new non-secure bit.

OLD_MODE enum

The old mode.

SECURITY_STATE enum

The new security state.

NOT_TAG_CHECKED_INST

MTE Tag not checked for current instruction.

PERIODIC

Called for every quantum. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The instruction count of this CPU.

PC unsigned int

The address of the next instruction to be executed on this CPU.

PMU_COUNTER_OVERFLOW

PMU counter overflow. Fields:

COUNTER_GROUP enum

Counter group.

COUNTER_RANGE enum

Counter range.

EVENT_ID unsigned int

Event Identifier, field is valid only for Event counter group.

INDEX unsigned int

Counter index (as selected by PMSELR), field is valid only for Event/Cycle counter groups.

INTERRUPT bool

Is interrupt enabled on overflow for this counter.

POTINDEX_UPDATE

Value of POT index for current pc changed. Fields:

NEW_POTINDEX unsigned int

New POTIndex.

OLD_POTINDEX unsigned int

Old POTIndex.

PREFETCH_MEMORY64

Prefetch from PRFM or PRFUM instructions. Fields:

PRFOP enum

Prefetch hint.

VADDR unsigned int

Virtual address of the location that should be prefetched.

PRELOAD64_REQUEST

Preload requests from PRFM or PRFUM instructions. Fields:

PRFOP enum

Prefetch hint.

VADDR unsigned int

Virtual address of the location that should be prefetched.

PRELOAD_DATA

Data preload from PLD instruction. Fields:

VADDR unsigned int

Virtual address of the data that should be preloaded.

PRELOAD_INST

Instruction preload from PLI instruction. Fields:

VADDR unsigned int

Virtual address of the instruction that should be preloaded.

PRE_CORE_LOAD

Trace just before a core load. Fields:

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

PRE_CORE_STORE

Trace just before a core store. Fields:

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

PRE_TTB_READ

This event is triggered before reads caused by a translation table walk. Fields:

ADDR unsigned int

The physical address of the read.

PSTATE

PSTATE fields update. Fields:

ALLINT unsigned int

All interrupt mask bit.

BTYPE unsigned int

Branch target identification bit.

CORE_NUM unsigned int

Core number in a multi processor.

DAIF unsigned int

Exception mask bits.

DIT unsigned int

Data Independent Timing bit.

EL unsigned int

Current Exception level field.

EXLOCK unsigned int

Exception return state lock bit.

IL unsigned int

Illegal Execution state bit.

NZCV unsigned int

Condition flags.

PACM unsigned int

PACM bit.

PAN unsigned int

Privileged Access Never state bit.

PM unsigned int

PMU exception mask bit.

PPEND unsigned int

PMU exception pending bit.

SM unsigned int

StreamingMode status bit.

SP unsigned int

Stack pointer register selection bit.

SS unsigned int

Software Step bit.

SSBS unsigned int

Speculative Store Bypass Safe bit.

TCO unsigned int

Tag Check Override bit.

TOUCHED unsigned int

Mask of the fields that have been touched since this trace has last been emitted. Note that touched does not necessarily mean the value changed. The index of the field is the bit index in the mask plus the index of this field plus one.

UAO unsigned int

User Access Override bit.

UINJ unsigned int

Undefined Instruction exceptions bit.

ZA unsigned int

ZA storage and ZT0 register enable bit.

nRW unsigned int

Current Execution state bit.

PSTATE_PACTIVE_OPMODE_TRANSITION

Trace pactive and Opmode mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Operating mode after the transition.

OLD_STATE enum

Operating mode before the transition.

PSTATE_PACTIVE_OPMODE_TRANSITION_DENIED

Trace the denied pactive and Opmode mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Operating mode after the transition.

OLD_STATE enum

Operating mode before the transition.

REASON string

Reason for the transition being denied.

PSTATE_PACTIVE_POWER_TRANSITION

Trace pactive and power mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Power mode after the transition.

OLD_STATE enum

Power mode before the transition.

PSTATE_PACTIVE_POWER_TRANSITION_DENIED

Trace the denied pactive and power mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Power mode after the transition.

OLD_STATE enum

Power mode before the transition.

REASON string

Reason for the transition being denied.

PSTATE_PACTIVE_THREAD_OPMODE_TRANSITION

Trace pactive and Opmode mode transitions for threads on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Operating mode after the transition.

OLD_STATE enum

Operating mode before the transition.

QUANTUM_START

Called at the start of every quantum.

RANGE_PRELOAD_REQUEST

Preload requests from RPRFM. Fields:

COUNT unsigned int

Number of blocks of data to access.

LENGTH unsigned int

Number of contiguous bytes to be accessed in each block.

OP unsigned int

Range prefetch operation .

REUSE unsigned int

Maximum number of bytes to be accessed before next RPRFM .

STRIDE signed int

Stride to advance after accessing LENGTH bytes.

VADDR unsigned int

Base virtual address of range prefetch request.

RUN_STATE

Run state transition. Fields:

INST_COUNT unsigned int

Ticks count at point of transition.

NEW enum

New run state.

OLD enum

Old run state.

SEMIHOSTING_CALL

Call of a semihost function occurred. Fields:

PC unsigned int

The program counter after the semihosting call.

SEMIHOSTING_PRECALL

About to call semihost. Fields:

EL2 bool

Use EL2 translation regime.

NS bool

Is Non-Secure.

REG_WIDTH unsigned int

The current register width in bytes.

SIGNAL

External signal state change. Fields:

SIGNAL enum

Signal that changed.

STATE bool

Signal asserted state.

SOFTWARE_STEP

Return the Debug Software Step state. Fields:

STATE enum

Software Step state.

SPSR

SPSR change. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

MODE enum

Which of the banked SPSR registers is written.

OLD_VALUE unsigned int

The old SPSR value.

VALUE unsigned int

The new SPSR value.

START_COMPILE

Compilation started. Fields:

VADDR unsigned int

Instruction where compilation begins.

STG_MULTIPLE_MICRO_OPS_ADDRESS_RANGE

Address range for MTE stores with multiple uops. Fields:

ADDR_END unsigned int

Final address.

ADDR_START unsigned int

Starting address.

STORE_MULTIPLE_REGS_ADDRESS_RANGE

Address range for stores with multiple registers. Fields:

ADDR_END unsigned int

Final address.

ADDR_START unsigned int

Starting address.

SUBOPTIMAL_LDST_RETIRED

suboptimal load/store. Fields:

ACCESS_SIZE unsigned int

Log2 of the access width in bytes used for alignment checking (1: half-word, 2: word ...).

OPTIMAL_ALIGN_SELECTOR unsigned int

Selector to decide the optimal alignment size (1: access width, 2: 4-bytes, 3: 8-bytes, ..., 12: 4-kbytes).

VADDR unsigned int

The virtual address of the access.

SYNC

Called for every synchronization, that is every quantum break. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The instruction count of this CPU.

LOCAL_QUANTUM unsigned int

The local quantum of this CPU.

LOCAL_TIME unsigned int

The local time of this CPU.

SYNC_REASON enum

Reason for sync.

SYSCALL

System call instruction executed. Fields:

IMM unsigned int

Immediate value of the system call instruction.

TYPE enum

System call type.

VADDR unsigned int

Instruction that caused the system call.

SYSREG_READ128

System Coprocessor 128 bit register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

REGNUM enum

Internal register number.

REGNUM_PHYS enum

Internal register number physically accessed (e.g. in presense of VHE redirection).

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_READ64

System Coprocessor register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REGNUM enum

Internal register number.

REGNUM_OPERAND enum

Register number corresponding to the instruction operand.

REGNUM_PHYS enum

Internal register number physically accessed (e.g. in presense of VHE redirection).

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_UPDATE128

Triggers when the system updates a register. Fields:

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG enum

Register number.

UNKNOWN unsigned int

Bits of the register which became unknown.

VALUE unsigned int

Value written to the register.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_UPDATE32

Triggers when the system updates a register. Fields:

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG enum

Register number.

UNKNOWN unsigned int

Bits of the register which became unknown.

VALUE unsigned int

Value written to the register.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int
opcode 2.

SYSREG_UPDATE64

Triggers when the system updates a register. Fields:

CRm unsigned int
CRm.

CRn unsigned int
CRn.

NS bool
Secure or nonsecure banked register is accessed.

REG enum
Register number.

UNKNOWN unsigned int
Bits of the register which became unknown.

VALUE unsigned int
Value written to the register.

opc unsigned int
opcode 1.

opc0 unsigned int
opcode 0.

opc2 unsigned int
opcode 2.

SYSREG_WRITE128

System Coprocessor 128 bit register write. Fields:

CORE_NUM unsigned int
Core number in a multi processor.

CRm unsigned int
CRm.

CRn unsigned int
CRn.

REGNUM enum
Internal register number.

REGNUM_PHYS enum
Internal register number physically accessed (e.g. in presense of VHE redirection).

UNDEF bool
The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_WRITE64

System Coprocessor register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REGNUM enum

Internal register number.

REGNUM_OPERAND enum

Register number corresponding to the instruction operand.

REGNUM_PHYS enum

Internal register number physically accessed (e.g. in presense of VHE redirection).

UNDEF bool

The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

UNALIGNED_LDST_RETIRED

Processor unaligned load/store. Fields:

ACCESS_SIZE unsigned int

Log2 of the access width in bytes used for alignment checking (1=>half-word, 2->word ...).

NUMBER_OF_BEATS unsigned int

Number of accesses (beats) in this burst (Total size in bytes is ACCESS_SIZE * NUMBER_OF_BEATS).

VADDR unsigned int

The virtual address of the access.

VFP_D_REGS

VFP/NEON D 64 bit register write. Fields:

ALIASING enum

Type of register aliasing used.

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a write to the corresponding bit in the register. For example, 0x1 would indicate a write to VALUE[0].

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

VFP_D_REGS_READ

VFP/NEON D 64 bit register read. Fields:

ID unsigned int

The register number.

VALUE unsigned int

The current value of the register.

VFP_Q_REGS

VFP/NEON Q 128 bit register write. Fields:

ALIASING enum

Type of register aliasing used.

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a write to the corresponding byte in the register. For example, 0x1 would indicate a write to VALUE[7:0].

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

VFP_Q_REGS_READ

VFP/NEON Q 128 bit register read. Fields:

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a read to the corresponding byte in the register. For example, 0x1 would indicate a read to VALUE[7:0].

VALUE unsigned int

The current value of the register.

VFP_SYS_REGS

Writes to the VFP/NEON units system registers. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ID enum

Which VFP system register is written.

MASK unsigned int

The mask indicating updated cumulative bits.

OLD_VALUE unsigned int

The register's old value overwritten.

VALUE unsigned int

The new value written to the VFP system register.

VFP_S_REGS

VFP/NEON S 32 bit register write. Fields:

ALIASING enum

Type of register aliasing used.

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a write to the corresponding bit in the register. For example, 0x1 would indicate a write to VALUE[0].

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

VFP_S_REGS_READ

VFP/NEON S 32 bit register read. Fields:

ID unsigned int

The register number.

VALUE unsigned int

The current value of the register.

VIRTUAL_TAG_ALLOCATED

Virtual tag information when allocated. Fields:

TAG_VADDR unsigned int

Virtual tag virtual address.

VADDR unsigned int

Data Virtual address.

VIRTUAL_TAG unsigned int

Virtual tag value.

WAYPOINT

Signals end of basic block execution. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ISSET enum

Origin instruction set.

IS_COND bool

Indicates if this is a conditional waypoint.

PC unsigned int

Origin address (or 0 if unavailable).

TAKEN bool

Indicates if this waypoint was taken.

TARGET unsigned int

Destination address.

TARGET_ISET enum

Destination instruction set.

WFE_END

WFE ended. Fields:

INST_COUNT unsigned int

Ticks count when leaving WFE.

WFE_EVENT_REGISTER

WFE event register status: set/clear, reason. Fields:

INST_COUNT unsigned int

Ticks count.

REASON enum

Reason for set/clear. Only REASON==0 (Cleared by WFE) clears the bit, all other reasons set it.

WFE_IGNORED

WFE ignored. Fields:

EVENT bool

This WFE was ignored because the event register was set.

INST_COUNT unsigned int

Ticks count when ignoring WFE.

TRAPPED bool

This WFE was trapped.

WFE_START

WFE entered. Fields:

INST_COUNT unsigned int

Ticks count when entering WFE.

WFI_END

WFI ended. Fields:

INST_COUNT unsigned int

Ticks count when leaving WFI.

WFI_IGNORED

WFI ignored. Fields:

DISABLED bool

This WFI was ignored because WFI is disabled.

INST_COUNT unsigned int

Ticks count when ignoring WFI.

REASON enum

specifies reason why WFI trace was ignored.

TRAPPED bool

This WFI was trapped.

WFI_START

WFI entered. Fields:

INST_COUNT unsigned int

Ticks count when entering WFI.

WFI_WAKEUP

WFI wakeup. Fields:

INST_COUNT unsigned int

Ticks count when WFI wakeup occurred.

REASON enum

Reason for wakeup.

2.15 ARM_Cortex-A35

This section describes the trace sources.

AA64_ASE_SVE_REGS

Modification of AArch64 Advanced SIMD or SVE Vector Registers (via SIMD or Scalar operations), SVE Predicate Registers and ZT0. Fields:

ID enum

The register identifier.

MASK unsigned int

Bitmask of the register to signal the modified bits in the VALUE field.

SM bool

Whether the PE is in Streaming Mode.

VALUE unsigned int

Value of AArch64 vector register.

ASYNC_MEMORY_FAULT

Context ID Register write. Fields:

FAULT unsigned int

Fault status present in ESR.ISS encoding for Serror in ARCH64 and present in DFSR encoding for external abort format in ARCH32.

PADDR unsigned int

Physical Address (or 0 if unavailable).

VADDR signed int

Virtual Address (or 0 if unavailable).

ATOMIC_END_ACCESS

Bus trace access for atomics. Fields:

ACCESS_FAIL bool

Memory access failed.

ADDR unsigned int

The virtual address of the access.

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

COMPARE_VALUE unsigned int

Compare value for CAS.

LOAD_VALUE unsigned int

Loaded values.

NSDESC unsigned int

The security state of the access.

OPERAND_VALUE unsigned int

Operation's operand.

OPERATION enum

Operation type.

PADDR unsigned int

The physical address of the access.

PRIV bool

Is this a privileged access?.

ATOMIC_START_ACCESS

Bus trace access for atomics. Fields:

ADDR unsigned int

The virtual address of the access.

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

COMPARE_VALUE unsigned int

Compare value for CAS.

NSDESC unsigned int

The security state of the access.

OPERAND_VALUE unsigned int

Operation's operand.

OPERATION enum

Operation type.

PADDR unsigned int

The physical address of the access.

PRIV bool

Is this a privileged access?.

ArchMsg.Error.exit_code

DISPLAY exit code: %{EXITCODE} component kind: %{KIND} reason: %{REASON}. Fields:

EXITCODE signed int

Exit code to be returned.

KIND string

Component kind that invoked the exit code trace.

REASON string

Reason why the exit code trace was invoked (optional).

ArchMsg.Warning.IgnoredWarmResetRequest

Event fires when software attempts to request warm reset through write to deprecated DBGPRCR.CWRR.

ArchMsg.Warning.ReportedRevisionRegisterRead

DISPLAY Reported revision and patch do not match that of the cpu. Fields:

ORIGINAL_VALUE unsigned int

The value the register would have based on the modelled revision.

REGISTER string

The name of the register that has ben read.

REPORTED_VALUE unsigned int

The value the register has been overridden to.

ArchMsg.Warning.Unpredictably Indexed PM Event Register

TAGS UNPREDICTABLE DISPLAY Event %{IsDirect:(Selector value|Register)} %{SEL} is >= %{N}. Fields:

IsDirect unsigned int

Direct Access rather than indirect.

N unsigned int

Number of Accessible Registers.

SEL unsigned int

Selector.

ArchMsg.Warning.cache_contents_unknown

A cache has been activated without being invalidated and may contain **UNKNOWN** content. **DISPLAY** %{SIDE:(D|I)}-Cache was enabled in %{REGIME:(SECURE|NON_SECURE|HYP)} regime but was not invalidated since power-on: cache lines could contain **UNKNOWN** content %{SIDE:(Note: D-side caches must invalidate SECURE lines even if the cache is only used by NON_SECURE code in order to ensure that unexpected dirty lines tagged SECURE are not naturally evicted)}. Fields:

REGIME unsigned int

Current translation regime.

SIDE unsigned int

0 d-side cache or unified cache, 1 i-side.

ArchMsg.Warning.dcmvac_matches_watchpoint

TAGS **UNPREDICTABLE** **DISPLAY** Watchpoints matching an AArch32 DCIMVA are implementation defined.

ArchMsg.Warning.decode_fieldmismatch

TAGS **UNPREDICTABLE** **DISPLAY** Instruction is unpredictable with different fields %{FIELD1} and %{FIELD2}. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_fracbitsnegative

TAGS **UNPREDICTABLE** **DISPLAY** VCVT Instruction is unpredictable with a negative number of fraction bits.

ArchMsg.Warning.decode_initblock

TAGS **UNPREDICTABLE** **DISPLAY** Instruction is unpredictable in an IT block.

ArchMsg.Warning.decode_initblocknotlast

TAGS **UNPREDICTABLE** **DISPLAY** Instruction is unpredictable in an IT block when not the last.

ArchMsg.Warning.decode_invalid_condition

TAGS **UNPREDICTABLE** **DISPLAY** Instruction is unpredictable when the condition is not AL.

ArchMsg.Warning.decode_invalidfieldcombination

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with the values in %{FIELD1} and %{FIELD2}. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_invalidvalue

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with the value in %{FIELD}. Fields:

FIELD enum

Field name.

ArchMsg.Warning.decode_mem_hint_unallocated

TAGS **UNPREDICTABLE** DISPLAY There is no memory hint allocated to this bit pattern.

ArchMsg.Warning.decode_registermatch

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with %{FIELD1} and %{FIELD2} as the same register. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_registermismatch

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with %{FIELD1} and %{FIELD2} as different registers. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_registeroutofrange

TAGS **UNPREDICTABLE** DISPLAY Floating-point instruction is unpredictable indexing beyond the end of the register bank.

ArchMsg.Warning.decode_sbzsbo

TAGS **UNPREDICTABLE** DISPLAY Reserved bits in the instruction are not canonical.

ArchMsg.Warning.decode_transactiontoobig

TAGS **UNPREDICTABLE** DISPLAY Attempt to perform a memory transaction of more than 128 bytes.

ArchMsg.Warning.decode_unpred_other

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable.

ArchMsg.Warning.decode_unpreduseofpc

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable when %{FIELD} is PC. Fields:

FIELD enum

Field name.

ArchMsg.Warning.decode_unpreduseofr13

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable when %{FIELD} is R13. Fields:

FIELD enum

Field name.

ArchMsg.Warning.decode_unpreduseofr13inreglist

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with R13 in register-list.

ArchMsg.Warning.decode_useofd16orhigher

TAGS **UNPREDICTABLE** DISPLAY Attempt to use register D16 or higher where it is not allowed.

ArchMsg.Warning.decode_writebackandbaseinlist

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with write back when the base register is in the transfer list.

ArchMsg.Warning.decode_zeroregistersinlist

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable when no registers are to be transferred.

ArchMsg.Warning.recursive_branch

An instruction is performing a branch that targets the same instruction. If this was intended then a WFI might be more effective DISPLAY recursive branch detected at PC=%{PC}. PRIMARY KEY PC. Fields:

PC unsigned int

Address of instruction causing the branch.

ArchMsg.Warning.recursive_exception

An instruction has generated an exception that targets the same instruction. DISPLAY recursive %{TYPE} exception detected at PC=%{PC} (CPSR=%{CPSR}). PRIMARY KEY PC. Fields:

CPSR unsigned int

Processor state.

PC unsigned int

Address of instruction causing the exception.

TYPE enum

Type of exception.

ArchMsg.Warning.reentrant_vector_catch

TAGS **UNPREDICTABLE** DISPLAY Vector catch debug event on Prefetch or Data abort vector %{ADDR}. Fields:

ADDR unsigned int
vector address.

ArchMsg.Warning.reserved_it_state

TAGS **UNPREDICTABLE** DISPLAY Execution with reserved IT state %{ITSTATE}. Fields:

ITSTATE unsigned int
reserved IT state.

ArchMsg.Warning.tlb_contents_unknown

Translation has been enabled without invalidating the TLB. DISPLAY Translation was enabled in the %{PAS:(Secure|Non-secure|Root|Realm)} EL%{EL:d} translation regime (vmid=%{VMID:x}) without invalidating %{INVALIDITY:(|d-side |i-side |)}entries since power-on: %{INVALIDITY:(|D||I)}TLB could contain **UNKNOWN** content. Fields:

EL unsigned int
Exception level of translation regime.

INVALIDITY unsigned int
Which TLB may be invalid (if TLBs are separate).

NON_SECURE bool
Whether the translation regime is non-secure.

PAS unsigned int
Physical address space of translation regime.

VMID unsigned int
Current VMID, if applicable.

ArchMsg.Warning.unknown_DLR_DSPSR

Exiting debug state with unknown DLR or DSPSR.

ArchMsg.Warning.unknown_ELR_SPSR

Returning from debug exception with unknown ELR or SPSR.

ArchMsg.Warning.unpred_FAR_on_LRCPC3_non_atomic_mem_access

TAGS **UNPREDICTABLE** Unpredictable FAR/FSTATUS on LRCPC3 pair instructions resulting in two separate memory accessesDISPLAY Unpredictable FAR on LRCPC3 non atomic memory access at %{ADDR1} with FSTATUS %{FSTATUS1} or %{ADDR2} with FSTATUS %{FSTATUS2}. Fields:

ADDR1 unsigned int
Possible FAR address 1.

ADDR2 unsigned int
Possible FAR address 2.

FSTATUS1 unsigned int
Possible fault status value 1.

FSTATUS2 unsigned int
Possible fault status value 2.

ArchMsg.Warning.unpred_async_tag_check_fail_on_sve_speculative_load_do_update_ffr

Unpredictable set to FALSE of the FFR element associated to SVE speculative load with an async tag check fail TAGS **UNPREDICTABLE**.

ArchMsg.Warning.unpredictable_a32_breakpoint

TAGS **UNPREDICTABLE** DISPLAY Unpredictable breakpoint on a A32 instruction at %{ADDR1} Offset %{OFFSET} BAS %{BAS} for a %{IS_ADDRESS_MISMATCH_BREAKPOINT:(Mismatch|Match)} Breakpoint number %{N:d} Breakpoint address %{ADDR2}. Fields:

ADDR1 unsigned int

Address.

ADDR2 unsigned int

Breakpoint address.

BAS unsigned int

BAS field.

IS_ADDRESS_MISMATCH_BREAKPOINT bool

Is a Mismatch breakpoint, not a Match breakpoint.

N unsigned int

Breakpoint number.

OFFSET unsigned int

Always has a value of zero.

ArchMsg.Warning.unpredictable_banked_register_access

TAGS **UNPREDICTABLE** DISPLAY Access to banked register (R=%{R:d} SYSm=%{SYSM:2x}) is unpredictable. Fields:

R unsigned int

R bit.

SYSM unsigned int

SYSm.

ArchMsg.Warning.unpredictable_ccfail_undef

TAGS **UNPREDICTABLE** DISPLAY An instruction %{OPCODE} at VA=%{VADDR} failed its condition codes check but would otherwise have caused a non-data-dependent trap or undefined exception. Behaviour is implementation-defined choice of exception or nop. Fields:

OPCODE unsigned int

opcode of the instruction.

VADDR unsigned int

Virtual address of the instruction being executed.

ArchMsg.Warning.unpredictable_ici_bits

TAGS **UNPREDICTABLE** DISPLAY ICI bits do not apply or are architecturally unpredictable for instruction.

ArchMsg.Warning.unpredictable_t32_breakpoint

TAGS **UNPREDICTABLE** DISPLAY Unpredictable breakpoint on a T32 instruction at %{ADDR1} Offset %{OFFSET} BAS %{BAS} for a %{IS_ADDRESS_MISMATCH_BREAKPOINT:(Mismatch|Match)} Breakpoint number %{N:d} Breakpoint address %{ADDR2}. Fields:

ADDR1 unsigned int

Address.

ADDR2 unsigned int

Breakpoint address.

BAS unsigned int

BAS field.

IS_ADDRESS_MISMATCH_BREAKPOINT bool

Is a Mismatch breakpoint, not a Match breakpoint.

N unsigned int

Breakpoint number.

OFFSET unsigned int

Indicates which halfword of a 32-bit Thumb instruction the breakpoint is set on. 0 for the first halfword or if the instruction is 16 bits wide, or 1 for the second halfword.

ArchMsg.Warning.unpredictable_t32_vector_catch

TAGS **UNPREDICTABLE** DISPLAY Unpredictable Vector catch exception on a T32 instruction at %{ADDR} Offset %{OFFSET}. Fields:

ADDR unsigned int

Address.

OFFSET unsigned int

Indicates which halfword of a 32-bit Thumb instruction the vector catch exception is set on. 0 for the first halfword or if the instruction is 16 bits wide, or 1 for the second halfword.

ArchMsg.Warning.unpredictable_watchpoint_far

TAGS **UNPREDICTABLE** DISPLAY The value for FAR/EDWAR for hitting this watchpoint may be anywhere in the range %{LOWER_BOUND} to %{UPPER_BOUND}. Fields:

LOWER_BOUND unsigned int

the lowest address accessed by the instruction that triggered the watchpoint.

UPPER_BOUND unsigned int

the highest watchpointed address accessed by that instruction.

ArchMsg.Warning.unpredictable_watchpoint_far_isscmp

DISPLAY The value for FAR/EDWAR for hitting this watchpoint may be anywhere in the range %{LOWER_BOUND} to %{UPPER_BOUND}. Fields:

LOWER_BOUND unsigned int

the lowest address accessed by the instruction that triggered the watchpoint.

UPPER_BOUND unsigned int

the highest watchpointed address accessed by that instruction.

ArchMsg.Warning.warning_AdvSIMDExpandImmUnexpectedZero

TAGS **UNPREDICTABLE** DISPLAY AdvSIMDExpandImm may treat this immediate value as **UNPREDICTABLE**.

ArchMsg.Warning.warning_ConditionalSMC

TAGS **UNPREDICTABLE** DISPLAY SMC instruction has **UNPREDICTABLE** effects when conditional and when combined with traps.

ArchMsg.Warning.warning_TLBID_out_of_range

TLBID out of range in TLBI according to the configuration of NIS, NVIS, NOS and NVOS.

Fields:

NIS unsigned int

NIS.

NOS unsigned int

NOS.

NVIS unsigned int

NVIS.

NVOS unsigned int

NVOS.

TLBID unsigned int

The TLBID value used in the TLBI instruction.

ArchMsg.Warning.warning_TTL_or_TG_mismatches_current_translation_granule

TTL or TG mismatches current translation granule in TLBI VA/RVA/IPA/RIPA. Fields:

CURRENT_TG unsigned int

Current TG as indicated by TCR_ELx or VTCR_EL2.

TLBI_TG unsigned int

TG (or TTL[3:2]) used in TLBI instruction.

ArchMsg.Warning.warning_access_crosses_page_boundary

TAGS **UNPREDICTABLE** DISPLAY Access crosses page boundary.

ArchMsg.Warning.warning_access_crosses_page_boundary_has_fault

TAGS **UNPREDICTABLE** DISPLAY Access that crosses page boundary has fault(s).

ArchMsg.Warning.warning_access_crosses_page_boundary_has_fault_on_both_pages

TAGS **UNPREDICTABLE** DISPLAY Access that crosses page boundary has a fault on each page. Fields:

ADDR1 unsigned int

FAR address on first page.

ADDR2 unsigned int

FAR address on second page.

FSTATUS1 unsigned int

Fault status on first page.

FSTATUS2 unsigned int

Fault status on second page.

ArchMsg.Warning.warning_access_crosses_page_boundary_spanning_different_memory

TAGS **UNPREDICTABLE** DISPLAY Access at address %{ADDR} crosses page from MEM_TYPE %{MEMTYPE_PAGE1} to %{MEMTYPE_PAGE2}. Fields:

ADDR unsigned int

address of access crossing page.

MEMTYPE_PAGE1 signed int

First Page's memory type (0-Normal, 1-Non-Normal).

MEMTYPE_PAGE2 signed int

Second Page's memory type (0-Normal, 1-Non-Normal).

ArchMsg.Warning.warning_access_wraps_around_memory

TAGS **UNPREDICTABLE** DISPLAY Access wraps around memory in %{WIDTH:d}bit mode. Fields:

WIDTH unsigned int

Address width.

ArchMsg.Warning.warning_bcr_linking_status

TAGS **UNPREDICTABLE** DISPLAY DBGBCR.{BT2,BT} combination gives unpredictable behaviour.

ArchMsg.Warning.warning_bcr_mask_reserved

TAGS **UNPREDICTABLE** DISPLAY DBGBCR_ELn.MASK is set to a reserved value.

ArchMsg.Warning.warning_bvr_masked_bit_not_zero

TAGS **UNPREDICTABLE** DISPLAY DBGBVRn_EL1.VA contains a bit masked by DBGBCRn_EL1.MASK that is not zero.

ArchMsg.Warning.warning_ccsidr_unimplemented_level

TAGS **UNPREDICTABLE** DISPLAY CCSIDR read while CSSELR %{CSSELR:x} points to an unimplemented cache level. Fields:

CSSELR unsigned int

current effective value of CSSELR.

ArchMsg.Warning.warning_change_to_ns_when_tge_set

TAGS **UNPREDICTABLE** DISPLAY Attempting to change NS to 1 when HCR.TGE = 1.

ArchMsg.Warning.warning_contiguous_bit_check_abort

TAGS **UNPREDICTABLE** DISPLAY An abort occurred whilst attempting to check TLB entry is contiguous at %{ENTRY_ADDR}. Fields:

ENTRY_ADDR unsigned int

address of conflicting TLB entry.

ArchMsg.Warning.warning_contiguous_bit_error

TAGS **UNPREDICTABLE** DISPLAY TLB entry at %{ENTRY_ADDR} was not contiguous with entry at %{CONTIG_ADDR} Contents are %{ENTRY_DATA} but expected %{CONTIG_DATA}. Fields:

CONTIG_ADDR unsigned int

address of the TLB first read, with which this entry is expected to be contiguous.

CONTIG_DATA unsigned int

expected contents based on the entry at CONTIG_ADDR.

ENTRY_ADDR unsigned int

address of conflicting TLB entry.

ENTRY_DATA unsigned int

contents of conflicting TLB entry.

ArchMsg.Warning.warning_contiguous_bit_error_D128

TAGS **UNPREDICTABLE** DISPLAY TLB entry at %{ENTRY_ADDR} was not contiguous with entry at %{CONTIG_ADDR} Contents are %{ENTRY_DATA_HI} %{ENTRY_DATA_LO} but expected %{CONTIG_DATA}. Fields:

CONTIG_ADDR unsigned int

address of the TLB first read, with which this entry is expected to be contiguous.

CONTIG_DATA unsigned int

expected contents based on the entry at CONTIG_ADDR.

ENTRY_ADDR unsigned int

address of conflicting TLB entry.

ENTRY_DATA_HI unsigned int

Higher bit contents of conflicting TLB entry.

ENTRY_DATA_LO unsigned int

Lower bit contents of conflicting TLB entry.

ArchMsg.Warning.warning_cp10_cp11_mismatch

TAGS **UNPREDICTABLE** DISPLAY Unpredictable configuration of CP10 and CP11 controls at EL %{EL:d}. Fields:

EL unsigned int

EL owning the control.

ArchMsg.Warning.warning_cp10_cp11_reserved_value

TAGS **UNPREDICTABLE** DISPLAY Unpredictable configuration of CPACR CP10 ({CP10}) or CP11 ({CP11}). Fields:

CP10 unsigned int

CP10 access.

CP11 unsigned int

CP11 access.

ArchMsg.Warning.warning_csselr_level_out_of_range

TAGS **UNPREDICTABLE** DISPLAY CSSELR.Level out of range, written value {LEVEL} with only {IMPLEMENTED:d} levels implemented. Fields:

IMPLEMENTED unsigned int

number of cache levels implemented.

LEVEL unsigned int

written value of CSSELR.Level.

TYPE enum

The type of cache selected by CSSELR.InD and TnD.

ArchMsg.Warning.warning_debug_flow_control_bits_not_obeyed

TAGS **UNPREDICTABLE** DISPLAY Flow control bits for {REGISTER} were not obeyed, causing {ERROR}. Fields:

ERROR enum

type of error resulting.

REGISTER enum

register under consideration.

ArchMsg.Warning.warning_debug_register_access_during_reset

TAGS **UNPREDICTABLE** DISPLAY Unpredictable {IS_WRITE:(read|write)} access to debug register offset {OFFSET} during reset. Fields:

IS_WRITE unsigned int

Write Not Read.

OFFSET unsigned int

Register Offset.

ArchMsg.Warning.warning_decode_cps_inconsistent_fields

TAGS **UNPREDICTABLE** DISPLAY The mode setting fields in the CPS instruction are inconsistent.

ArchMsg.Warning.warning_decode_invalid_state

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable in current exception-level/security state.

ArchMsg.Warning.warning_default_cacheable_mmu_on

TAGS **UNPREDICTABLE** DISPLAY Default cacheable is enabled (HCR.DC = 1) while MMU is enabled.

ArchMsg.Warning.warning_default_cacheable_vmmu_off

TAGS **UNPREDICTABLE** DISPLAY Default cacheable is enabled (HCR.DC = 1) while HCR.VM = 0.

ArchMsg.Warning.warning_deprecated_wvr_bit_2_set

TAGS **UNPREDICTABLE** DISPLAY DBGWVR%{N:d}_EL1=%{WVR:h} has bit 2 set, deprecated in ARMv8. Fields:

N unsigned int

index.

WVR unsigned int

DBGWVR.

ArchMsg.Warning.warning_eret_while_software_step_active_pending

Missing ISB between setting MDSCR_EL1.SS and ERET TAGS **UNPREDICTABLE**.

ArchMsg.Warning.warning_exclusive_to_non_normal

TAGS **UNPREDICTABLE** DISPLAY Exclusive Access to Non-Normal Memory %{ADDR}. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_exclusive_to_non_writeback

TAGS **UNPREDICTABLE** DISPLAY Exclusive Access to Non-Writeback-Cacheable Memory %{ADDR}. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_execute_from_device_memory

TAGS **UNPREDICTABLE** DISPLAY An attempt was made to execute from Device memory.

ArchMsg.Warning.warning_illegal_cpsr_mode

TAGS **UNPREDICTABLE** DISPLAY Writing an illegal or unimplement mode (%{NEW_MODE:h}) to CPSR was unpredictable. Fields:

NEW_MODE unsigned int

New value of CPSR.M.

ArchMsg.Warning.warning_illegal_srs_mode

TAGS **UNPREDICTABLE** DISPLAY Illegal or **UNPREDICTABLE** mode (%{MODE:h}) used for SRS instruction. Fields:

MODE unsigned int

mode for Banked SP.

ArchMsg.Warning.warning_implementation_defined_read_debug_register_in_SCS

Read Access to Debug registers from the processor is IMPLEMENTATION DEFINED TAGS IMP_DEF. Fields:

PPB_OFFSET unsigned int
debug register offset.

ArchMsg.Warning.warning_implementation_defined_sequential_security_transitions_supported

The behavior of sequential instruction fetches that cross from non-secure to secure memory and contain an SG instruction is CONSTRAINED UNPREDICTABLE TAGS UNPREDICTABLE.

ArchMsg.Warning.warning_implementation_defined_stack_limit_check_supported

It is IMPLEMENTATION DEFINED whether stack pointer limit checking is performed for this instruction TAGS IMP_DEF. Fields:

ADDRESS unsigned int
Address of instruction.

OPCODE unsigned int
instruction opcode.

ArchMsg.Warning.warning_implementation_defined_taggednonsharedmemorytype

TAGS IMP_DEF DISPLAY Memory attribute has Tagged, non shared memory type; it is IMPLEMENTATION DEFINED whether the memory location is treated as Tagged or non-Tagged.

ArchMsg.Warning.warning_implementation_defined_write_debug_register_in_SCS

Write Access to Debug registers from the processor is IMPLEMENTATION DEFINED TAGS IMP_DEF. Fields:

DATA unsigned int
data attempted to be written.

PPB_OFFSET unsigned int
debug register offset.

ArchMsg.Warning.warning_invalid_tcr_granule

TAGS UNPREDICTABLE DISPLAY TCR.TG%{TG_ID:(0|1)} bits have been set to a granule size %{{REQUEST:d}}K not implemented - using %{{SUBSTITUTE:d}}K. Fields:

REQUEST signed int
page size requested (or 0 for reserved).

SUBSTITUTE signed int
best guess available page size.

TG_ID bool
bits TG0 or TG1.

ArchMsg.Warning.warning_load_multiple_user_registers_from_user_mode

TAGS UNPREDICTABLE DISPLAY An LDM(user registers) instruction executed from user mode.

ArchMsg.Warning.warning_mte_in_mem_attr_not_supported

TAGS UNPREDICTABLE DISPLAY Memory attribute %{ATTR} has Tagged memory type but Armv8.5-MTE is not implemented. Fields:

ATTR unsigned int

Memory attribute.

ArchMsg.Warning.warning_reserved_breakpoint_state_match

TAGS UNPREDICTABLE DISPLAY %{IS_BREAKPOINT:(Watchpoint|Breakpoint)} programmed with a reserved combination of HMC, SSC and %{IS_BREAKPOINT:(PAC|PMC)}. Fields:

IS_BREAKPOINT unsigned int

Is a breakpoint, not a watchpoint.

ArchMsg.Warning.warning_shareability

TAGS UNPREDICTABLE DISPLAY Unpredictable: combination of the 1st and 2nd stages of translation is Normal Inner Non-Cacheable, Outer Non-Cacheable.

ArchMsg.Warning.warning_software_step_set_while_enabled

TAGS UNPREDICTABLE DISPLAY MDSCR_EL1.SS set to 1 while software step debug exceptions are enabled.

ArchMsg.Warning.warning_thumb_instruction_wraps_around_memory

TAGS UNPREDICTABLE DISPLAY Thumb instruction wraps around memory.

ArchMsg.Warning.warning_ttbr_sbz_bits_are_not_zero

TAGS UNPREDICTABLE DISPLAY %{TTBR}[%{MSB:d}:0] should be zero, but are not. Fields:

MSB unsigned int

x-1.

TTBR enum

which TTBR register.

ArchMsg.Warning.warning_unaligned_address_dbgdtrrx_write

TAGS UNPREDICTABLE DISPLAY Unaligned address %{ADDR} in X0/R0 when DBGDTRRX written in memory access mode. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unaligned_address_dbgdtrtx_read

TAGS UNPREDICTABLE DISPLAY Unaligned address %{ADDR} in X0/R0 when DBGDTRTX read in memory access mode. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unaligned_to_device

TAGS UNPREDICTABLE DISPLAY Unaligned Access to Device Memory %{ADDR}. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unaligned_to_strongly_ordered

TAGS **UNPREDICTABLE** DISPLAY Unaligned Access to Strongly Ordered Memory %{ADDR}.

Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unknown_sau_rnr

TAGS **UNPREDICTABLE** DISPLAY SAU_RNR was set to an unsupported value.

ArchMsg.Warning.warning_unpredictable_AIRCR_PRIS_and_BFHFNMINs

TAGS **UNPREDICTABLE** DISPLAY The effect of setting both AIRCR.BFHFNMINs and AIRCR.PRIS to 1 is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_AIRCR_VECTCLRACTIVE_when_not_in_debug

TAGS **UNPREDICTABLE** DISPLAY The effect of writing a 1 to AIRCR.VECTCLRACTIVE if the processor is not halted in Debug state is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_AIRCR_VECTRESET_and_SYSRESETREQ

TAGS **UNPREDICTABLE** DISPLAY When the processor is halted in Debug state, if a write to the register writes a 1 to both VECTRESET and SYSRESETREQ, the behavior is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_AIRCR_VECTRESET_when_not_in_debug

TAGS **UNPREDICTABLE** DISPLAY The effect of writing a 1 to AIRCR.VECTRESET if the processor is not halted in Debug state is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_AIRCR_incorrect_VKEY

TAGS **UNPREDICTABLE** DISPLAY The value 0x05FA must be written to AIRCR.VKEY, otherwise the register write is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_EXC_RETURN_Reserved_Bit

EXC_RETURN[23:7] are reserved with the special condition that all bits should be written as one. Values other than all 1s are **UNPREDICTABLE**. TAGS **UNPREDICTABLE**. Fields:

EXC_RETURN unsigned int

Special PC value.

SBO_MASK unsigned int

Which bits Should Be One to not UNPRED.

ArchMsg.Warning.warning_unpredictable_PMEVTYPEPERn_ELO_VS_reserved_value

TAGS **UNPREDICTABLE** DISPLAY Unpredictable reserved value of PMEVTYPER<n>_ELO.VS.

ArchMsg.Warning.warning_unpredictable_change_to_DEMCR_MON_STEP_at_insufficient_priority

TAGS **UNPREDICTABLE** DISPLAY The effect of changing DEMCR.MON_STEP at an execution priority that is lower than the priority of the DebugMonitor exception is **CONSTRAINED UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_change_to_priority_of_active_exception

TAGS UNPREDICTABLE DISPLAY Changing the priority of an active exception is UNPREDICTABLE.

ArchMsg.Warning.warning_unpredictable_clear_lspact

TAGS UNPREDICTABLE DISPLAY UNPREDICTABLE state on context switch. Fields:

DRVEXC_TAKEN unsigned int

Derived exception taken, not pended.

REACHED_LAST_STORE unsigned int

Only the last faulted.

ArchMsg.Warning.warning_unpredictable_dbg_exit_unaligned_dlr

TAGS UNPREDICTABLE DISPLAY Debug exit to AArch32 with DLR[0] set to 1. PC[0] can be set to 0 or to DLR[0].

ArchMsg.Warning.warning_unpredictable_exception_catch

TAGS UNPREDICTABLE DISPLAY Generation of an exception catch event is unpredictable.

ArchMsg.Warning.warning_unpredictable_exception_return_inconsistent_state

TAGS UNPREDICTABLE DISPLAY state on exception return is unpredictable.

ArchMsg.Warning.warning_unpredictable_exception_return_instruction

TAGS UNPREDICTABLE DISPLAY exception return instruction is unpredictable.

ArchMsg.Warning.warning_unpredictable_in_debug_state

TAGS UNPREDICTABLE DISPLAY Instruction is UNPREDICTABLE when executed in debug state.

ArchMsg.Warning.warning_unpredictable_mair_encoding

TAGS UNPREDICTABLE DISPLAY Memory attribute %{ATTR} has invalid MAIR encoding; this may lead to unpredictable results. Fields:

ATTR unsigned int

Memory attribute.

ArchMsg.Warning.warning_unpredictable_pmu_counter_access

TAGS UNPREDICTABLE DISPLAY Access to PMU counters from non-secure EL0 or EL1 is UNPREDICTABLE with MDCR_EL2.HPMN set to 0.

ArchMsg.Warning.warning_unpredictable_prioritization_breakpoint_match_vector_catch

TAGS UNPREDICTABLE DISPLAY Two events with the same priority occurred at the same instruction: Address Matching Vector Catch debug event, and Breakpoint debug event. It is constrained unpredictable which is taken.

ArchMsg.Warning.warning_unpredictable_stack_selection

TAGS UNPREDICTABLE DISPLAY UNPREDICTABLE state on context switch.

ArchMsg.Warning.warning_unpredictable_tsize_out_of_range

TAGS UNPREDICTABLE DISPLAY %{TCR}.T%{TTBR:d}SZ (%{TSIZE:d} bits) is out of range. Must be between 25 and %{TSIZE_MAX:d}. Fields:

TCR enum

Which TCR.

TSize signed int

Value of TSize.

TSize_MAX signed int

The maximum allowed value.

TTBR unsigned int

Which TTBR region.

ArchMsg.Warning.warning_unpredictable_unaligned_pc_as_base_register

TAGS **UNPREDICTABLE** DISPLAY The PC value must be word-aligned, otherwise the behavior of the instruction is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_unaligned_pop_stack

TAGS **UNPREDICTABLE** DISPLAY **CONSTRAINED UNPREDICTABLE** unaligned stack frame `%{FRAME_ADDR:x}` in exception-return. Fields:

FRAME_ADDR unsigned int

unaligned stack frame address.

ArchMsg.Warning.warning_unpredictable_unaligned_to_non_writeback

TAGS **UNPREDICTABLE** DISPLAY Unaligned Access to Non-Writeback-Cacheable Memory.

ArchMsg.Warning.warning_unpredictable_vmsa_memattrib

TAGS **UNPREDICTABLE** DISPLAY Unpredictable vmsa memory attribute with texcb value programmed to `%{TEXCB}` when tex remap is `%{TEX_REMAP}`. Fields:

TEXCB unsigned int

texcb value.

TEX_REMAP bool

use tex remap.

ArchMsg.Warning.warning_unpredictable_vtcr_t0sz_sl0

TAGS **UNPREDICTABLE** DISPLAY The combination of VTCR.TOSZ=`%{TOSZ:d}` and VTCR.SL0=`%{SL0:d}` is unpredictable. Fields:

SL0 signed int

VTCR.SL0, starting level for stage 2 translation table walks.

TOSZ signed int

VTCR.TOSZ, required input address range.

ArchMsg.Warning.warning_unpredictable_write_DHCSR

TAGS **UNPREDICTABLE** DISPLAY **UNPREDICTABLE** set of bit-changes in DHCSR. Fields:

NEW unsigned int

data attempted to be written.

OLD unsigned int

previous value.

ArchMsg.Warning.warning_unpredictable_write_SHCSR

TAGS UNPREDICTABLE DISPLAY UNPREDICTABLE setting a pending bit to 1 for an exception with priority \geq execution priority.

ArchMsg.Warning.warning_unsupported_access_to_memory_mapped_register

TAGS UNPREDICTABLE DISPLAY Access to memory mapped register is unsupported.

ArchMsg.Warning.warning_user_jmcr_access

TAGS UNPREDICTABLE DISPLAY User %{WRITE:(write|read)} access to JMCR is unpredictable. Fields:

WRITE unsigned int
access is write.

ArchMsg.Warning.warning_wcr_mask_and_bas

TAGS UNPREDICTABLE DISPLAY DBGWCR_EL1.MASK is non-zero and BAS is not 0xff.

ArchMsg.Warning.warning_wcr_mask_reserved

TAGS UNPREDICTABLE DISPLAY DBGWCR_ELn.MASK is set to a reserved value.

ArchMsg.Warning.warning_wcr_non_configuous_bas

TAGS UNPREDICTABLE DISPLAY DBGWCR_EL1.BAS has non-contiguous set of ones.

ArchMsg.Warning.warning_wvr_masked_bit_not_zero

TAGS UNPREDICTABLE DISPLAY DBGWVRn_EL1.VA contains a bit masked by DBGWCRn_EL1.MASK that is not zero.

ArchMsg.Why.why_branch_target_exception

A branch target exception has occurred. DISPLAY branch target exception: target pc=%{PC}, target opcode=%{OPCODE}. PRIMARY KEY PC. Fields:

BTYPE unsigned int
pstate.BTYPE at time branch taken.

OPCODE unsigned int
opcode at branch target.

PC unsigned int
Virtual Address of branch target.

ArchMsg.Why.why_illegal_state

An illegal mode change has occurred. DISPLAY illegal mode change: %{MESSAGE}. Fields:

MESSAGE string
The message.

BRANCH_MISPREDICT

Simulating branch mispredict. Fields:

PC unsigned int
Origin address (or 0 if unavailable).

BRA_DIR

Direct branches, to immediate address. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISET enum

The instruction set of the branch instruction.

IS_COND bool

Indicates if this is a conditional branch.

IS_HINTED bool

Whether the conditional instruction is hinted.

IS_ISB bool

Whether the instruction is ISB.

IS_LINK bool

Indicates if this is a branch with link.

PC unsigned int

The address of the branch instruction.

TARGET_ISET enum

The instruction set after the branch.

TARGET_PC unsigned int

The address the instruction branches to.

BRA_INDIR

Indirect branches, perhaps to a register. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISET enum

The instruction set of the branch instruction.

IS_COND bool

Indicates if this is a conditional branch.

IS_HINTED bool

Whether the conditional instruction is hinted.

IS_ISB bool

Whether the instruction is ISB.

IS_LINK bool

Indicates if this is a branch with link.

PC unsigned int

The address of the branch instruction.

TARGET_ISET enum

The instruction set after the branch.

TARGET_PC unsigned int

The address the instruction branches to.

BRA_INDIR_CCFAIL

Indirect branches fail, perhaps to a register.

BRA_RET_CCFAIL

Branch return fail, perhaps to a register.

CACHE_MAINTENANCE_OP

Cache Maintenance Operation. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

DATA unsigned int

Specified MVA or Set/way.

FUNCTION enum

Clean / Invalidate.

SCOPE enum

Affected region.

SIDE enum

Inst / Data.

CCFAIL

Conditional instruction condition check fail. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CCFAIL_UNC

Unconditional instruction condition check fail. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CCPASS

Conditional instruction condition check pass. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CCPASS_UNC

Unconditional instruction condition check pass. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CODE_CACHE_MAINT

Code cache maintenance. Fields:

END_INCL unsigned int

Inclusive end address for VA or PA requests.

NS bool

Non-secure world for PA.

START unsigned int

Start address for VA or PA requests.

TYPE enum

Request type.

COMPILE_BLOCK_END

Last instruction of basic block translated. Fields:

VADDR unsigned int

Address of next instruction after this basic block.

COMPILE_BLOCK_START

First instruction of basic block translated. Fields:

VADDR unsigned int

Address of first instruction in basic block.

COMPILE_INST

ARM instruction compiled. Fields:

DISASS string

Disassembly of the instruction.

ISET enum

The instruction set of this instruction.

ITSTATE unsigned int

The ITSTATE current for the instruction.

OPCODE unsigned int

The opcode of the instruction.

PC unsigned int

The address of the instruction.

SIZE unsigned int

The size of the instruction in bytes.

CONTEXTIDR

Context ID Register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

NS bool

Secure or nonsecure banked register is accessed.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The new value written.

CONTEXT_SYNC

Called for every context synchronization event. Fields:

ARCH_CSE bool

Architectural Context Synchronization Event.

PC unsigned int

The address of the synchronization instruction.

SYSREG_SYNC bool

System register synchronization event.

CORE_ENDIAN

Core BE8 Big-Endian state changed. Fields:

BE8 bool

Core BE8 Big-Endian state.

CORE_INFO

Static processor attributes. Only triggered by a call to DumpState(). Fields:

ARCH_PROFILE enum

The architecture profile of the core.

CLUSTER_ID unsigned int

The cluster ID of this processor.

CORE_NUM unsigned int

The number of this core in an MP processor.

DETAILED_CORE_FEATURES enum

Detailed core features as a bitfield. Enum values are bitmasks to extract the info from the field.

FPU_VERSION enum

The VFP version implemented by the core.

MEM_ARCH enum

The memory architecture of the core.

MIDR_EL1 unsigned int

MIDR_EL1 value.

NUM_CORES unsigned int

The number of cores in this MP processor.

QUANTUM_SIZE unsigned int

The default quantum size of the core (in ticks).

SECURITY_FEATURES bool

Does the core have security features?.

CORE_LOADS

Processor load accesses. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ACQREL enum

Is this an acquire/release.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DATA unsigned int

The data read or written.

ELEMENT_SIZE unsigned int

Width of each element.

ENDIANNESS enum

Byte ordering of this access.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

LOCK enum

Normal, exclusive or locked access.

MECID unsigned int

Memory Encryption Context ID, if applicable.

MECID2 unsigned int

Memory Encryption Context ID of the second page, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PAGE_SIZE unsigned int

Page size as $\log_2(\text{size in bytes})$.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK, 2=Exclusive Failed, 3=Unsupported.

SIZE unsigned int

Width of the access in bytes. Only required if DATA is not traced.

TRANS bool

Is this a translated access.

VADDR unsigned int

The virtual address of the access.

CORE_REGS

Changes of the core registers R0 to R14. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number, 0 to 14.

MODE enum

Bank of the register accessed.

OLD_VALUE unsigned int

The old value overwritten.

PHYS_ID enum

The physical register accessed.

VALUE unsigned int

The new value written to the register.

CORE_REGS64

Changes of the core registers X0..X30, SP_ELn. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ID enum

The register number, 0-30 for X0-X30, >=32 for SP_ELn.

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

CORE_REGS64_READ

General purpose AArch64 register has been read X0..X30, SP_ELn. Fields:

ID enum

The register number, 0-30 for X0-X30, >=32 for SP_ELn.

VALUE unsigned int

The value read from the register.

CORE_STALLED_TRANSITION

Detect when a core is prevented from progressing PChannel states. Fields:

CURRENT_STATE enum

Current power mode of core.

REASON string

The reason a transition was stalled, and what must be done to unblock it.

TARGET_STATE enum

Desired power mode of core after transition.

CORE_STORES

Processor store accesses. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ACQREL enum

Is this an acquire/release.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DATA unsigned int

The data read or written.

ELEMENT_SIZE unsigned int

Width of each element.

ENDIANNESS enum

Byte ordering of this access.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

LOCK enum

Normal, exclusive or locked access.

MECID unsigned int

Memory Encryption Context ID, if applicable.

MECID2 unsigned int

Memory Encryption Context ID of the second page, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PAGE_SIZE unsigned int

Page size as $\log_2(\text{size in bytes})$.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK, 2=Exclusive Failed, 3=Unsupported.

SIZE unsigned int

Width of the access in bytes. Only required if DATA is not traced.

TRANS bool

Is this a translated access.

VADDR unsigned int

The virtual address of the access.

CP14_READ

System Coprocessor register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP14 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc1 unsigned int

opcode 1.

opc2 unsigned int

opcode 2.

CP14_WRITE

System Coprocessor register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP14 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc1 unsigned int

opcode 1.

opc2 unsigned int

opcode 2.

CP15_READ

System Coprocessor register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP15 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc1 unsigned int

opcode 1.

opc2 unsigned int

opcode 2.

CP15_READ64

System Coprocessor register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP15 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc unsigned int

opcode 1.

CP15_WRITE

System Control Coprocessor register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP15 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc1 unsigned int

opcode 1.

opc2 unsigned int
opcode 2.

CP15_WRITE64

System Coprocessor register write. Fields:

CORE_NUM unsigned int
Core number in a multi processor.

CRm unsigned int
CRm.

NS bool
Secure or nonsecure banked register is accessed.

REG_NAME string
Name of the CP15 register accessed.

SECURITY_STATE enum
The current security state.

UNDEF bool
The register accessed is undefined.

UPDATED_VALUE unsigned int
Updated value of the register now it has been written.

VALUE unsigned int
The new value written.

opc unsigned int
opcode 1.

CPSR

CPSR change. (In AArch64 PSTATE represented in SPSR format). Fields:

CORE_NUM unsigned int
Core number in a multi processor.

OLD_VALUE unsigned int
The old CPSR value.

UNKNOWN unsigned int
Bits within the register that have unknown value.

VALUE unsigned int
The new CPSR value.

CRYPTO_SPEC

Every crypto instruction speculatively executed.

CorePowerStateChange

Core power-up state has changed. DISPLAY Core powered %{PU:(up|down)}. Fields:

PU bool

Whether the core powered up.

DATA_CACHE_SET_ALLOC_TAGS

Set allocation tags from DC GVA instruction. Fields:

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

NSDESC unsigned int

The security state of the access.

PADDR unsigned int

The physical address of the access.

PAS enum

The physical address space of the page.

RESPONSE enum

0=Aborted, 1=OK.

SIZE unsigned int

The size of memory in bytes.

TAG unsigned int

The value of the tag being set.

TAGGING_TYPE enum

Type of Memory tagging (Physical or Virtual).

TAG_VADDR unsigned int

The virtual address of the Virtual Tag if Virtual tagging is used; otherwise 0.

VADDR unsigned int

Virtual address used for setting allocation tags.

DATA_CACHE_ZERO

Zero and invalidate cache line from DC ZVA instruction. Fields:

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

NSDESC unsigned int

The security state of the access.

PADDR unsigned int

The physical address of the access.

PADDR_TAG unsigned int

If tag checked, the physical address tag for the access.

PAS enum

The physical address space of the page.

RESPONSE enum

0=Aborted, 1=OK.

SIZE unsigned int

The size of zero'd memory in bytes.

TAG_CHECKED bool

Is this access tag checked.

VADDR unsigned int

Virtual address of the zero'd cache line.

DEBUG_EVENT

Hardware debug support event. Fields:

EVENT enum

Description of event.

VALUE unsigned int

data value.

DEBUG_TARGET_ISET

Hardware debug target ISA. Fields:

TARGET_ISET enum

The target instruction set of the debug state.

DMI_ALLOCATE

DMI allocated. Fields:

ACCESS_DIRECTION enum

READ / WRITE.

PADDR unsigned int

The physical address of the DMI.

PTR unsigned int

DMI range pointer.

SIDE enum

DSIDE / ISIDE.

SIZE unsigned int

The size of DMI.

VADDR unsigned int

The virtual address of the DMI..

DMI_HIT

DMI hit, access used DMI (only fire for small DMI regions i.e. less than a full page). Fields:

ACCESS_DIRECTION enum

READ / WRITE.

PADDR unsigned int

The physical address of the DMI.

SIDE enum

DSIDE / ISIDE.

SIZE unsigned int

The size of the DMI.

VADDR unsigned int

The virtual address of the DMI..

DMI_REVOKE

DMI revoked. Fields:

ACCESS_DIRECTION enum

READ / WRITE.

PADDR unsigned int

The physical address of the DMI.

SIDE enum

DSIDE / ISIDE.

SIZE unsigned int

The size of DMI.

VADDR unsigned int

The virtual address of the DMI..

END_COMPILE

Compilation end. Fields:

END_OF_PAGE_COUNT unsigned int

Number of basic blocks exited due to page boundary since START_COMPILE.

FETCHFAIL_COUNT unsigned int

Number of basic blocks exited due to ifetch failure START_COMPILE.

INST_COUNT unsigned int

Number of instructions compiled since START_COMPILE.

NONSEQ_COUNT unsigned int

Number of basic blocks exited due to non-sequential instructions since START_COMPILE.

PAGE_STRADDLE_COUNT unsigned int

Number of basic blocks exited due to unaligned instructions crossing page since START_COMPILE.

ESB

Error Synchronization Barrier hit. Fields:

ASYNC bool

ESB is async.

IMPLICIT bool

ESB is implicit.

EXCEPTION

Exception taken. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ESR unsigned int

The value assigned to ESR when exception is taken.

ESR_EC unsigned int

The Value assigned to ESR.EC when exception is taken.

IS_PHYSICAL bool

Physical or Virtual exception.

IS_TRAP bool

Exception coming from a trap.

LR unsigned int

The value assigned to the link register.

PC unsigned int

The location where the exception occurred.

PREFERRED_RETURN unsigned int

The preferred return address for the exception.

TAKEN_LOCALLY bool

Whether exception is taken locally(same exception level).

TARGET_ISET enum

The instruction set of the exception handler code.

TARGET_PC unsigned int

The address the exception branches to.

VECTOR enum

The exception vector.

VECTOR_OFFSET unsigned int

The offset of an exception vector from the base address.

EXCEPTION_END

Every exception completed.

EXCEPTION_INFO

Trace human-readable reason of the current exception for data/instruction aborts, floating point exceptions and unknown exceptions. Fields:

DESCRIPTION string

Reason for current exception.

ESR_EC unsigned int

The value of ESR.EC when exception is taken.

EXCEPTION_RAISE

Every exception raised.

EXCEPTION_RETURN

Branches on leaving exception. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISSET enum

The instruction set of the branch instruction.

IS_COND bool

Indicates if this is a conditional branch.

IS_HINTED bool

Whether the conditional instruction is hinted.

IS_ISB bool

Whether the instruction is ISB.

IS_LINK bool

Indicates if this is a branch with link.

PC unsigned int

The address of the branch instruction.

TARGET_ISET enum

The instruction set after the branch.

TARGET_PC unsigned int

The address the instruction branches to.

EXCEPTION_RETURN_PREBRANCH

Exception return event that must occur before PC and processor state have updated.

EXCEPTION_START

Every exception started.

FIQ_TAKEN

FIQ taken exception.

FP_EXCEPTIONS_SIGNALLED

Trace which FP exceptions were signalled as a part of the FP operations performed by an instruction. Fields:

FP_EXCEPTIONS_SIGNALLED_MASK unsigned int

Indicates which FP exceptions were signalled by an instruction. This uses the same encoding as the cumulative floating-point exception bitfields in FPSR.

FP_STATE

Floating point state. Fields:

FPSR unsigned int

FPSR register value.

MASK unsigned int

The mask indicating updated cumulative bits.

FREQ_CHANGED

Core frequency changed. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The instruction count of this core.

NEW_FREQ signed int

The new frequency of this core (expressed in Hz).

OLD_FREQ signed int

The old frequency of this core (expressed in Hz).

HLT

HLT instruction occurred. Fields:

IMM16 unsigned int

The 16-bit immediate value encoded in HLT instruction.

INST

Every instruction executed. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DEBUG_STATE bool

The instruction is executed in debug state.

DISASS string

Disassembly of instruction.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISET enum

The current instruction set.

ITSTATE unsigned int

The current ITSTATE.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

MODE enum

The mode the core is in.

NS unsigned int

The core's non-secure bit.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

OPCODE unsigned int

The opcode of the instruction.

PADDR unsigned int

The physical address of the instruction.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the instruction.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the page.

PC unsigned int

The address of the instruction.

SECURITY_STATE enum

The core's security state.

SIZE unsigned int

The size of the instruction in bytes.

INST_END

Every instruction completed.

INST_START

Every instruction started. Fields:

DEBUG_STATE **bool**

The instruction is executed in debug state.

ISET **enum**

The current instruction set.

MODE **enum**

The mode the core is in.

NS **enum**

The current Secure State.

PC **unsigned int**

The address of the conditional instruction.

SECURITY_STATE **enum**

The current Security State.

INST_STRADDLE

Instruction straddles boundary. Fields:

DEBUG_STATE **bool**

The instruction is executed in debug state.

ISET **enum**

The current instruction set.

MODE **enum**

The mode the core is in.

NS **enum**

The current Secure State.

PC **unsigned int**

The address of the conditional instruction.

SECURITY_STATE **enum**

The current Security State.

IRQ_TAKEN

IRQ taken exception.

ISB

ISB instruction occurred. Fields:

PC **unsigned int**

The address of the ISB instruction.

LOAD_MULTIPLE_REGS_LIST

List of destination registers (not source) of a multiple registers load instruction. Fields:

REG string

Register to add into the list.

LOCAL_MONITOR

Local monitor activity. Fields:

PADDR unsigned int

Local Monitor Address.

State enum

State of the monitor (Open/Exclusive).

MAPPING_CREATE

A memory mapping has been created. This is an implementation detail of the model and so may change without notice. Fields:

ASID unsigned int

The ASID used during translation, or 0 if no ASID was used.

IS_VALID bool

Whether the mapping is valid or not.

PADDR unsigned int

Base physical address of region, or 0 if not known.

SIDE enum

I-side or D-side.

SIZE unsigned int

Size of the page, or 0 if not known.

VADDR unsigned int

Base virtual address of region.

VMID unsigned int

The VMID used during translation, or 0 if no VMID was used.

MAPPING_DESTROY

A memory mapping has been destroyed. This is an implementation detail of the model and so may change without notice. Fields:

ASID unsigned int

The ASID used during translation.

CAUSE enum

The cause of invalidation.

PADDR unsigned int

Base physical address of page.

SIDE enum

I-side or D-side.

SIZE **unsigned int**

Size of the page.

VADDR **unsigned int**

Base virtual address of page.

VMID **unsigned int**

The VMID used during translation.

MEMMAP_DEBUG_READ

Memory mapped reads to the debug registers. Fields:

ADDR **unsigned int**

address.

CORE_NUM **unsigned int**

Core number in a multi processor.

EXT **bool**

Whether access is from an external device (such as the DAP).

NS **enum**

Secure state of the access.

REG_NAME **string**

Name of the debug register accessed.

UNDEF **bool**

The register accessed is undefined.

VALUE **unsigned int**

The value read.

MEMMAP_DEBUG_WRITE

Memory mapped writes to the debug registers. Fields:

ADDR **unsigned int**

address.

CORE_NUM **unsigned int**

Core number in a multi processor.

EXT **bool**

Whether access is from an external device (such as the DAP).

NS **enum**

Secure state of the access.

REG_NAME **string**

Name of the debug register accessed.

UNDEF **bool**

The register accessed is undefined.

VALUE unsigned int

The new value written.

MMU_D128_TTB_READ

This event is triggered by reads caused by a D128 translation table walk. Fields:

ABORTED bool

Was the walk successful.

DATA_HIGH unsigned int

The data upper bits read.

DATA_LOW unsigned int

The data lower bits read.

IPA unsigned int

For stage 1, the IPA of the read.

LEVEL unsigned int

Translation table level.

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

OUTADDR unsigned int

The output physical address.

PADDR unsigned int

The physical address of the read.

PASmem enum

Physical Address Space of the accessed memory.

PASreq enum

Physical Address Space of the requested lookup.

REGIME_EL enum

Entry matches in this translation regime.

REQUESTER enum

The requester of this table walk.

SIDE enum

Inst / Data.

STAGE unsigned int

Translation stage.

MMU_D128_TTB_WRITE

This event is triggered by writes caused by a D128 translation table walk. Fields:

ABORTED bool

Was the walk successful.

DATA_HIGH unsigned int

The data upper bits written.

DATA_LOW unsigned int

The data lower bits written.

IPA unsigned int

For stage 1, the IPA of the write.

IS_HACDBS_ACCESS bool

If the access was due to HACDBS_CLEAN.

LEVEL unsigned int

Translation table level.

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

OUTADDR unsigned int

The output physical address.

PADDR unsigned int

The physical address of the write.

PASmem enum

Physical Address Space of the accessed memory.

PASreq enum

Physical Address Space of the requested lookup.

REGIME_EL enum

Entry matches in this translation regime.

SIDE enum

Inst / Data.

STAGE unsigned int

Translation stage.

MMU_TRANS

Address translation information. Fields:

ASID unsigned int

Address space identifier.

CORE_NUM unsigned int

Core number in a multi processor.

Hyp bool

Entry matches in Hyp state only.

INNERCACHE_RA bool

Is the inner cache allocate on read.

INNERCACHE_TRANSIENT bool

Is the inner write-through transient.

INNERCACHE_TYPE enum

Inner Caching scheme (NC/MB/WA).

INNERCACHE_WA bool

Is the inner cache allocate on write.

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MEMTYPE enum

Memory type.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC enum

Is secure side supposed to access secure or nonsecure memory.

OUTERCACHE_RA bool

Is the outer cache allocate on read.

OUTERCACHE_TRANSIENT bool

Is the outer write-through transient.

OUTERCACHE_TYPE enum

Outer Caching scheme (NC/MB/WA).

OUTERCACHE_WA bool

Is the outer cache allocate on write.

PADDR unsigned int

Physical address of the access.

PAGESIZE unsigned int

Page size as log2(size).

PAS enum

PAS supposed to be accessed is secure or nonsecure memory or Root/Realm if RME is implemented.

SH enum

Shareability.

SIDE enum

Inst / Data.

STAGE1_PERM unsigned int

Stage 1 permissions, [5:3] Privileged access (XWR) [2:0] User access (XWR).

STAGE2_PERM unsigned int

Stage 2 permission mask [5:3] Privileged access (XWR) [2:0] User access (XWR).

VADDR unsigned int

Virtual address of the access.

VMID unsigned int

Virtual machine identifier.

XS bool

XS Attribute.

nG enum

Flag indicating whether ASID will be matched.

MMU_TTB_READ

This event is triggered by reads caused by a translation table walk. Fields:

ABORTED bool

Was the walk successful.

DATA unsigned int

The data read.

DESC_KIND enum

The encoding scheme of the descriptors used for this walk.

IPA unsigned int

For stage 1, the IPA of the read.

LEVEL unsigned int

Translation table level.

LPAE bool

Is this for an LPAE translation (*DEPRECATED*, use DESC_KIND).

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSmem bool

Non secure memory access.

NSreq bool

Non secure request.

OUTADDR unsigned int

The output physical address.

PADDR unsigned int

The physical address of the read.

PASmem enum

Physical Address Space of the accessed memory.

PASreq enum

Physical Address Space of the requested lookup.

REGIME_EL enum

Entry matches in this translation regime.

REQUESTER enum

The requester of this table walk.

SIDE enum

Inst / Data.

STAGE unsigned int

Translation stage.

MMU_TTB_WRITE

This event is triggered by writes caused by a translation table walk. Fields:

ABORTED bool

Was the walk successful.

DATA unsigned int

The data written.

DESC_KIND enum

The encoding scheme of the descriptors used for this walk.

IPA unsigned int

For stage 1, the IPA of the write.

IS_HACDBS_ACCESS bool

If the access was due to HACDBS_CLEAN.

LEVEL unsigned int

Translation table level.

LPAE bool

Is this for an LPAE translation (*DEPRECATED*, use DESC_KIND).

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSmem bool

Non secure memory access.

NSreq bool

Non secure request.

OUTADDR unsigned int

The output physical address.

PADDR unsigned int

The physical address of the write.

PASmem enum

Physical Address Space of the accessed memory.

PASreq enum

Physical Address Space of the requested lookup.

REGIME_EL enum

Entry matches in this translation regime.

SIDE enum

Inst / Data.

STAGE unsigned int

Translation stage.

MODE_CHANGE

Mode change. Fields:

CORE_NUM *unsigned int*

Core number in a multi processor.

MODE *enum*

The new mode.

NON_SECURE *enum*

The core's new non-secure bit.

OLD_MODE *enum*

The old mode.

SECURITY_STATE *enum*

The new security state.

NOT_TAG_CHECKED_INST

MTE Tag not checked for current instruction.

PERIODIC

Called for every quantum. Fields:

CORE_NUM *unsigned int*

Core number in a multi processor.

INST_COUNT *unsigned int*

The instruction count of this CPU.

PC *unsigned int*

The address of the next instruction to be executed on this CPU.

PMU_COUNTER_OVERFLOW

PMU counter overflow. Fields:

COUNTER_GROUP *enum*

Counter group.

COUNTER_RANGE *enum*

Counter range.

EVENT_ID *unsigned int*

Event Identifier, field is valid only for Event counter group.

INDEX *unsigned int*

Counter index (as selected by PMSELR), field is valid only for Event/Cycle counter groups.

INTERRUPT *bool*

Is interrupt enabled on overflow for this counter.

POTINDEX_UPDATE

Value of POT index for current pc changed. Fields:

NEW_POTINDEX unsigned int

New POTIndex.

OLD_POTINDEX unsigned int

Old POTIndex.

PREFETCH_MEMORY64

Prefetch from PRFM or PRFUM instructions. Fields:

PRFOP enum

Prefetch hint.

VADDR unsigned int

Virtual address of the location that should be prefetched.

PRELOAD64_REQUEST

Preload requests from PRFM or PRFUM instructions. Fields:

PRFOP enum

Prefetch hint.

VADDR unsigned int

Virtual address of the location that should be prefetched.

PRELOAD_DATA

Data preload from PLD instruction. Fields:

VADDR unsigned int

Virtual address of the data that should be preloaded.

PRELOAD_INST

Instruction preload from PLI instruction. Fields:

VADDR unsigned int

Virtual address of the instruction that should be preloaded.

PRE_CORE_LOAD

Trace just before a core load. Fields:

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

PRE_CORE_STORE

Trace just before a core store. Fields:

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

PRE_TTB_READ

This event is triggered before reads caused by a translation table walk. Fields:

ADDR unsigned int

The physical address of the read.

PSTATE

PSTATE fields update. Fields:

ALLINT unsigned int

All interrupt mask bit.

BTYPE unsigned int

Branch target identification bit.

CORE_NUM unsigned int

Core number in a multi processor.

DAIF unsigned int

Exception mask bits.

DIT unsigned int

Data Independent Timing bit.

EL unsigned int

Current Exception level field.

EXLOCK unsigned int

Exception return state lock bit.

IL unsigned int

Illegal Execution state bit.

NZCV unsigned int

Condition flags.

PACM unsigned int

PACM bit.

PAN unsigned int

Privileged Access Never state bit.

PM unsigned int

PMU exception mask bit.

PPEND unsigned int

PMU exception pending bit.

SM unsigned int

StreamingMode status bit.

SP unsigned int

Stack pointer register selection bit.

SS unsigned int

Software Step bit.

SSBS unsigned int

Speculative Store Bypass Safe bit.

TCO unsigned int

Tag Check Override bit.

TOUCHED unsigned int

Mask of the fields that have been touched since this trace has last been emitted. Note that touched does not necessarily mean the value changed. The index of the field is the bit index in the mask plus the index of this field plus one.

UAO unsigned int

User Access Override bit.

UINJ unsigned int

Undefined Instruction exceptions bit.

ZA unsigned int

ZA storage and ZT0 register enable bit.

nRW unsigned int

Current Execution state bit.

PSTATE_PACTIVE_OPMODE_TRANSITION

Trace pactive and Opmode mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Operating mode after the transition.

OLD_STATE enum

Operating mode before the transition.

PSTATE_PACTIVE_OPMODE_TRANSITION_DENIED

Trace the denied pactive and Opmode mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Operating mode after the transition.

OLD_STATE enum

Operating mode before the transition.

REASON string

Reason for the transition being denied.

PSTATE_PACTIVE_POWER_TRANSITION

Trace pactive and power mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Power mode after the transition.

OLD_STATE enum

Power mode before the transition.

PSTATE_PACTIVE_POWER_TRANSITION_DENIED

Trace the denied pactive and power mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Power mode after the transition.

OLD_STATE enum

Power mode before the transition.

REASON string

Reason for the transition being denied.

PSTATE_PACTIVE_THREAD_OPMODE_TRANSITION

Trace pactive and Opmode mode transitions for threads on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Operating mode after the transition.

OLD_STATE enum

Operating mode before the transition.

QUANTUM_START

Called at the start of every quantum.

RANGE_PRELOAD_REQUEST

Preload requests from RPRFM. Fields:

COUNT unsigned int

Number of blocks of data to access.

LENGTH unsigned int

Number of contiguous bytes to be accessed in each block.

OP unsigned int

Range prefetch operation .

REUSE unsigned int

Maximum number of bytes to be accessed before next RPRFM .

STRIDE signed int

Stride to advance after accessing LENGTH bytes.

VADDR unsigned int

Base virtual address of range prefetch request.

RUN_STATE

Run state transition. Fields:

INST_COUNT unsigned int

Ticks count at point of transition.

NEW enum

New run state.

OLD enum

Old run state.

SEMIHOSTING_CALL

Call of a semihost function occurred. Fields:

PC unsigned int

The program counter after the semihosting call.

SEMIHOSTING_PRECALL

About to call semihost. Fields:

EL2 bool

Use EL2 translation regime.

NS bool

Is Non-Secure.

REG_WIDTH unsigned int

The current register width in bytes.

SIGNAL

External signal state change. Fields:

SIGNAL enum

Signal that changed.

STATE bool

Signal asserted state.

SOFTWARE_STEP

Return the Debug Software Step state. Fields:

STATE enum

Software Step state.

SPSR

SPSR change. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

MODE enum

Which of the banked SPSR registers is written.

OLD_VALUE unsigned int

The old SPSR value.

VALUE unsigned int

The new SPSR value.

START_COMPILE

Compilation started. Fields:

VADDR unsigned int

Instruction where compilation begins.

STG_MULTIPLE_MICRO_OPS_ADDRESS_RANGE

Address range for MTE stores with multiple uops. Fields:

ADDR_END unsigned int

Final address.

ADDR_START unsigned int

Starting address.

STORE_MULTIPLE_REGS_ADDRESS_RANGE

Address range for stores with multiple registers. Fields:

ADDR_END unsigned int

Final address.

ADDR_START unsigned int

Starting address.

SUBOPTIMAL_LDST_RETIRED

suboptimal load/store. Fields:

ACCESS_SIZE unsigned int

Log2 of the access width in bytes used for alignment checking (1: half-word, 2: word ...).

OPTIMAL_ALIGN_SELECTOR unsigned int

Selector to decide the optimal alignment size (1: access width, 2: 4-bytes, 3: 8-bytes, ..., 12: 4-kbytes).

VADDR unsigned int

The virtual address of the access.

SYNC

Called for every synchronization, that is every quantum break. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The instruction count of this CPU.

LOCAL_QUANTUM unsigned int

The local quantum of this CPU.

LOCAL_TIME unsigned int

The local time of this CPU.

SYNC_REASON enum

Reason for sync.

SYSCALL

System call instruction executed. Fields:

IMM unsigned int

Immediate value of the system call instruction.

TYPE enum

System call type.

VADDR unsigned int

Instruction that caused the system call.

SYSREG_READ128

System Coprocessor 128 bit register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

REGNUM enum

Internal register number.

REGNUM_PHYS enum

Internal register number physically accessed (e.g. in presense of VHE redirection).

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_READ64

System Coprocessor register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REGNUM enum

Internal register number.

REGNUM_OPERAND enum

Register number corresponding to the instruction operand.

REGNUM_PHYS enum

Internal register number physically accessed (e.g. in presense of VHE redirection).

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_UPDATE128

Triggers when the system updates a register. Fields:

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG enum

Register number.

UNKNOWN unsigned int

Bits of the register which became unknown.

VALUE unsigned int

Value written to the register.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_UPDATE32

Triggers when the system updates a register. Fields:

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG enum

Register number.

UNKNOWN unsigned int

Bits of the register which became unknown.

VALUE unsigned int

Value written to the register.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int
opcode 2.

SYSREG_UPDATE64

Triggers when the system updates a register. Fields:

CRm unsigned int
CRm.

CRn unsigned int
CRn.

NS bool
Secure or nonsecure banked register is accessed.

REG enum
Register number.

UNKNOWN unsigned int
Bits of the register which became unknown.

VALUE unsigned int
Value written to the register.

opc unsigned int
opcode 1.

opc0 unsigned int
opcode 0.

opc2 unsigned int
opcode 2.

SYSREG_WRITE128

System Coprocessor 128 bit register write. Fields:

CORE_NUM unsigned int
Core number in a multi processor.

CRm unsigned int
CRm.

CRn unsigned int
CRn.

REGNUM enum
Internal register number.

REGNUM_PHYS enum
Internal register number physically accessed (e.g. in presense of VHE redirection).

UNDEF bool
The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_WRITE64

System Coprocessor register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REGNUM enum

Internal register number.

REGNUM_OPERAND enum

Register number corresponding to the instruction operand.

REGNUM_PHYS enum

Internal register number physically accessed (e.g. in presense of VHE redirection).

UNDEF bool

The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

UNALIGNED_LDST_RETIRED

Processor unaligned load/store. Fields:

ACCESS_SIZE unsigned int

Log2 of the access width in bytes used for alignment checking (1=>half-word, 2->word ...).

NUMBER_OF_BEATS unsigned int

Number of accesses (beats) in this burst (Total size in bytes is ACCESS_SIZE * NUMBER_OF_BEATS).

VADDR unsigned int

The virtual address of the access.

VFP_D_REGS

VFP/NEON D 64 bit register write. Fields:

ALIASING enum

Type of register aliasing used.

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a write to the corresponding bit in the register. For example, 0x1 would indicate a write to VALUE[0].

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

VFP_D_REGS_READ

VFP/NEON D 64 bit register read. Fields:

ID unsigned int

The register number.

VALUE unsigned int

The current value of the register.

VFP_Q_REGS

VFP/NEON Q 128 bit register write. Fields:

ALIASING enum

Type of register aliasing used.

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a write to the corresponding byte in the register. For example, 0x1 would indicate a write to VALUE[7:0].

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

VFP_Q_REGS_READ

VFP/NEON Q 128 bit register read. Fields:

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a read to the corresponding byte in the register. For example, 0x1 would indicate a read to VALUE[7:0].

VALUE unsigned int

The current value of the register.

VFP_SYS_REGS

Writes to the VFP/NEON units system registers. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ID enum

Which VFP system register is written.

MASK unsigned int

The mask indicating updated cumulative bits.

OLD_VALUE unsigned int

The register's old value overwritten.

VALUE unsigned int

The new value written to the VFP system register.

VFP_S_REGS

VFP/NEON S 32 bit register write. Fields:

ALIASING enum

Type of register aliasing used.

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a write to the corresponding bit in the register. For example, 0x1 would indicate a write to VALUE[0].

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

VFP_S_REGS_READ

VFP/NEON S 32 bit register read. Fields:

ID unsigned int

The register number.

VALUE unsigned int

The current value of the register.

VIRTUAL_TAG_ALLOCATED

Virtual tag information when allocated. Fields:

TAG_VADDR unsigned int

Virtual tag virtual address.

VADDR unsigned int

Data Virtual address.

VIRTUAL_TAG unsigned int

Virtual tag value.

WAYPOINT

Signals end of basic block execution. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ISSET enum

Origin instruction set.

IS_COND bool

Indicates if this is a conditional waypoint.

PC unsigned int

Origin address (or 0 if unavailable).

TAKEN bool

Indicates if this waypoint was taken.

TARGET unsigned int

Destination address.

TARGET_ISET enum

Destination instruction set.

WFE_END

WFE ended. Fields:

INST_COUNT unsigned int

Ticks count when leaving WFE.

WFE_EVENT_REGISTER

WFE event register status: set/clear, reason. Fields:

INST_COUNT unsigned int

Ticks count.

REASON enum

Reason for set/clear. Only REASON==0 (Cleared by WFE) clears the bit, all other reasons set it.

WFE_IGNORED

WFE ignored. Fields:

EVENT bool

This WFE was ignored because the event register was set.

INST_COUNT unsigned int

Ticks count when ignoring WFE.

TRAPPED bool

This WFE was trapped.

WFE_START

WFE entered. Fields:

INST_COUNT unsigned int

Ticks count when entering WFE.

WFI_END

WFI ended. Fields:

INST_COUNT unsigned int

Ticks count when leaving WFI.

WFI_IGNORED

WFI ignored. Fields:

DISABLED bool

This WFI was ignored because WFI is disabled.

INST_COUNT unsigned int

Ticks count when ignoring WFI.

REASON enum

specifies reason why WFI trace was ignored.

TRAPPED bool

This WFI was trapped.

WFI_START

WFI entered. Fields:

INST_COUNT unsigned int

Ticks count when entering WFI.

WFI_WAKEUP

WFI wakeup. Fields:

INST_COUNT unsigned int

Ticks count when WFI wakeup occurred.

REASON enum

Reason for wakeup.

2.16 ARM_Cortex-A510

This section describes the trace sources.

AA64_ASE_SVE_REGS

Modification of AArch64 Advanced SIMD or SVE Vector Registers (via SIMD or Scalar operations), SVE Predicate Registers and ZT0. Fields:

ID enum

The register identifier.

MASK unsigned int

Bitmask of the register to signal the modified bits in the VALUE field.

SM bool

Whether the PE is in Streaming Mode.

VALUE unsigned int

Value of AArch64 vector register.

ASYNC_MEMORY_FAULT

Context ID Register write. Fields:

FAULT unsigned int

Fault status present in ESR.ISS encoding for Serror in ARCH64 and present in DFSR encoding for external abort format in ARCH32.

PADDR unsigned int

Physical Address (or 0 if unavailable).

VADDR signed int

Virtual Address (or 0 if unavailable).

ATOMIC_END_ACCESS

Bus trace access for atomics. Fields:

ACCESS_FAIL bool

Memory access failed.

ADDR unsigned int

The virtual address of the access.

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

COMPARE_VALUE unsigned int

Compare value for CAS.

LOAD_VALUE unsigned int

Loaded values.

NSDESC unsigned int

The security state of the access.

OPERAND_VALUE unsigned int

Operation's operand.

OPERATION enum

Operation type.

PADDR unsigned int

The physical address of the access.

PRIV bool

Is this a privileged access?.

ATOMIC_START_ACCESS

Bus trace access for atomics. Fields:

ADDR unsigned int

The virtual address of the access.

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

COMPARE_VALUE unsigned int

Compare value for CAS.

NSDESC unsigned int

The security state of the access.

OPERAND_VALUE unsigned int

Operation's operand.

OPERATION enum

Operation type.

PADDR unsigned int

The physical address of the access.

PRIV bool

Is this a privileged access?.

ArchMsg.Error.exit_code

DISPLAY exit code: %{EXITCODE} component kind: %{KIND} reason: %{REASON}. Fields:

EXITCODE signed int

Exit code to be returned.

KIND string

Component kind that invoked the exit code trace.

REASON string

Reason why the exit code trace was invoked (optional).

ArchMsg.Warning.IgnoredWarmResetRequest

Event fires when software attempts to request warm reset through write to deprecated DBGPRCR.CWRR.

ArchMsg.Warning.RAS_ErrselrUnpredictableState

DISPLAY ERRSELR is in unpredictable state due to value greater than number_of_error_records written to it.

ArchMsg.Warning.RAS_UnpredictableStateWrite

DISPLAY %{REG_NAME} write ignored due to unpredictable ERRSELR state. Fields:

REG_NAME string

Name of register being written.

ArchMsg.Warning.ReportedRevisionRegisterRead

DISPLAY Reported revision and patch do not match that of the cpu. Fields:

ORIGINAL_VALUE unsigned int

The value the register would have based on the modelled revision.

REGISTER string

The name of the register that has been read.

REPORTED_VALUE unsigned int

The value the register has been overridden to.

ArchMsg.Warning.Unpredictably Indexed PM Event Register

TAGS **UNPREDICTABLE** DISPLAY Event %{IsDirect:(Selector value|Register)} %{SEL} is >= %{N}.
Fields:

IsDirect unsigned int

Direct Access rather than indirect.

N unsigned int

Number of Accessible Registers.

SEL unsigned int

Selector.

ArchMsg.Warning.cache_contents_unknown

A cache has been activated without being invalidated and may contain **UNKNOWN** content.
DISPLAY %{SIDE:(D|I)}-Cache was enabled in %{REGIME:(SECURE|NON_SECURE|HYP)} regime but was not invalidated since power-on: cache lines could contain **UNKNOWN** content
%{SIDE:(Note: D-side caches must invalidate SECURE lines even if the cache is only used by NON_SECURE code in order to ensure that unexpected dirty lines tagged SECURE are not naturally evicted)}. Fields:

REGIME unsigned int

Current translation regime.

SIDE unsigned int

0 d-side cache or unified cache, 1 i-side.

ArchMsg.Warning.dcimvac_matches_watchpoint

TAGS **UNPREDICTABLE** DISPLAY Watchpoints matching an AArch32 DCIMVA are implementation defined.

ArchMsg.Warning.decode_fieldmismatch

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with different fields %{FIELD1} and %{FIELD2}. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_frachitsnegative

TAGS **UNPREDICTABLE** DISPLAY VCVT Instruction is unpredictable with a negative number of fraction bits.

ArchMsg.Warning.decode_initblock

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable in an IT block.

ArchMsg.Warning.decode_initblocknotlast

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable in an IT block when not the last.

ArchMsg.Warning.decode_invalid_condition

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable when the condition is not AL.

ArchMsg.Warning.decode_invalidfieldcombination

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with the values in %{FIELD1} and %{FIELD2}. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_invalidvalue

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with the value in %{FIELD}. Fields:

FIELD enum

Field name.

ArchMsg.Warning.decode_mem_hint_unallocated

TAGS **UNPREDICTABLE** DISPLAY There is no memory hint allocated to this bit pattern.

ArchMsg.Warning.decode_registermatch

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with %{FIELD1} and %{FIELD2} as the same register. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_registermismatch

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with %{FIELD1} and %{FIELD2} as different registers. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_registeroutofrange

TAGS **UNPREDICTABLE** DISPLAY Floating-point instruction is unpredictable indexing beyond the end of the register bank.

ArchMsg.Warning.decode_sbzsbo

TAGS UNPREDICTABLE DISPLAY Reserved bits in the instruction are not canonical.

ArchMsg.Warning.decode_transactiontoobig

TAGS UNPREDICTABLE DISPLAY Attempt to perform a memory transaction of more than 128 bytes.

ArchMsg.Warning.decode_unpred_other

TAGS UNPREDICTABLE DISPLAY Instruction is unpredictable.

ArchMsg.Warning.decode_unpreduseofpc

TAGS UNPREDICTABLE DISPLAY Instruction is unpredictable when %{FIELD} is PC. Fields:

FIELD enum

Field name.

ArchMsg.Warning.decode_unpreduseofr13

TAGS UNPREDICTABLE DISPLAY Instruction is unpredictable when %{FIELD} is R13. Fields:

FIELD enum

Field name.

ArchMsg.Warning.decode_unpreduseofr13inreglist

TAGS UNPREDICTABLE DISPLAY Instruction is unpredictable with R13 in register-list.

ArchMsg.Warning.decode_useofd16orhigher

TAGS UNPREDICTABLE DISPLAY Attempt to use register D16 or higher where it is not allowed.

ArchMsg.Warning.decode_writebackandbaseinlist

TAGS UNPREDICTABLE DISPLAY Instruction is unpredictable with write back when the base register is in the transfer list.

ArchMsg.Warning.decode_zeroregistersinlist

TAGS UNPREDICTABLE DISPLAY Instruction is unpredictable when no registers are to be transferred.

ArchMsg.Warning.misaligned_ptrs

DISPLAY TRBPTR_EL1(%{TRBPTR_EL1}) or TRBTRG_EL1(%{TRBTRG_EL1}) are not aligned as per TRBIDR_EL1.Align(%{TRBIDR_EL1}). Fields:

TRBIDR_EL1 unsigned int

TRBIDR_EL1.Align value.

TRBPTR_EL1 unsigned int

Current write pointer.

TRBTRG_EL1 unsigned int

Trigger pointer.

ArchMsg.Warning.out_of_range_ptrs

DISPLAY TRBPTR_EL1(%{TRBPTR_EL1}) is out of range as per TRBBASER_EL1(%{TRBBASER_EL1}) or TRBLIMITR_EL1.LIMIT(%{TRBLIMITR_EL1}). Fields:

TRBBASER_EL1 unsigned int

Base pointer.

TRBLIMITR_EL1 unsigned int

Limit pointer.

TRBPTR_EL1 unsigned int

Current write pointer.

ArchMsg.Warning.recursive_branch

An instruction is performing a branch that targets the same instruction. If this was intended then a WFI might be more effective. DISPLAY recursive branch detected at PC=%{PC}.

PRIMARY KEY PC. Fields:

PC unsigned int

Address of instruction causing the branch.

ArchMsg.Warning.recursive_exception

An instruction has generated an exception that targets the same instruction. DISPLAY recursive %{TYPE} exception detected at PC=%{PC} (CPSR=%{CPSR}). PRIMARY KEY PC. Fields:

CPSR unsigned int

Processor state.

PC unsigned int

Address of instruction causing the exception.

TYPE enum

Type of exception.

ArchMsg.Warning.reentrant_vector_catch

TAGS **UNPREDICTABLE** DISPLAY Vector catch debug event on Prefetch or Data abort vector %{ADDR}. Fields:

ADDR unsigned int

vector address.

ArchMsg.Warning.reserved_it_state

TAGS **UNPREDICTABLE** DISPLAY Execution with reserved IT state %{ITSTATE}. Fields:

ITSTATE unsigned int

reserved IT state.

ArchMsg.Warning.tlb_contents_unknown

Translation has been enabled without invalidating the TLB. DISPLAY Translation was enabled in the %{PAS:(Secure|Non-secure|Root|Realm)} EL%{EL:d} translation regime (vmid=%{VMID:x}) without invalidating %{INVALIDITY:(|d-side |i-side |)}entries since power-on: %{INVALIDITY:(|D||I)}TLB could contain **UNKNOWN** content. Fields:

EL unsigned int

Exception level of translation regime.

INVALIDITY unsigned int

Which TLB may be invalid (if TLBs are separate).

NON_SECURE bool

Whether the translation regime is non-secure.

PAS unsigned int

Physical address space of translation regime.

VMID unsigned int

Current VMID, if applicable.

ArchMsg.Warning.unknown_DLR_DSPSR

Exiting debug state with unknown DLR or DSPSR.

ArchMsg.Warning.unknown_ELR_SPSR

Returning from debug exception with unknown ELR or SPSR.

ArchMsg.Warning.unpred_FAR_on_LRCPC3_non_atomic_mem_access

TAGS **UNPREDICTABLE** Unpredictable FAR/FSTATUS on LRCPC3 pair instructions resulting in two separate memory accesses
 DISPLAY Unpredictable FAR on LRCPC3 non atomic memory access at %{ADDR1} with FSTATUS %{FSTATUS1} or %{ADDR2} with FSTATUS %{FSTATUS2}. Fields:

ADDR1 unsigned int

Possible FAR address 1.

ADDR2 unsigned int

Possible FAR address 2.

FSTATUS1 unsigned int

Possible fault status value 1.

FSTATUS2 unsigned int

Possible fault status value 2.

ArchMsg.Warning.unpred_async_tag_check_fail_on_sve_speculative_load_do_update_ffr

Unpredictable set to FALSE of the FFR element associated to SVE speculative load with an async tag check fail TAGS **UNPREDICTABLE**.

ArchMsg.Warning.unpredictable_a32_breakpoint

TAGS **UNPREDICTABLE** DISPLAY Unpredictable breakpoint on a A32 instruction at %{ADDR1} Offset %{OFFSET} BAS %{BAS} for a %{IS_ADDRESS_MISMATCH_BREAKPOINT:(Mismatch|Match)} Breakpoint number %{N:d} Breakpoint address %{ADDR2}. Fields:

ADDR1 unsigned int

Address.

ADDR2 unsigned int

Breakpoint address.

BAS unsigned int

BAS field.

IS_ADDRESS_MISMATCH_BREAKPOINT bool

Is a Mismatch breakpoint, not a Match breakpoint.

N unsigned int

Breakpoint number.

OFFSET unsigned int

Always has a value of zero.

ArchMsg.Warning.unpredictable_banked_register_access

TAGS **UNPREDICTABLE** DISPLAY Access to banked register (R=%{R:d} SYSm=%{SYSM:2x}) is unpredictable. Fields:

R unsigned int

R bit.

SYSM unsigned int

SYSm.

ArchMsg.Warning.unpredictable_ccfail_undef

TAGS **UNPREDICTABLE** DISPLAY An instruction %{OPCODE} at VA=%{VADDR} failed its condition codes check but would otherwise have caused a non-data-dependent trap or undefined exception. Behaviour is implementation-defined choice of exception or nop. Fields:

OPCODE unsigned int

opcode of the instruction.

VADDR unsigned int

Virtual address of the instruction being executed.

ArchMsg.Warning.unpredictable_ici_bits

TAGS **UNPREDICTABLE** DISPLAY ICI bits do not apply or are architecturally unpredictable for instruction.

ArchMsg.Warning.unpredictable_t32_breakpoint

TAGS **UNPREDICTABLE** DISPLAY Unpredictable breakpoint on a T32 instruction at %{ADDR1} Offset %{OFFSET} BAS %{BAS} for a %{IS_ADDRESS_MISMATCH_BREAKPOINT:(Mismatch|Match)} Breakpoint number %{N:d} Breakpoint address %{ADDR2}. Fields:

ADDR1 unsigned int

Address.

ADDR2 unsigned int

Breakpoint address.

BAS unsigned int

BAS field.

IS_ADDRESS_MISMATCH_BREAKPOINT bool

Is a Mismatch breakpoint, not a Match breakpoint.

N unsigned int

Breakpoint number.

OFFSET unsigned int

Indicates which halfword of a 32-bit Thumb instruction the breakpoint is set on. 0 for the first halfword or if the instruction is 16 bits wide, or 1 for the second halfword.

ArchMsg.Warning.unpredictable_t32_vector_catch

TAGS **UNPREDICTABLE** DISPLAY Unpredictable Vector catch exception on a T32 instruction at %{ADDR} Offset %{OFFSET}. Fields:

ADDR unsigned int

Address.

OFFSET unsigned int

Indicates which halfword of a 32-bit Thumb instruction the vector catch exception is set on. 0 for the first halfword or if the instruction is 16 bits wide, or 1 for the second halfword.

ArchMsg.Warning.unpredictable_watchpoint_far

TAGS **UNPREDICTABLE** DISPLAY The value for FAR/EDWAR for hitting this watchpoint may be anywhere in the range %{LOWER_BOUND} to %{UPPER_BOUND}. Fields:

LOWER_BOUND unsigned int

the lowest address accessed by the instruction that triggered the watchpoint.

UPPER_BOUND unsigned int

the highest watchpointed address accessed by that instruction.

ArchMsg.Warning.unpredictable_watchpoint_far_isscmp

DISPLAY The value for FAR/EDWAR for hitting this watchpoint may be anywhere in the range %{LOWER_BOUND} to %{UPPER_BOUND}. Fields:

LOWER_BOUND unsigned int

the lowest address accessed by the instruction that triggered the watchpoint.

UPPER_BOUND unsigned int

the highest watchpointed address accessed by that instruction.

ArchMsg.Warning.warning_AdvSIMDExpandImmUnexpectedZero

TAGS **UNPREDICTABLE** DISPLAY AdvSIMDExpandImm may treat this immediate value as **UNPREDICTABLE**.

ArchMsg.Warning.warning_ConditionalSMC

TAGS **UNPREDICTABLE** DISPLAY SMC instruction has **UNPREDICTABLE** effects when conditional and when combined with traps.

ArchMsg.Warning.warning_TLBID_out_of_range

TLBID out of range in TLBI according to the configuration of NIS, NVIS, NOS and NVOS. Fields:

NIS unsigned int

NIS.

NOS unsigned int

NOS.

NVIS unsigned int

NVIS.

NVOS unsigned int

NVOS.

TLBID unsigned int

The TLBID value used in the TLBI instruction.

ArchMsg.Warning.warning_TTL_or_TG_mismatches_current_translation_granule

TTL or TG mismatches current translation granule in TLBI VA/RVA/IPA/RIPA. Fields:

CURRENT_TG unsigned int

Current TG as indicated by TCR_ELx or VTCR_EL2.

TLBI_TG unsigned int

TG (or TTL[3:2]) used in TLBI instruction.

ArchMsg.Warning.warning_access_crosses_page_boundary

TAGS **UNPREDICTABLE** DISPLAY Access crosses page boundary.

ArchMsg.Warning.warning_access_crosses_page_boundary_has_fault

TAGS **UNPREDICTABLE** DISPLAY Access that crosses page boundary has fault(s).

ArchMsg.Warning.warning_access_crosses_page_boundary_has_fault_on_both_pages

TAGS **UNPREDICTABLE** DISPLAY Access that crosses page boundary has a fault on each page.
Fields:

ADDR1 unsigned int

FAR address on first page.

ADDR2 unsigned int

FAR address on second page.

FSTATUS1 unsigned int

Fault status on first page.

FSTATUS2 unsigned int

Fault status on second page.

ArchMsg.Warning.warning_access_crosses_page_boundary_spanning_different_memory

TAGS **UNPREDICTABLE** DISPLAY Access at address %{ADDR} crosses page from MEM_TYPE %{MEMTYPE_PAGE1} to %{MEMTYPE_PAGE2}. Fields:

ADDR unsigned int

address of access crossing page.

MEMTYPE_PAGE1 signed int

First Page's memory type (0-Normal, 1-Non-Normal).

MEMTYPE_PAGE2 signed int

Second Page's memory type (0-Normal, 1-Non-Normal).

ArchMsg.Warning.warning_access_wraps_around_memory

TAGS UNPREDICTABLE DISPLAY Access wraps around memory in %{WIDTH:d}bit mode.
Fields:

WIDTH unsigned int

Address width.

ArchMsg.Warning.warning_bcr_linking_status

TAGS UNPREDICTABLE DISPLAY DBGBCR.{BT2,BT} combination gives unpredictable behaviour.

ArchMsg.Warning.warning_bcr_mask_reserved

TAGS UNPREDICTABLE DISPLAY DBGBCR_ELn.MASK is set to a reserved value.

ArchMsg.Warning.warning_bvr_masked_bit_not_zero

TAGS UNPREDICTABLE DISPLAY DBGBVRn_EL1.VA contains a bit masked by DBGBCRn_EL1.MASK that is not zero.

ArchMsg.Warning.warning_ccsidr_unimplemented_level

TAGS UNPREDICTABLE DISPLAY CCSIDR read while CSSELR %{CSSELR:x} points to an unimplemented cache level. Fields:

CSSELR unsigned int

current effective value of CSSELR.

ArchMsg.Warning.warning_change_to_ns_when_tge_set

TAGS UNPREDICTABLE DISPLAY Attempting to change NS to 1 when HCR.TGE = 1.

ArchMsg.Warning.warning_contiguous_bit_check_abort

TAGS UNPREDICTABLE DISPLAY An abort occurred whilst attempting to check TLB entry is contiguous at %{ENTRY_ADDR}. Fields:

ENTRY_ADDR unsigned int

address of conflicting TLB entry.

ArchMsg.Warning.warning_contiguous_bit_error

TAGS UNPREDICTABLE DISPLAY TLB entry at %{ENTRY_ADDR} was not contiguous with entry at %{CONTIG_ADDR} Contents are %{ENTRY_DATA} but expected %{CONTIG_DATA}.
Fields:

CONTIG_ADDR unsigned int

address of the TLB first read, with which this entry is expected to be contiguous.

CONTIG_DATA unsigned int

expected contents based on the entry at CONTIG_ADDR.

ENTRY_ADDR unsigned int

address of conflicting TLB entry.

ENTRY_DATA unsigned int

contents of conflicting TLB entry.

ArchMsg.Warning.warning_contiguous_bit_error_D128

TAGS **UNPREDICTABLE** DISPLAY TLB entry at %{ENTRY_ADDR} was not contiguous with entry at %{CONTIG_ADDR} Contents are %{ENTRY_DATA_HI} %{ENTRY_DATA_LO} but expected %{CONTIG_DATA}. Fields:

CONTIG_ADDR unsigned int

address of the TLB first read, with which this entry is expected to be contiguous.

CONTIG_DATA unsigned int

expected contents based on the entry at CONTIG_ADDR.

ENTRY_ADDR unsigned int

address of conflicting TLB entry.

ENTRY_DATA_HI unsigned int

Higher bit contents of conflicting TLB entry.

ENTRY_DATA_LO unsigned int

Lower bit contents of conflicting TLB entry.

ArchMsg.Warning.warning_cp10_cp11_mismatch

TAGS **UNPREDICTABLE** DISPLAY Unpredictable configuration of CP10 and CP11 controls at EL %{EL:d}. Fields:

EL unsigned int

EL owning the control.

ArchMsg.Warning.warning_cp10_cp11_reserved_value

TAGS **UNPREDICTABLE** DISPLAY Unpredictable configuration of CPACR CP10 (%{CP10}) or CP11 (%{CP11}). Fields:

CP10 unsigned int

CP10 access.

CP11 unsigned int

CP11 access.

ArchMsg.Warning.warning_csselr_level_out_of_range

TAGS **UNPREDICTABLE** DISPLAY CSSELR.Level out of range, written value %{LEVEL} with only %{IMPLEMENTED:d} levels implemented. Fields:

IMPLEMENTED unsigned int

number of cache levels implemented.

LEVEL unsigned int

written value of CSSELR.Level.

TYPE enum

The type of cache selected by CSSELR.InD and TnD.

ArchMsg.Warning.warning_debug_flow_control_bits_not_obeyed

TAGS **UNPREDICTABLE** DISPLAY Flow control bits for %{REGISTER} were not obeyed, causing %{ERROR}. Fields:

ERROR enum

type of error resulting.

REGISTER enum

register under consideration.

ArchMsg.Warning.warning_debug_register_access_during_reset

TAGS **UNPREDICTABLE** DISPLAY Unpredictable %{IS_WRITE:(read|write)} access to debug register offset %{OFFSET} during reset. Fields:

IS_WRITE unsigned int

Write Not Read.

OFFSET unsigned int

Register Offset.

ArchMsg.Warning.warning_decode_cps_inconsistent_fields

TAGS **UNPREDICTABLE** DISPLAY The mode setting fields in the CPS instruction are inconsistent.

ArchMsg.Warning.warning_decode_invalid_state

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable in current exception-level/security state.

ArchMsg.Warning.warning_default_cacheable_mmu_on

TAGS **UNPREDICTABLE** DISPLAY Default cacheable is enabled (HCR.DC = 1) while MMU is enabled.

ArchMsg.Warning.warning_default_cacheable_vmmu_off

TAGS **UNPREDICTABLE** DISPLAY Default cacheable is enabled (HCR.DC = 1) while HCR.VM = 0.

ArchMsg.Warning.warning_deprecated_wvr_bit_2_set

TAGS **UNPREDICTABLE** DISPLAY DBGWVR%{N:d}_EL1=%{WVR:h} has bit 2 set, deprecated in ARMv8. Fields:

N unsigned int

index.

WVR unsigned int

DBGWVR.

ArchMsg.Warning.warning_eret_while_software_step_active_pending

Missing ISB between setting MDSCR_EL1.SS and ERET TAGS **UNPREDICTABLE**.

ArchMsg.Warning.warning_exclusive_to_non_normal

TAGS **UNPREDICTABLE** DISPLAY Exclusive Access to Non-Normal Memory %{ADDR}. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_exclusive_to_non_writeback

TAGS **UNPREDICTABLE** DISPLAY Exclusive Access to Non-Writeback-Cacheable Memory %{ADDR}. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_execute_from_device_memory

TAGS **UNPREDICTABLE** DISPLAY An attempt was made to execute from Device memory.

ArchMsg.Warning.warning_illegal_cpsr_mode

TAGS **UNPREDICTABLE** DISPLAY Writing an illegal or unimplement mode (%{NEW_MODE:h}) to CPSR was unpredictable. Fields:

NEW_MODE unsigned int

New value of CPSR.M.

ArchMsg.Warning.warning_illegal_srs_mode

TAGS **UNPREDICTABLE** DISPLAY Illegal or **UNPREDICTABLE** mode (%{MODE:h}) used for SRS instruction. Fields:

MODE unsigned int

mode for Banked SP.

ArchMsg.Warning.warning_implementation_defined_read_debug_register_in_SCS

Read Access to Debug registers from the processor is **IMPLEMENTATION DEFINED** TAGS **IMP_DEF**. Fields:

PPB_OFFSET unsigned int

debug register offset.

ArchMsg.Warning.warning_implementation_defined_sequential_security_transitions_supported

The behavior of sequential instruction fetches that cross from non-secure to secure memory and contain an SG instruction is **CONSTRAINED UNPREDICTABLE** TAGS **UNPREDICTABLE**.

ArchMsg.Warning.warning_implementation_defined_stack_limit_check_supported

It is **IMPLEMENTATION DEFINED** whether stack pointer limit checking is performed for this instruction TAGS **IMP_DEF**. Fields:

ADDRESS unsigned int

Address of instruction.

OPCODE unsigned int

instruction opcode.

ArchMsg.Warning.warning_implementation_defined_taggednonsharedmemorytype

TAGS IMP_DEF DISPLAY Memory attribute has Tagged, non shared memory type; it is **IMPLEMENTATION DEFINED** whether the memory location is treated as Tagged or non-Tagged.

ArchMsg.Warning.warning_implementation_defined_write_debug_register_in_SCS

Write Access to Debug registers from the processor is **IMPLEMENTATION DEFINED** TAGS IMP_DEF. Fields:

DATA unsigned int

data attempted to be written.

PPB_OFFSET unsigned int

debug register offset.

ArchMsg.Warning.warning_invalid_tcr_granule

TAGS **UNPREDICTABLE** DISPLAY TCR.TG%{TG_ID:(0|1)} bits have been set to a granule size %{{REQUEST:d}K not implemented - using %{{SUBSTITUTE:d}K. Fields:

REQUEST signed int

page size requested (or 0 for reserved).

SUBSTITUTE signed int

best guess available page size.

TG_ID bool

bits TG0 or TG1.

ArchMsg.Warning.warning_load_multiple_user_registers_from_user_mode

TAGS **UNPREDICTABLE** DISPLAY An LDM(user registers) instruction executed from user mode.

ArchMsg.Warning.warning_mte_in_mem_attr_not_supported

TAGS **UNPREDICTABLE** DISPLAY Memory attribute %{{ATTR}} has Tagged memory type but Armv8.5-MTE is not implemented. Fields:

ATTR unsigned int

Memory attribute.

ArchMsg.Warning.warning_reserved_breakpoint_state_match

TAGS **UNPREDICTABLE** DISPLAY %{{IS_BREAKPOINT:(Watchpoint|Breakpoint)}} programmed with a reserved combination of HMC, SSC and %{{IS_BREAKPOINT:(PAC|PMC)}}. Fields:

IS_BREAKPOINT unsigned int

Is a breakpoint, not a watchpoint.

ArchMsg.Warning.warning_shareability

TAGS **UNPREDICTABLE** DISPLAY Unpredictable: combination of the 1st and 2nd stages of translation is Normal Inner Non-Cacheable, Outer Non-Cacheable.

ArchMsg.Warning.warning_software_step_set_while_enabled

TAGS **UNPREDICTABLE** DISPLAY MDSCR_EL1.SS set to 1 while software step debug exceptions are enabled.

ArchMsg.Warning.warning_thumb_instruction_wraps_around_memory

TAGS UNPREDICTABLE DISPLAY Thumb instruction wraps around memory.

ArchMsg.Warning.warning_ttbr_sbz_bits_are_not_zero

TAGS UNPREDICTABLE DISPLAY %{TTBR}[%{MSB:d}:0] should be zero, but are not. Fields:

MSB unsigned int

x-1.

TTBR enum

which TTBR register.

ArchMsg.Warning.warning_unaligned_address_dbgdtrrx_write

TAGS UNPREDICTABLE DISPLAY Unaligned address %{ADDR} in X0/R0 when DBGDTRRX written in memory access mode. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unaligned_address_dbgdtrtx_read

TAGS UNPREDICTABLE DISPLAY Unaligned address %{ADDR} in X0/R0 when DBGDTRTX read in memory access mode. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unaligned_to_device

TAGS UNPREDICTABLE DISPLAY Unaligned Access to Device Memory %{ADDR}. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unaligned_to_strongly_ordered

TAGS UNPREDICTABLE DISPLAY Unaligned Access to Strongly Ordered Memory %{ADDR}. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unknown_sau_rnr

TAGS UNPREDICTABLE DISPLAY SAU_RNR was set to an unsupported value.

ArchMsg.Warning.warning_unpredictable_AIRCR_PRIS_and_BFHFNMINs

TAGS UNPREDICTABLE DISPLAY The effect of setting both AIRCR.BFHFNMINs and AIRCR.PRIS to 1 is UNPREDICTABLE.

ArchMsg.Warning.warning_unpredictable_AIRCR_VECTCLRACTIVE_when_not_in_debug

TAGS UNPREDICTABLE DISPLAY The effect of writing a 1 to AIRCR.VECTCLRACTIVE if the processor is not halted in Debug state is UNPREDICTABLE.

ArchMsg.Warning.warning_unpredictable_AIRCR_VECTRESET_and_SYSRESETREQ

TAGS **UNPREDICTABLE** DISPLAY When the processor is halted in Debug state, if a write to the register writes a 1 to both VECTRESET and SYSRESETREQ, the behavior is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_AIRCR_VECTRESET_when_not_in_debug

TAGS **UNPREDICTABLE** DISPLAY The effect of writing a 1 to AIRCR.VECTRESET if the processor is not halted in Debug state is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_AIRCR_incorrect_VKEY

TAGS **UNPREDICTABLE** DISPLAY The value 0x05FA must be written to AIRCR.VKEY, otherwise the register write is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_EXC_RETURN_Reserved_Bit

EXC_RETURN[23:7] are reserved with the special condition that all bits should be written as one. Values other than all 1s are **UNPREDICTABLE**. TAGS **UNPREDICTABLE**. Fields:

EXC_RETURN unsigned int

Special PC value.

SBO_MASK unsigned int

Which bits Should Be One to not UNPRED.

ArchMsg.Warning.warning_unpredictable_PMEVTYPEPn_ELO_VS_reserved_value

TAGS **UNPREDICTABLE** DISPLAY Unpredictable reserved value of PMEVTYPEn<n>_ELO.VS.

ArchMsg.Warning.warning_unpredictable_change_to_DEMCR_MON_STEP_at_insufficient_priority

TAGS **UNPREDICTABLE** DISPLAY The effect of changing DEMCR.MON_STEP at an execution priority that is lower than the priority of the DebugMonitor exception is **CONSTRAINED UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_change_to_priority_of_active_exception

TAGS **UNPREDICTABLE** DISPLAY Changing the priority of an active exception is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_clear_lspact

TAGS **UNPREDICTABLE** DISPLAY **UNPREDICTABLE** state on context switch. Fields:

DRVEXC_TAKEN unsigned int

Derived exception taken, not pended.

REACHED_LAST_STORE unsigned int

Only the last faulted.

ArchMsg.Warning.warning_unpredictable_dbg_exit_unaligned_dlr

TAGS **UNPREDICTABLE** DISPLAY Debug exit to AArch32 with DLR[0] set to 1. PC[0] can be set to 0 or to DLR[0].

ArchMsg.Warning.warning_unpredictable_exception_catch

TAGS **UNPREDICTABLE** DISPLAY Generation of an exception catch event is unpredictable.

ArchMsg.Warning.warning_unpredictable_exception_return_inconsistent_state

TAGS **UNPREDICTABLE** DISPLAY state on exception return is unpredictable.

ArchMsg.Warning.warning_unpredictable_exception_return_instruction

TAGS **UNPREDICTABLE** DISPLAY exception return instruction is unpredictable.

ArchMsg.Warning.warning_unpredictable_in_debug_state

TAGS **UNPREDICTABLE** DISPLAY Instruction is **UNPREDICTABLE** when executed in debug state.

ArchMsg.Warning.warning_unpredictable_mair_encoding

TAGS **UNPREDICTABLE** DISPLAY Memory attribute %{ATTR} has invalid MAIR encoding; this may lead to unpredictable results. Fields:

ATTR unsigned int

Memory attribute.

ArchMsg.Warning.warning_unpredictable_pmu_counter_access

TAGS **UNPREDICTABLE** DISPLAY Access to PMU counters from non-secure EL0 or EL1 is **UNPREDICTABLE** with MDCR_EL2.HPMN set to 0.

ArchMsg.Warning.warning_unpredictable_prioritization_breakpoint_match_vector_catch

TAGS **UNPREDICTABLE** DISPLAY Two events with the same priority occurred at the same instruction: Address Matching Vector Catch debug event, and Breakpoint debug event. It is constrained unpredictable which is taken.

ArchMsg.Warning.warning_unpredictable_stack_selection

TAGS **UNPREDICTABLE** DISPLAY **UNPREDICTABLE** state on context switch.

ArchMsg.Warning.warning_unpredictable_tsize_out_of_range

TAGS **UNPREDICTABLE** DISPLAY %{TCR}.T%{TTBR:d}SZ (%{TSIZE:d} bits) is out of range. Must be between 25 and %{TSIZE_MAX:d}. Fields:

TCR enum

Which TCR.

TSIZE signed int

Value of TSize.

TSIZE_MAX signed int

The maximum allowed value.

TTBR unsigned int

Which TTBR region.

ArchMsg.Warning.warning_unpredictable_unaligned_pc_as_base_register

TAGS **UNPREDICTABLE** DISPLAY The PC value must be word-aligned, otherwise the behavior of the instruction is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_unaligned_pop_stack

TAGS **UNPREDICTABLE** DISPLAY **CONSTRAINED UNPREDICTABLE** unaligned stack frame %{FRAME_ADDR:x} in exception-return. Fields:

FRAME_ADDR unsigned int

unaligned stack frame address.

ArchMsg.Warning.warning_unpredictable_unaligned_to_non_writeback

TAGS **UNPREDICTABLE** DISPLAY Unaligned Access to Non-Writeback-Cacheable Memory.

ArchMsg.Warning.warning_unpredictable_vmsa_memattrib

TAGS **UNPREDICTABLE** DISPLAY Unpredictable vmsa memory attribute with texcb value programmed to %{TEXCB} when tex remap is %{TEX_REMAP}. Fields:

TEXCB unsigned int

texcb value.

TEX_REMAP bool

use tex remap.

ArchMsg.Warning.warning_unpredictable_vtcr_t0sz_sl0

TAGS **UNPREDICTABLE** DISPLAY The combination of VTCR.TOSZ=%{TOSZ:d} and VTCR.SL0=%{SL0:d} is unpredictable. Fields:

SL0 signed int

VTCR.SL0, starting level for stage 2 translation table walks.

TOSZ signed int

VTCR.TOSZ, required input address range.

ArchMsg.Warning.warning_unpredictable_write_DHCSR

TAGS **UNPREDICTABLE** DISPLAY **UNPREDICTABLE** set of bit-changes in DHCSR. Fields:

NEW unsigned int

data attempted to be written.

OLD unsigned int

previous value.

ArchMsg.Warning.warning_unpredictable_write_SHCSR

TAGS **UNPREDICTABLE** DISPLAY **UNPREDICTABLE** setting a pending bit to 1 for an exception with priority >= execution priority.

ArchMsg.Warning.warning_unsupported_access_to_memory_mapped_register

TAGS **UNPREDICTABLE** DISPLAY Access to memory mapped register is unsupported.

ArchMsg.Warning.warning_user_jmcr_access

TAGS **UNPREDICTABLE** DISPLAY User %{WRITE:(write|read)} access to JMCR is unpredictable. Fields:

WRITE unsigned int

access is write.

ArchMsg.Warning.warning_wcr_mask_and_bas

TAGS **UNPREDICTABLE** DISPLAY DBGWCR_EL1.MASK is non-zero and BAS is not 0xff.

ArchMsg.Warning.warning_wcr_mask_reserved

TAGS **UNPREDICTABLE** DISPLAY DBGWCR_ELn.MASK is set to a reserved value.

ArchMsg.Warning.warning_wcr_non_configuous_bas

TAGS **UNPREDICTABLE** DISPLAY DBGWCR_EL1.BAS has non-contiguous set of ones.

ArchMsg.Warning.warning_wvr_masked_bit_not_zero

TAGS **UNPREDICTABLE** DISPLAY DBGWVRn_EL1.VA contains a bit masked by DBGWCRn_EL1.MASK that is not zero.

ArchMsg.Why.why_branch_target_exception

A branch target exception has occurred. DISPLAY branch target exception: target pc=%{PC}, target opcode=%{OPCODE}. PRIMARY KEY PC. Fields:

BTYPE unsigned int

pstate.BTYPE at time branch taken.

OPCODE unsigned int

opcode at branch target.

PC unsigned int

Virtual Address of branch target.

ArchMsg.Why.why_illegal_state

An illegal mode change has occurred. DISPLAY illegal mode change: %{MESSAGE}. Fields:

MESSAGE string

The message.

BRANCH_MISPREDICT

Simulating branch mispredict. Fields:

PC unsigned int

Origin address (or 0 if unavailable).

BRA_DIR

Direct branches, to immediate address. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISSET enum

The instruction set of the branch instruction.

IS_COND bool

Indicates if this is a conditional branch.

IS_HINTED bool

Whether the conditional instruction is hinted.

IS_ISB bool

Whether the instruction is ISB.

IS_LINK bool

Indicates if this is a branch with link.

PC unsigned int

The address of the branch instruction.

TARGET_ISET enum

The instruction set after the branch.

TARGET_PC unsigned int

The address the instruction branches to.

BRA_INDIR

Indirect branches, perhaps to a register. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISET enum

The instruction set of the branch instruction.

IS_COND bool

Indicates if this is a conditional branch.

IS_HINTED bool

Whether the conditional instruction is hinted.

IS_ISB bool

Whether the instruction is ISB.

IS_LINK bool

Indicates if this is a branch with link.

PC unsigned int

The address of the branch instruction.

TARGET_ISET enum

The instruction set after the branch.

TARGET_PC unsigned int

The address the instruction branches to.

BRA_INDIR_CCFAIL

Indirect branches fail, perhaps to a register.

BRA_RET_CCFAIL

Branch return fail, perhaps to a register.

CACHE_MAINTENANCE_OP

Cache Maintenance Operation. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

DATA unsigned int

Specified MVA or Set/way.

FUNCTION enum

Clean / Invalidate.

SCOPE enum

Affected region.

SIDE enum

Inst / Data.

CCFAIL

Conditional instruction condition check fail. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CCFAIL_UNC

Unconditional instruction condition check fail. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CCPASS

Conditional instruction condition check pass. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CCPASS_UNC

Unconditional instruction condition check pass. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CODE_CACHE_MAINT

Code cache maintenance. Fields:

END_INCL unsigned int

Inclusive end address for VA or PA requests.

NS bool

Non-secure world for PA.

START unsigned int

Start address for VA or PA requests.

TYPE enum

Request type.

COMPILE_BLOCK_END

Last instruction of basic block translated. Fields:

VADDR unsigned int

Address of next instruction after this basic block.

COMPILE_BLOCK_START

First instruction of basic block translated. Fields:

VADDR unsigned int

Address of first instruction in basic block.

COMPILE_INST

ARM instruction compiled. Fields:

DISASS string

Disassembly of the instruction.

ISSET enum

The instruction set of this instruction.

ITSTATE unsigned int

The ITSTATE current for the instruction.

OPCODE unsigned int

The opcode of the instruction.

PC unsigned int

The address of the instruction.

SIZE unsigned int

The size of the instruction in bytes.

CONTEXTIDR

Context ID Register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

NS bool

Secure or nonsecure banked register is accessed.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The new value written.

CONTEXT_SYNC

Called for every context synchronization event. Fields:

ARCH_CSE bool

Architectural Context Synchronization Event.

PC unsigned int

The address of the synchronization instruction.

SYSREG_SYNC bool

System register synchronization event.

CORE_ENDIAN

Core BE8 Big-Endian state changed. Fields:

BE8 bool

Core BE8 Big-Endian state.

CORE_INFO

Static processor attributes. Only triggered by a call to DumpState(). Fields:

ARCH_PROFILE enum

The architecture profile of the core.

CLUSTER_ID unsigned int

The cluster ID of this processor.

CORE_NUM unsigned int

The number of this core in an MP processor.

DETAILED_CORE_FEATURES enum

Detailed core features as a bitfield. Enum values are bitmasks to extract the info from the field.

FPU_VERSION enum

The VFP version implemented by the core.

MEM_ARCH enum

The memory architecture of the core.

MIDR_EL1 unsigned int

MIDR_EL1 value.

NUM_CORES unsigned int

The number of cores in this MP processor.

QUANTUM_SIZE unsigned int

The default quantum size of the core (in ticks).

SECURITY_FEATURES bool

Does the core have security features?.

CORE_LOADS

Processor load accesses. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ACQREL enum

Is this an acquire/release.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DATA unsigned int

The data read or written.

ELEMENT_SIZE unsigned int

Width of each element.

ENDIANNESS enum

Byte ordering of this access.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

LOCK enum

Normal, exclusive or locked access.

MECID unsigned int

Memory Encryption Context ID, if applicable.

MECID2 unsigned int

Memory Encryption Context ID of the second page, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PADDR_TAG unsigned int

If tag checked, the physical address tag for the access.

PAGE_SIZE unsigned int

Page size as $\log_2(\text{size in bytes})$.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK, 2=Exclusive Failed, 3=Unsupported.

SIZE unsigned int

Width of the access in bytes. Only required if DATA is not traced.

TAG_CHECKED bool

Is this access tag checked.

TAG_CHECKED2 bool

Is the second page access tag checked.

TRANS bool

Is this a translated access.

VADDR unsigned int

The virtual address of the access.

CORE_REGS

Changes of the core registers R0 to R14. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number, 0 to 14.

MODE enum

Bank of the register accessed.

OLD_VALUE unsigned int

The old value overwritten.

PHYS_ID enum

The physical register accessed.

VALUE unsigned int

The new value written to the register.

CORE_REGS64

Changes of the core registers X0..X30, SP_ELn. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ID enum

The register number, 0-30 for X0-X30, >=32 for SP_ELn.

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

CORE_REGS64_READ

General purpose AArch64 register has been read X0..X30, SP_ELn. Fields:

ID enum

The register number, 0-30 for X0-X30, >=32 for SP_ELn.

VALUE unsigned int

The value read from the register.

CORE_STALLED_TRANSITION

Detect when a core is prevented from progressing PChannel states. Fields:

CURRENT_STATE enum

Current power mode of core.

REASON string

The reason a transition was stalled, and what must be done to unblock it.

TARGET_STATE enum

Desired power mode of core after transition.

CORE_STORES

Processor store accesses. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ACQREL enum

Is this an acquire/release.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DATA unsigned int

The data read or written.

ELEMENT_SIZE unsigned int

Width of each element.

ENDIANNESS enum

Byte ordering of this access.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

LOCK enum

Normal, exclusive or locked access.

MECID unsigned int

Memory Encryption Context ID, if applicable.

MECID2 unsigned int

Memory Encryption Context ID of the second page, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PADDR_TAG unsigned int

If tag checked, the physical address tag for the access.

PAGE_SIZE unsigned int

Page size as $\log_2(\text{size in bytes})$.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK, 2=Exclusive Failed, 3=Unsupported.

SIZE unsigned int

Width of the access in bytes. Only required if DATA is not traced.

TAG_CHECKED bool

Is this access tag checked.

TAG_CHECKED2 bool

Is the second page access tag checked.

TRANS bool

Is this a translated access.

VADDR unsigned int

The virtual address of the access.

CP14_READ

System Coprocessor register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP14 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc1 unsigned int

opcode 1.

opc2 unsigned int

opcode 2.

CP14_WRITE

System Coprocessor register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP14 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc1 unsigned int

opcode 1.

opc2 unsigned int

opcode 2.

CP15_READ

System Coprocessor register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP15 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc1 unsigned int

opcode 1.

opc2 unsigned int

opcode 2.

CP15_READ64

System Coprocessor register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP15 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc unsigned int

opcode 1.

CP15_WRITE

System Control Coprocessor register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP15 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc1 unsigned int

opcode 1.

opc2 unsigned int

opcode 2.

CP15_WRITE64

System Coprocessor register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP15 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc unsigned int

opcode 1.

CPSR

CPSR change. (In AArch64 PSTATE represented in SPSR format). Fields:

CORE_NUM unsigned int

Core number in a multi processor.

OLD_VALUE unsigned int

The old CPSR value.

UNKNOWN unsigned int

Bits within the register that have unknown value.

VALUE unsigned int

The new CPSR value.

CRYPTO_SPEC

Every crypto instruction speculatively executed.

CorePowerStateChange

Core power-up state has changed. DISPLAY Core powered %{PU:(up|down)}. Fields:

PU bool

Whether the core powered up.

DATA_CACHE_SET_ALLOC_TAGS

Set allocation tags from DC GVA instruction. Fields:

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

NSDESC unsigned int

The security state of the access.

PADDR unsigned int

The physical address of the access.

PAS enum

The physical address space of the page.

RESPONSE enum

0=Aborted, 1=OK.

SIZE unsigned int

The size of memory in bytes.

TAG unsigned int

The value of the tag being set.

TAGGING_TYPE enum

Type of Memory tagging (Physical or Virtual).

TAG_VADDR unsigned int

The virtual address of the Virtual Tag if Virtual tagging is used; otherwise 0.

VADDR unsigned int

Virtual address used for setting allocation tags.

DATA_CACHE_ZERO

Zero and invalidate cache line from DC ZVA instruction. Fields:

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

NSDESC unsigned int

The security state of the access.

PADDR unsigned int

The physical address of the access.

PADDR_TAG unsigned int

If tag checked, the physical address tag for the access.

PAS enum

The physical address space of the page.

RESPONSE enum

0=Aborted, 1=OK.

SIZE unsigned int

The size of zero'd memory in bytes.

TAG_CHECKED bool

Is this access tag checked.

VADDR unsigned int

Virtual address of the zero'd cache line.

DEBUG_EVENT

Hardware debug support event. Fields:

EVENT enum

Description of event.

VALUE unsigned int

data value.

DEBUG_TARGET_ISET

Hardware debug target ISA. Fields:

TARGET_ISET enum

The target instruction set of the debug state.

DMI_ALLOCATE

DMI allocated. Fields:

ACCESS_DIRECTION enum

READ / WRITE.

PADDR unsigned int

The physical address of the DMI.

PTR unsigned int

DMI range pointer.

SIDE enum

DSIDE / ISIDE.

SIZE unsigned int

The size of DMI.

VADDR unsigned int

The virtual address of the DMI..

DMI_HIT

DMI hit, access used DMI (only fire for small DMI regions i.e. less than a full page). Fields:

ACCESS_DIRECTION enum

READ / WRITE.

PADDR unsigned int

The physical address of the DMI.

SIDE enum

DSIDE / ISIDE.

SIZE unsigned int

The size of the DMI.

VADDR unsigned int

The virtual address of the DMI..

DMI_REVOKE

DMI revoked. Fields:

ACCESS_DIRECTION enum

READ / WRITE.

PADDR unsigned int

The physical address of the DMI.

SIDE enum

DSIDE / ISIDE.

SIZE unsigned int

The size of DMI.

VADDR unsigned int

The virtual address of the DMI..

END_COMPILE

Compilation end. Fields:

END_OF_PAGE_COUNT unsigned int

Number of basic blocks exited due to page boundary since START_COMPILE.

FETCHFAIL_COUNT unsigned int

Number of basic blocks exited due to ifetch failure START_COMPILE.

INST_COUNT unsigned int

Number of instructions compiled since START_COMPILE.

NONSEQ_COUNT unsigned int

Number of basic blocks exited due to non-sequential instructions since START_COMPILE.

PAGE_STRADDLE_COUNT unsigned int

Number of basic blocks exited due to unaligned instructions crossing page since START_COMPILE.

ESB

Error Synchronization Barrier hit. Fields:

ASYNC bool

ESB is async.

IMPLICIT bool

ESB is implicit.

EXCEPTION

Exception taken. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ESR unsigned int

The value assigned to ESR when exception is taken.

ESR_EC unsigned int

The Value assigned to ESR.EC when exception is taken.

IS_PHYSICAL bool

Physical or Virtual exception.

IS_TRAP bool

Exception coming from a trap.

LR unsigned int

The value assigned to the link register.

PC unsigned int

The location where the exception occurred.

PREFERRED_RETURN unsigned int

The preferred return address for the exception.

TAKEN_LOCALLY bool

Whether exception is taken locally(same exception level).

TARGET_ISET enum

The instruction set of the exception handler code.

TARGET_PC unsigned int

The address the exception branches to.

VECTOR enum

The exception vector.

VECTOR_OFFSET unsigned int

The offset of an exception vector from the base address.

EXCEPTION_END

Every exception completed.

EXCEPTION_INFO

Trace human-readable reason of the current exception for data/instruction aborts, floating point exceptions and unknown exceptions. Fields:

DESCRIPTION string

Reason for current exception.

ESR_EC unsigned int

The value of ESR.EC when exception is taken.

EXCEPTION_RAISE

Every exception raised.

EXCEPTION_RETURN

Branches on leaving exception. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISSET enum

The instruction set of the branch instruction.

IS_COND bool

Indicates if this is a conditional branch.

IS_HINTED bool

Whether the conditional instruction is hinted.

IS_ISB bool

Whether the instruction is ISB.

IS_LINK bool

Indicates if this is a branch with link.

PC unsigned int

The address of the branch instruction.

TARGET_ISET enum

The instruction set after the branch.

TARGET_PC unsigned int

The address the instruction branches to.

EXCEPTION_RETURN_PREBRANCH

Exception return event that must occur before PC and processor state have updated.

EXCEPTION_START

Every exception started.

FIQ_TAKEN

FIQ taken exception.

FP_EXCEPTIONS_SIGNALLED

Trace which FP exceptions were signalled as a part of the FP operations performed by an instruction. Fields:

FP_EXCEPTIONS_SIGNALLED_MASK unsigned int

Indicates which FP exceptions were signalled by an instruction. This uses the same encoding as the cumulative floating-point exception bitfields in FPSR.

FP_STATE

Floating point state. Fields:

FPSR unsigned int

FPSR register value.

MASK unsigned int

The mask indicating updated cumulative bits.

FREQ_CHANGED

Core frequency changed. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The instruction count of this core.

NEW_FREQ signed int

The new frequency of this core (expressed in Hz).

OLD_FREQ signed int

The old frequency of this core (expressed in Hz).

HLT

HLT instruction occurred. Fields:

IMM16 unsigned int

The 16-bit immediate value encoded in HLT instruction.

INST

Every instruction executed. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DEBUG_STATE bool

The instruction is executed in debug state.

DISASS string

Disassembly of instruction.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISET enum

The current instruction set.

ITSTATE unsigned int

The current ITSTATE.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

MODE enum

The mode the core is in.

NS unsigned int

The core's non-secure bit.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

OPCODE unsigned int

The opcode of the instruction.

PADDR unsigned int

The physical address of the instruction.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the instruction.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the page.

PC unsigned int

The address of the instruction.

SECURITY_STATE enum

The core's security state.

SIZE unsigned int

The size of the instruction in bytes.

INST_END

Every instruction completed.

INST_START

Every instruction started. Fields:

DEBUG_STATE bool

The instruction is executed in debug state.

ISET enum

The current instruction set.

MODE enum

The mode the core is in.

NS enum

The current Secure State.

PC unsigned int

The address of the conditional instruction.

SECURITY_STATE enum

The current Security State.

INST_STRADDLE

Instruction straddles boundary. Fields:

DEBUG_STATE bool

The instruction is executed in debug state.

ISET enum

The current instruction set.

MODE enum

The mode the core is in.

NS enum

The current Secure State.

PC unsigned int

The address of the conditional instruction.

SECURITY_STATE enum

The current Security State.

IRQ_TAKEN

IRQ taken exception.

ISB

ISB instruction occurred. Fields:

PC unsigned int

The address of the ISB instruction.

LOAD_MULTIPLE_REGS_LIST

List of destination registers (not source) of a multiple registers load instruction. Fields:

REG string

Register to add into the list.

LOCAL_MONITOR

Local monitor activity. Fields:

PADDR unsigned int

Local Monitor Address.

State enum

State of the monitor (Open/Exclusive).

MAPPING_CREATE

A memory mapping has been created. This is an implementation detail of the model and so may change without notice. Fields:

ASID unsigned int

The ASID used during translation, or 0 if no ASID was used.

IS_VALID bool

Whether the mapping is valid or not.

PADDR unsigned int

Base physical address of region, or 0 if not known.

SIDE enum

I-side or D-side.

SIZE unsigned int

Size of the page, or 0 if not known.

VADDR unsigned int

Base virtual address of region.

VMID unsigned int

The VMID used during translation, or 0 if no VMID was used.

MAPPING_DESTROY

A memory mapping has been destroyed. This is an implementation detail of the model and so may change without notice. Fields:

ASID unsigned int

The ASID used during translation.

CAUSE enum

The cause of invalidation.

PADDR unsigned int

Base physical address of page.

SIDE enum

I-side or D-side.

SIZE unsigned int

Size of the page.

VADDR unsigned int

Base virtual address of page.

VMID unsigned int

The VMID used during translation.

MEMMAP_DEBUG_READ

Memory mapped reads to the debug registers. Fields:

ADDR unsigned int

address.

CORE_NUM unsigned int

Core number in a multi processor.

EXT bool

Whether access is from an external device (such as the DAP).

NS enum

Secure state of the access.

REG_NAME string

Name of the debug register accessed.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

MEMMAP_DEBUG_WRITE

Memory mapped writes to the debug registers. Fields:

ADDR unsigned int

address.

CORE_NUM unsigned int

Core number in a multi processor.

EXT bool

Whether access is from an external device (such as the DAP).

NS enum

Secure state of the access.

REG_NAME string

Name of the debug register accessed.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The new value written.

MEMTAG_LOADS

Allocation tag memory read; when attributes and system configuration allow tag accesses.

Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

NUM_GRANULES unsigned int

Number of tags to be Loaded/stored.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK.

TAGGING_TYPE enum

Type of Memory tagging (Physical or Virtual).

TAGS unsigned int

The tags read or written. Tags for subsequent granules would be found at subsequent bytes in this field (One byte for each tag).

TAG_VADDR unsigned int

The virtual address of the Virtual Tag if Virtual tagging is used; otherwise 0.

VADDR unsigned int

The virtual address of the access.

MEMTAG_LOAD_INST

Allocation tag load instruction; regardless of tags being actually accessed. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

NUM_GRANULES unsigned int

Number of tags to be Loaded/stored.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK.

TAGGING_TYPE enum

Type of Memory tagging (Physical or Virtual).

TAGS unsigned int

The tags read or written. Tags for subsequent granules would be found at subsequent bytes in this field (One byte for each tag).

TAG_VADDR unsigned int

The virtual address of the Virtual Tag if Virtual tagging is used; otherwise 0.

VADDR unsigned int

The virtual address of the access.

MEMTAG_STORES

Allocation tag memory write; when attributes and system configuration allow tag accesses.
Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

NUM_GRANULES unsigned int

Number of tags to be Loaded/stored.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK.

TAGGING_TYPE enum

Type of Memory tagging (Physical or Virtual).

TAGS unsigned int

The tags read or written. Tags for subsequent granules would be found at subsequent bytes in this field (One byte for each tag).

TAG_VADDR unsigned int

The virtual address of the Virtual Tag if Virtual tagging is used; otherwise 0.

VADDR unsigned int

The virtual address of the access.

MENTAG_STORE_INST

Allocation tag store instruction; regardless of tags being actually accessed. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

NUM_GRANULES unsigned int

Number of tags to be Loaded/stored.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK.

TAGGING_TYPE enum

Type of Memory tagging (Physical or Virtual).

TAGS unsigned int

The tags read or written. Tags for subsequent granules would be found at subsequent bytes in this field (One byte for each tag).

TAG_VADDR unsigned int

The virtual address of the Virtual Tag if Virtual tagging is used; otherwise 0.

VADDR unsigned int

The virtual address of the access.

MMU_D128_TTB_READ

This event is triggered by reads caused by a D128 translation table walk. Fields:

ABORTED bool

Was the walk successful.

DATA_HIGH unsigned int

The data upper bits read.

DATA_LOW unsigned int

The data lower bits read.

IPA unsigned int

For stage 1, the IPA of the read.

LEVEL unsigned int

Translation table level.

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

OUTADDR unsigned int

The output physical address.

PADDR unsigned int

The physical address of the read.

PASmem enum

Physical Address Space of the accessed memory.

PASreq enum

Physical Address Space of the requested lookup.

REGIME_EL enum

Entry matches in this translation regime.

REQUESTER enum

The requester of this table walk.

SIDE enum

Inst / Data.

STAGE unsigned int

Translation stage.

MMU_D128_TTB_WRITE

This event is triggered by writes caused by a D128 translation table walk. Fields:

ABORTED bool

Was the walk successful.

DATA_HIGH unsigned int

The data upper bits written.

DATA_LOW unsigned int

The data lower bits written.

IPA unsigned int

For stage 1, the IPA of the write.

IS_HACDBS_ACCESS bool

If the access was due to HACDBS_CLEAN.

LEVEL unsigned int

Translation table level.

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

OUTADDR unsigned int

The output physical address.

PADDR unsigned int

The physical address of the write.

PASmem enum

Physical Address Space of the accessed memory.

PASreq enum

Physical Address Space of the requested lookup.

REGIME_EL enum

Entry matches in this translation regime.

SIDE enum

Inst / Data.

STAGE unsigned int

Translation stage.

MMU_TRANS

Address translation information. Fields:

ASID unsigned int

Address space identifier.

CORE_NUM unsigned int

Core number in a multi processor.

Hyp bool

Entry matches in Hyp state only.

INNERCACHE_RA bool

Is the inner cache allocate on read.

INNERCACHE_TRANSIENT bool

Is the inner write-through transient.

INNERCACHE_TYPE enum

Inner Caching scheme (NC/MB/WA).

INNERCACHE_WA bool

Is the inner cache allocate on write.

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MEMTYPE enum

Memory type.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC enum

Is secure side supposed to access secure or nonsecure memory.

OUTERCACHE_RA bool

Is the outer cache allocate on read.

OUTERCACHE_TRANSIENT bool

Is the outer write-through transient.

OUTERCACHE_TYPE enum

Outer Caching scheme (NC/MB/WA).

OUTERCACHE_WA bool

Is the outer cache allocate on write.

PADDR unsigned int

Physical address of the access.

PAGESIZE unsigned int

Page size as $\log_2(\text{size})$.

PAS enum

PAS supposed to be accessed is secure or nonsecure memory or Root/Realm if RME is implemented.

SH enum

Shareability.

SIDE enum

Inst / Data.

STAGE1_PERM unsigned int

Stage 1 permissions, [5:3] Privileged access (XWR) [2:0] User access (XWR).

STAGE2_PERM unsigned int

Stage 2 permission mask [5:3] Privileged access (XWR) [2:0] User access (XWR).

VADDR unsigned int

Virtual address of the access.

VMID unsigned int

Virtual machine identifier.

XS bool

XS Attribute.

nG enum

Flag indicating whether ASID will be matched.

MMU_TTB_READ

This event is triggered by reads caused by a translation table walk. Fields:

ABORTED bool

Was the walk successful.

DATA unsigned int

The data read.

DESC_KIND enum

The encoding scheme of the descriptors used for this walk.

IPA unsigned int

For stage 1, the IPA of the read.

LEVEL unsigned int

Translation table level.

LPAE bool

Is this for an LPAE translation (*DEPRECATED*, use DESC_KIND).

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSmem bool

Non secure memory access.

NSreq bool

Non secure request.

OUTADDR unsigned int

The output physical address.

PADDR unsigned int

The physical address of the read.

PASmem enum

Physical Address Space of the accessed memory.

PASreq enum

Physical Address Space of the requested lookup.

REGIME_EL enum

Entry matches in this translation regime.

REQUESTER enum

The requester of this table walk.

SIDE enum

Inst / Data.

STAGE unsigned int

Translation stage.

MMU_TTB_WRITE

This event is triggered by writes caused by a translation table walk. Fields:

ABORTED bool

Was the walk successful.

DATA unsigned int

The data written.

DESC_KIND enum

The encoding scheme of the descriptors used for this walk.

IPA unsigned int

For stage 1, the IPA of the write.

IS_HACDBS_ACCESS bool

If the access was due to HACDBS_CLEAN.

LEVEL unsigned int

Translation table level.

LPAE bool

Is this for an LPAE translation (*DEPRECATED*, use DESC_KIND).

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSmem bool

Non secure memory access.

NSreq bool

Non secure request.

OUTADDR unsigned int

The output physical address.

PADDR unsigned int

The physical address of the write.

PASmem enum

Physical Address Space of the accessed memory.

PASreq enum

Physical Address Space of the requested lookup.

REGIME_EL enum

Entry matches in this translation regime.

SIDE enum

Inst / Data.

STAGE unsigned int

Translation stage.

MODE_CHANGE

Mode change. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

MODE enum

The new mode.

NON_SECURE enum

The core's new non-secure bit.

OLD_MODE enum

The old mode.

SECURITY_STATE enum

The new security state.

NOT_TAG_CHECKED_INST

MTE Tag not checked for current instruction.

PERIODIC

Called for every quantum. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The instruction count of this CPU.

PC unsigned int

The address of the next instruction to be executed on this CPU.

PMU_COUNTER_OVERFLOW

PMU counter overflow. Fields:

COUNTER_GROUP enum

Counter group.

COUNTER_RANGE enum

Counter range.

EVENT_ID unsigned int

Event Identifier, field is valid only for Event counter group.

INDEX unsigned int

Counter index (as selected by PMSELR), field is valid only for Event/Cycle counter groups.

INTERRUPT bool

Is interrupt enabled on overflow for this counter.

POINTER_AUTH

Called when execution hits a pointer authentication check. Fields:

AUTH_STATE bool

Authentication result.

KEY_USED enum

Pointer authentication key.

POTINDEX_UPDATE

Value of POT index for current pc changed. Fields:

NEW_POTINDEX unsigned int

New POTIndex.

OLD_POTINDEX unsigned int

Old POTIndex.

PREFETCH_MEMORY64

Prefetch from PRFM or PRFUM instructions. Fields:

PRFOP enum

Prefetch hint.

VADDR unsigned int

Virtual address of the location that should be prefetched.

PRELOAD64_REQUEST

Preload requests from PRFM or PRFUM instructions. Fields:

PRFOP enum

Prefetch hint.

VADDR unsigned int

Virtual address of the location that should be prefetched.

PRELOAD_DATA

Data preload from PLD instruction. Fields:

VADDR unsigned int

Virtual address of the data that should be preloaded.

PRELOAD_INST

Instruction preload from PLI instruction. Fields:

VADDR unsigned int

Virtual address of the instruction that should be preloaded.

PRE_CORE_LOAD

Trace just before a core load. Fields:

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

PRE_CORE_STORE

Trace just before a core store. Fields:

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

PRE_TTB_READ

This event is triggered before reads caused by a translation table walk. Fields:

ADDR unsigned int

The physical address of the read.

PSTATE

PSTATE fields update. Fields:

ALLINT unsigned int

All interrupt mask bit.

BTYPE unsigned int

Branch target identification bit.

CORE_NUM unsigned int

Core number in a multi processor.

DAIF unsigned int

Exception mask bits.

DIT unsigned int

Data Independent Timing bit.

EL unsigned int

Current Exception level field.

EXLOCK unsigned int

Exception return state lock bit.

IL unsigned int

Illegal Execution state bit.

NZCV unsigned int

Condition flags.

PACM unsigned int

PACM bit.

PAN unsigned int

Privileged Access Never state bit.

PM unsigned int

PMU exception mask bit.

PPEND unsigned int

PMU exception pending bit.

SM unsigned int

StreamingMode status bit.

SP unsigned int

Stack pointer register selection bit.

SS unsigned int

Software Step bit.

SSBS unsigned int

Speculative Store Bypass Safe bit.

TCO unsigned int

Tag Check Override bit.

TOUCHED unsigned int

Mask of the fields that have been touched since this trace has last been emitted. Note that touched does not necessarily mean the value changed. The index of the field is the bit index in the mask plus the index of this field plus one.

UAO unsigned int

User Access Override bit.

UINJ unsigned int

Undefined Instruction exceptions bit.

ZA unsigned int

ZA storage and ZT0 register enable bit.

nRW unsigned int

Current Execution state bit.

PSTATE_PACTIVE_OPMODE_TRANSITION

Trace pactive and Opmode mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Operating mode after the transition.

OLD_STATE enum

Operating mode before the transition.

PSTATE_PACTIVE_OPMODE_TRANSITION_DENIED

Trace the denied pactive and Opmode mode transitions on PChannel. Fields:

DEVPACTIVE **unsigned int**

Integer representation of DEVPACTIVE[] signals.

NEW_STATE **enum**

Operating mode after the transition.

OLD_STATE **enum**

Operating mode before the transition.

REASON **string**

Reason for the transition being denied.

PSTATE_PACTIVE_POWER_TRANSITION

Trace pactive and power mode transitions on PChannel. Fields:

DEVPACTIVE **unsigned int**

Integer representation of DEVPACTIVE[] signals.

NEW_STATE **enum**

Power mode after the transition.

OLD_STATE **enum**

Power mode before the transition.

PSTATE_PACTIVE_POWER_TRANSITION_DENIED

Trace the denied pactive and power mode transitions on PChannel. Fields:

DEVPACTIVE **unsigned int**

Integer representation of DEVPACTIVE[] signals.

NEW_STATE **enum**

Power mode after the transition.

OLD_STATE **enum**

Power mode before the transition.

REASON **string**

Reason for the transition being denied.

PSTATE_PACTIVE_THREAD_OPMODE_TRANSITION

Trace pactive and Opmode mode transitions for threads on PChannel. Fields:

DEVPACTIVE **unsigned int**

Integer representation of DEVPACTIVE[] signals.

NEW_STATE **enum**

Operating mode after the transition.

OLD_STATE **enum**

Operating mode before the transition.

QUANTUM_START

Called at the start of every quantum.

RANGE_PRELOAD_REQUEST

Preload requests from RPRFM. Fields:

COUNT unsigned int

Number of blocks of data to access.

LENGTH unsigned int

Number of contiguous bytes to be accessed in each block.

OP unsigned int

Range prefetch operation .

REUSE unsigned int

Maximum number of bytes to be accessed before next RPRFM .

STRIDE signed int

Stride to advance after accessing LENGTH bytes.

VADDR unsigned int

Base virtual address of range prefetch request.

RUN_STATE

Run state transition. Fields:

INST_COUNT unsigned int

Ticks count at point of transition.

NEW enum

New run state.

OLD enum

Old run state.

SEMIHOSTING_CALL

Call of a semihost function occurred. Fields:

PC unsigned int

The program counter after the semihosting call.

SEMIHOSTING_PRECALL

About to call semihost. Fields:

EL2 bool

Use EL2 translation regime.

NS bool

Is Non-Secure.

REG_WIDTH unsigned int

The current register width in bytes.

SIGNAL

External signal state change. Fields:

SIGNAL *enum*

Signal that changed.

STATE *bool*

Signal asserted state.

SOFTWARE_STEP

Return the Debug Software Step state. Fields:

STATE *enum*

Software Step state.

SPSR

SPSR change. Fields:

CORE_NUM *unsigned int*

Core number in a multi processor.

MODE *enum*

Which of the banked SPSR registers is written.

OLD_VALUE *unsigned int*

The old SPSR value.

VALUE *unsigned int*

The new SPSR value.

START_COMPILE

Compilation started. Fields:

VADDR *unsigned int*

Instruction where compilation begins.

STG_MULTIPLE_MICRO_OPS_ADDRESS_RANGE

Address range for MTE stores with multiple uops. Fields:

ADDR_END *unsigned int*

Final address.

ADDR_START *unsigned int*

Starting address.

STORE_MULTIPLE_REGS_ADDRESS_RANGE

Address range for stores with multiple registers. Fields:

ADDR_END *unsigned int*

Final address.

ADDR_START *unsigned int*

Starting address.

SUBOPTIMAL_LDST_RETIRED

suboptimal load/store. Fields:

ACCESS_SIZE unsigned int

Log2 of the access width in bytes used for alignment checking (1: half-word, 2: word ...).

OPTIMAL_ALIGN_SELECTOR unsigned int

Selector to decide the optimal alignment size (1: access width, 2: 4-bytes, 3: 8-bytes, ..., 12: 4-kbytes).

VADDR unsigned int

The virtual address of the access.

SVE_INST_SPEC

SVE operations speculatively executed.

SVE_LD_RETIRED

SVE load instructions architecturally executed.

SVE_LOADS

SVE load accesses. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ACQREL enum

Is this an acquire/release.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DATA unsigned int

The data read or written.

ELEMENT_SIZE unsigned int

Width of each element.

ENDIANNESS enum

Byte ordering of this access.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

LOCK enum

Normal, exclusive or locked access.

MECID unsigned int

Memory Encryption Context ID, if applicable.

MECID2 unsigned int

Memory Encryption Context ID of the second page, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PADDR_TAG unsigned int

If tag checked, the physical address tag for the access.

PAGE_SIZE unsigned int

Page size as $\log_2(\text{size in bytes})$.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK, 2=Exclusive Failed, 3=Unsupported.

SIZE unsigned int

Width of the access in bytes. Only required if DATA is not traced.

TAG_CHECKED bool

Is this access tag checked.

TAG_CHECKED2 bool

Is the second page access tag checked.

TRANS bool

Is this a translated access.

VADDR unsigned int

The virtual address of the access.

SVE_MEM_PREFETCH

Scalable Vector Extension memory prefetch. Fields:

CACHE_LEVEL enum

The preferred cache level.

HINT enum

What the memory is expected to be used for.

SIZE unsigned int

The prefetch size in bytes.

TEMPORAL bool

Whether the data is temporal.

VADDR unsigned int

The virtual address of the prefetch.

SVE_REG_READ

Scalable Vector Extension register read. Fields:

ID enum

The register identifier.

SM bool

Whether the PE is in Streaming Mode.

VALUE unsigned int

The read data.

SVE_REG_UPDATE

Scalable Vector Extension register update. Fields:

ID enum

The register identifier.

SM bool

Whether the PE is in Streaming Mode.

VALUE unsigned int

The written data.

SVE_REG_WRITE

Scalable Vector Extension register write. Fields:

ID enum

The register identifier.

SM bool

Whether the PE is in Streaming Mode.

VALUE unsigned int

The written data.

SVE_STORES

SVE store accesses. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ACQREL enum

Is this an acquire/release.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DATA unsigned int

The data read or written.

ELEMENT_SIZE unsigned int

Width of each element.

ENDIANNESS enum

Byte ordering of this access.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

LOCK enum

Normal, exclusive or locked access.

MECID unsigned int

Memory Encryption Context ID, if applicable.

MECID2 unsigned int

Memory Encryption Context ID of the second page, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PADDR_TAG unsigned int

If tag checked, the physical address tag for the access.

PAGE_SIZE unsigned int

Page size as $\log_2(\text{size in bytes})$.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK, 2=Exclusive Failed, 3=Unsupported.

SIZE unsigned int

Width of the access in bytes. Only required if DATA is not traced.

TAG_CHECKED bool

Is this access tag checked.

TAG_CHECKED2 bool

Is the second page access tag checked.

TRANS bool

Is this a translated access.

VADDR unsigned int

The virtual address of the access.

SVE_ST_RETIRED

SVE store instructions architecturally executed.

SVE_UNALIGNED_LDST_RETIRED

SVE load or store instructions architecturally executed that would cause alignment fault if A bit is set.

SVE_Z_PRED_MEM_ACCESS_INFO

Information about SVE/SME predicated accesses to Z registers (non Gather/Scatter). Fields:

MEM_ELEM_SIZE unsigned int

Width of elements accessed in memory (in bytes).

RnW bool

Whether this is a Read not a Write.

VA unsigned int

Virtual base address computed for accesses.

SYNC

Called for every synchronization, that is every quantum break. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The instruction count of this CPU.

LOCAL_QUANTUM unsigned int

The local quantum of this CPU.

LOCAL_TIME unsigned int

The local time of this CPU.

SYNC_REASON enum

Reason for sync.

SYSCALL

System call instruction executed. Fields:

IMM unsigned int

Immediate value of the system call instruction.

TYPE enum

System call type.

VADDR unsigned int

Instruction that caused the system call.

SYSREG_READ128

System Coprocessor 128 bit register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

REGNUM enum

Internal register number.

REGNUM_PHYS enum

Internal register number physically accessed (e.g. in presense of VHE redirection).

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_READ64

System Coprocessor register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REGNUM enum

Internal register number.

REGNUM_OPERAND enum

Register number corresponding to the instruction operand.

REGNUM_PHYS enum

Internal register number physically accessed (e.g. in presense of VHE redirection).

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc unsigned int

opcode 1.

opc0 unsigned int
opcode 0.

opc2 unsigned int
opcode 2.

SYSREG_UPDATE128

Triggers when the system updates a register. Fields:

CRm unsigned int
CRm.

CRn unsigned int
CRn.

NS bool
Secure or nonsecure banked register is accessed.

REG enum
Register number.

UNKNOWN unsigned int
Bits of the register which became unknown.

VALUE unsigned int
Value written to the register.

opc unsigned int
opcode 1.

opc0 unsigned int
opcode 0.

opc2 unsigned int
opcode 2.

SYSREG_UPDATE32

Triggers when the system updates a register. Fields:

CRm unsigned int
CRm.

CRn unsigned int
CRn.

NS bool
Secure or nonsecure banked register is accessed.

REG enum
Register number.

UNKNOWN unsigned int
Bits of the register which became unknown.

VALUE unsigned int

Value written to the register.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_UPDATE64

Triggers when the system updates a register. Fields:

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG enum

Register number.

UNKNOWN unsigned int

Bits of the register which became unknown.

VALUE unsigned int

Value written to the register.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_WRITE128

System Coprocessor 128 bit register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

REGNUM enum

Internal register number.

REGNUM_PHYS enum

Internal register number physically accessed (e.g. in presense of VHE redirection).

UNDEF bool

The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_WRITE64

System Coprocessor register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REGNUM enum

Internal register number.

REGNUM_OPERAND enum

Register number corresponding to the instruction operand.

REGNUM_PHYS enum

Internal register number physically accessed (e.g. in presense of VHE redirection).

UNDEF bool

The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

TRBU_BUFFER_WRAP

DISPLAY Trace buffer current write pointer wrapped.

TRBU_TRIGGER_EVENT

DISPLAY Trace buffer Trigger Event.

TRBU_WRITE

DISPLAY PETU trace data has been written to the TRBU memory buffer at `%{BUFFER_ADDRESS}` with size `%{SIZE}` and data `%{DATA}`. Fields:

BUFFER_ADDRESS unsigned int

Virtual Address of start of buffer write.

DATA unsigned int

Data written to the buffer.

SIZE unsigned int

Size of sample written to buffer, in bytes.

UNALIGNED_LDST_RETIRED

Processor unaligned load/store. Fields:

ACCESS_SIZE unsigned int

Log2 of the access width in bytes used for alignment checking (1=>half-word, 2->word ...).

NUMBER_OF_BEATS unsigned int

Number of accesses (beats) in this burst (Total size in bytes is `ACCESS_SIZE * NUMBER_OF_BEATS`).

VADDR unsigned int

The virtual address of the access.

VFP_D_REGS

VFP/NEON D 64 bit register write. Fields:

ALIASING enum

Type of register aliasing used.

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a write to the corresponding bit in the register. For example, 0x1 would indicate a write to VALUE[0].

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

VFP_D_REGS_READ

VFP/NEON D 64 bit register read. Fields:

ID unsigned int

The register number.

VALUE unsigned int

The current value of the register.

VFP_Q_REGS

VFP/NEON Q 128 bit register write. Fields:

ALIASING enum

Type of register aliasing used.

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a write to the corresponding byte in the register. For example, 0x1 would indicate a write to VALUE[7:0].

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

VFP_Q_REGS_READ

VFP/NEON Q 128 bit register read. Fields:

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a read to the corresponding byte in the register. For example, 0x1 would indicate a read to VALUE[7:0].

VALUE unsigned int

The current value of the register.

VFP_SYS_REGS

Writes to the VFP/NEON units system registers. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ID enum

Which VFP system register is written.

MASK unsigned int

The mask indicating updated cumulative bits.

OLD_VALUE unsigned int

The register's old value overwritten.

VALUE unsigned int

The new value written to the VFP system register.

VFP_S_REGS

VFP/NEON S 32 bit register write. Fields:

ALIASING enum

Type of register aliasing used.

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a write to the corresponding bit in the register. For example, 0x1 would indicate a write to VALUE[0].

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

VFP_S_REGS_READ

VFP/NEON S 32 bit register read. Fields:

ID unsigned int

The register number.

VALUE unsigned int

The current value of the register.

VIRTUAL_TAG_ALLOCATED

Virtual tag information when allocated. Fields:

TAG_VADDR **unsigned int**

Virtual tag virtual address.

VADDR **unsigned int**

Data Virtual address.

VIRTUAL_TAG **unsigned int**

Virtual tag value.

WAYPOINT

Signals end of basic block execution. Fields:

CORE_NUM **unsigned int**

Core number in a multi processor.

ISET **enum**

Origin instruction set.

IS_COND **bool**

Indicates if this is a conditional waypoint.

PC **unsigned int**

Origin address (or 0 if unavailable).

TAKEN **bool**

Indicates if this waypoint was taken.

TARGET **unsigned int**

Destination address.

TARGET_ISET **enum**

Destination instruction set.

WFE_END

WFE ended. Fields:

INST_COUNT **unsigned int**

Ticks count when leaving WFE.

WFE_EVENT_REGISTER

WFE event register status: set/clear, reason. Fields:

INST_COUNT **unsigned int**

Ticks count.

REASON **enum**

Reason for set/clear. Only REASON==0 (Cleared by WFE) clears the bit, all other reasons set it.

WFE_IGNORED

WFE ignored. Fields:

EVENT bool

This WFE was ignored because the event register was set.

INST_COUNT unsigned int

Ticks count when ignoring WFE.

TRAPPED bool

This WFE was trapped.

WFE_START

WFE entered. Fields:

INST_COUNT unsigned int

Ticks count when entering WFE.

WFI_END

WFI ended. Fields:

INST_COUNT unsigned int

Ticks count when leaving WFI.

WFI_IGNORED

WFI ignored. Fields:

DISABLED bool

This WFI was ignored because WFI is disabled.

INST_COUNT unsigned int

Ticks count when ignoring WFI.

REASON enum

specifies reason why WFI trace was ignored.

TRAPPED bool

This WFI was trapped.

WFI_START

WFI entered. Fields:

INST_COUNT unsigned int

Ticks count when entering WFI.

WFI_WAKEUP

WFI wakeup. Fields:

INST_COUNT unsigned int

Ticks count when WFI wakeup occurred.

REASON enum

Reason for wakeup.

2.17 ARM_Cortex-A520

This section describes the trace sources.

AA64_ASE_SVE_REGS

Modification of AArch64 Advanced SIMD or SVE Vector Registers (via SIMD or Scalar operations), SVE Predicate Registers and ZT0. Fields:

ID **enum**

The register identifier.

MASK **unsigned int**

Bitmask of the register to signal the modified bits in the VALUE field.

SM **bool**

Whether the PE is in Streaming Mode.

VALUE **unsigned int**

Value of AArch64 vector register.

ASYNC_MEMORY_FAULT

Context ID Register write. Fields:

FAULT **unsigned int**

Fault status present in ESR.ISS encoding for Serror in ARCH64 and present in DFSR encoding for external abort format in ARCH32.

PADDR **unsigned int**

Physical Address (or 0 if unavailable).

VADDR **signed int**

Virtual Address (or 0 if unavailable).

ATOMIC_END_ACCESS

Bus trace access for atomics. Fields:

ACCESS_FAIL **bool**

Memory access failed.

ADDR **unsigned int**

The virtual address of the access.

ATTR **unsigned int**

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

COMPARE_VALUE **unsigned int**

Compare value for CAS.

LOAD_VALUE unsigned int

Loaded values.

NSDESC unsigned int

The security state of the access.

OPERAND_VALUE unsigned int

Operation's operand.

OPERATION enum

Operation type.

PADDR unsigned int

The physical address of the access.

PRIV bool

Is this a privileged access?.

ATOMIC_START_ACCESS

Bus trace access for atomics. Fields:

ADDR unsigned int

The virtual address of the access.

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

COMPARE_VALUE unsigned int

Compare value for CAS.

NSDESC unsigned int

The security state of the access.

OPERAND_VALUE unsigned int

Operation's operand.

OPERATION enum

Operation type.

PADDR unsigned int

The physical address of the access.

PRIV bool

Is this a privileged access?.

ArchMsg.Error.exit_code

DISPLAY exit code: %{EXITCODE} component kind: %{KIND} reason: %{REASON}. Fields:

EXITCODE signed int

Exit code to be returned.

KIND string

Component kind that invoked the exit code trace.

REASON string

Reason why the exit code trace was invoked (optional).

ArchMsg.Warning.IgnoredWarmResetRequest

Event fires when software attempts to request warm reset through write to deprecated DBGPRCR.CWRR.

ArchMsg.Warning.RAS_ErrselrUnpredictableState

DISPLAY ERRSELR is in unpredictable state due to value greater than number_of_error_records written to it.

ArchMsg.Warning.RAS_UnpredictableStateWrite

DISPLAY %{REG_NAME} write ignored due to unpredictable ERRSELR state. Fields:

REG_NAME string

Name of register being written.

ArchMsg.Warning.ReportedRevisionRegisterRead

DISPLAY Reported revision and patch do not match that of the cpu. Fields:

ORIGINAL_VALUE unsigned int

The value the register would have based on the modelled revision.

REGISTER string

The name of the register that has ben read.

REPORTED_VALUE unsigned int

The value the register has been overridden to.

ArchMsg.Warning.Unpredictably Indexed PM Event Register

TAGS UNPREDICTABLE DISPLAY Event %{IsDirect:(Selector value|Register)} %{SEL} is >= %{N}. Fields:

IsDirect unsigned int

Direct Access rather than indirect.

N unsigned int

Number of Accessible Registers.

SEL unsigned int

Selector.

ArchMsg.Warning.cache_contents_unknown

A cache has been activated without being invalidated and may contain **UNKNOWN** content. DISPLAY %{SIDE:(D|I)}-Cache was enabled in %{REGIME:(SECURE|NON_SECURE|HYP)} regime but was not invalidated since power-on: cache lines could contain **UNKNOWN** content %{SIDE:(Note: D-side caches must invalidate SECURE lines even if the cache is only used by NON_SECURE code in order to ensure that unexpected dirty lines tagged SECURE are not naturally evicted)}. Fields:

REGIME unsigned int

Current translation regime.

SIDE unsigned int

0 d-side cache or unified cache, 1 i-side.

ArchMsg.Warning.dcimvac_matches_watchpoint

TAGS **UNPREDICTABLE** DISPLAY Watchpoints matching an AArch32 DCIMVA are implementation defined.

ArchMsg.Warning.decode_fieldmismatch

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with different fields %{FIELD1} and %{FIELD2}. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_fracbitsnegative

TAGS **UNPREDICTABLE** DISPLAY VCVT Instruction is unpredictable with a negative number of fraction bits.

ArchMsg.Warning.decode_initblock

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable in an IT block.

ArchMsg.Warning.decode_initblocknotlast

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable in an IT block when not the last.

ArchMsg.Warning.decode_invalid_condition

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable when the condition is not AL.

ArchMsg.Warning.decode_invalidfieldcombination

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with the values in %{FIELD1} and %{FIELD2}. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_invalidvalue

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with the value in %{FIELD}. Fields:

FIELD enum

Field name.

ArchMsg.Warning.decode_mem_hint_unallocated

TAGS **UNPREDICTABLE** DISPLAY There is no memory hint allocated to this bit pattern.

ArchMsg.Warning.decode_registermatch

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with %{FIELD1} and %{FIELD2} as the same register. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_registermismatch

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with %{FIELD1} and %{FIELD2} as different registers. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_registeroutofrange

TAGS **UNPREDICTABLE** DISPLAY Floating-point instruction is unpredictable indexing beyond the end of the register bank.

ArchMsg.Warning.decode_sbzsbo

TAGS **UNPREDICTABLE** DISPLAY Reserved bits in the instruction are not canonical.

ArchMsg.Warning.decode_transactiontoobig

TAGS **UNPREDICTABLE** DISPLAY Attempt to perform a memory transaction of more than 128 bytes.

ArchMsg.Warning.decode_unpred_other

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable.

ArchMsg.Warning.decode_unpreduseofpc

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable when %{FIELD} is PC. Fields:

FIELD enum

Field name.

ArchMsg.Warning.decode_unpreduseofr13

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable when %{FIELD} is R13. Fields:

FIELD enum

Field name.

ArchMsg.Warning.decode_unpreduseofr13inreglist

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with R13 in register-list.

ArchMsg.Warning.decode_useofd16orhigher

TAGS **UNPREDICTABLE** DISPLAY Attempt to use register D16 or higher where it is not allowed.

ArchMsg.Warning.decode_writebackandbaseinlist

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with write back when the base register is in the transfer list.

ArchMsg.Warning.decode_zeroregistersinlist

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable when no registers are to be transferred.

ArchMsg.Warning.misaligned_ptrs

DISPLAY TRBPTR_EL1(%{TRBPTR_EL1}) or TRBTRG_EL1(%{TRBTRG_EL1}) are not aligned as per TRBIDR_EL1.Align(%{TRBIDR_EL1}). Fields:

TRBIDR_EL1 unsigned int

TRBIDR_EL1.Align value.

TRBPTR_EL1 unsigned int

Current write pointer.

TRBTRG_EL1 unsigned int

Trigger pointer.

ArchMsg.Warning.out_of_range_ptrs

DISPLAY TRBPTR_EL1(%{TRBPTR_EL1}) is out of range as per TRBBASER_EL1(%{TRBBASER_EL1}) or TRBLIMITR_EL1.LIMIT(%{TRBLIMITR_EL1}). Fields:

TRBBASER_EL1 unsigned int

Base pointer.

TRBLIMITR_EL1 unsigned int

Limit pointer.

TRBPTR_EL1 unsigned int

Current write pointer.

ArchMsg.Warning.recursive_branch

An instruction is performing a branch that targets the same instruction. If this was intended then a WFI might be more effective DISPLAY recursive branch detected at PC=%{PC}.

PRIMARY KEY PC. Fields:

PC unsigned int

Address of instruction causing the branch.

ArchMsg.Warning.recursive_exception

An instruction has generated an exception that targets the same instruction. DISPLAY recursive %{TYPE} exception detected at PC=%{PC} (CPSR=%{CPSR}). PRIMARY KEY PC. Fields:

CPSR unsigned int

Processor state.

PC unsigned int

Address of instruction causing the exception.

TYPE enum

Type of exception.

ArchMsg.Warning.reentrant_vector_catch

TAGS **UNPREDICTABLE** DISPLAY Vector catch debug event on Prefetch or Data abort vector %{ADDR}. Fields:

ADDR unsigned int

vector address.

ArchMsg.Warning.reserved_it_state

TAGS **UNPREDICTABLE** DISPLAY Execution with reserved IT state %{ITSTATE}. Fields:

ITSTATE unsigned int

reserved IT state.

ArchMsg.Warning.tlb_contents_unknown

Translation has been enabled without invalidating the TLB. DISPLAY Translation was enabled in the %{PAS:(Secure|Non-secure|Root|Realm)} EL%{EL:d} translation regime (vmid=%{VMID:x}) without invalidating %{INVALIDITY:(|d-side |i-side |)}entries since power-on: %{INVALIDITY:(|D|I|)}TLB could contain **UNKNOWN** content. Fields:

EL unsigned int

Exception level of translation regime.

INVALIDITY unsigned int

Which TLB may be invalid (if TLBs are separate).

NON_SECURE bool

Whether the translation regime is non-secure.

PAS unsigned int

Physical address space of translation regime.

VMID unsigned int

Current VMID, if applicable.

ArchMsg.Warning.unknown_DLR_DSPSR

Exiting debug state with unknown DLR or DSPSR.

ArchMsg.Warning.unknown_ELR_SPSR

Returning from debug exception with unknown ELR or SPSR.

ArchMsg.Warning.unpred_FAR_on_LRCPC3_non_atomic_mem_access

TAGS **UNPREDICTABLE** Unpredictable FAR/FSTATUS on LRCPC3 pair instructions resulting in two separate memory accessesDISPLAY Unpredictable FAR on LRCPC3 non atomic memory access at %{ADDR1} with FSTATUS %{FSTATUS1} or %{ADDR2} with FSTATUS %{FSTATUS2}. Fields:

ADDR1 unsigned int

Possible FAR address 1.

ADDR2 unsigned int

Possible FAR address 2.

FSTATUS1 unsigned int

Possible fault status value 1.

FSTATUS2 unsigned int

Possible fault status value 2.

ArchMsg.Warning.unpred_async_tag_check_fail_on_sve_speculative_load_do_update_ffr

Unpredictable set to FALSE of the FFR element associated to SVE speculative load with an async tag check fail TAGS **UNPREDICTABLE**.

ArchMsg.Warning.unpredictable_a32_breakpoint

TAGS **UNPREDICTABLE** DISPLAY Unpredictable breakpoint on a A32 instruction at %{ADDR1} Offset %{OFFSET} BAS %{BAS} for a %{IS_ADDRESS_MISMATCH_BREAKPOINT:(Mismatch|Match)} Breakpoint number %{N:d} Breakpoint address %{ADDR2}. Fields:

ADDR1 unsigned int

Address.

ADDR2 unsigned int

Breakpoint address.

BAS unsigned int

BAS field.

IS_ADDRESS_MISMATCH_BREAKPOINT bool

Is a Mismatch breakpoint, not a Match breakpoint.

N unsigned int

Breakpoint number.

OFFSET unsigned int

Always has a value of zero.

ArchMsg.Warning.unpredictable_banked_register_access

TAGS **UNPREDICTABLE** DISPLAY Access to banked register (R=%{R:d} SYSm=%{SYSM:2x}) is unpredictable. Fields:

R unsigned int

R bit.

SYSM unsigned int

SYSm.

ArchMsg.Warning.unpredictable_ccfail_undef

TAGS **UNPREDICTABLE** DISPLAY An instruction %{OPCODE} at VA=%{VADDR} failed its condition codes check but would otherwise have caused a non-data-dependent trap or undefined exception. Behaviour is implementation-defined choice of exception or nop. Fields:

OPCODE unsigned int

opcode of the instruction.

VADDR unsigned int

Virtual address of the instruction being executed.

ArchMsg.Warning.unpredictable_ici_bits

TAGS **UNPREDICTABLE** DISPLAY ICI bits do not apply or are architecturally unpredictable for instruction.

ArchMsg.Warning.unpredictable_t32_breakpoint

TAGS **UNPREDICTABLE** DISPLAY Unpredictable breakpoint on a T32 instruction at %{ADDR1} Offset %{OFFSET} BAS %{BAS} for a %{IS_ADDRESS_MISMATCH_BREAKPOINT:(Mismatch|Match)} Breakpoint number %{N:d} Breakpoint address %{ADDR2}. Fields:

ADDR1 unsigned int

Address.

ADDR2 unsigned int

Breakpoint address.

BAS unsigned int

BAS field.

IS_ADDRESS_MISMATCH_BREAKPOINT bool

Is a Mismatch breakpoint, not a Match breakpoint.

N unsigned int

Breakpoint number.

OFFSET unsigned int

Indicates which halfword of a 32-bit Thumb instruction the breakpoint is set on. 0 for the first halfword or if the instruction is 16 bits wide, or 1 for the second halfword.

ArchMsg.Warning.unpredictable_t32_vector_catch

TAGS **UNPREDICTABLE** DISPLAY Unpredictable Vector catch exception on a T32 instruction at %{ADDR} Offset %{OFFSET}. Fields:

ADDR unsigned int

Address.

OFFSET unsigned int

Indicates which halfword of a 32-bit Thumb instruction the vector catch exception is set on. 0 for the first halfword or if the instruction is 16 bits wide, or 1 for the second halfword.

ArchMsg.Warning.unpredictable_watchpoint_far

TAGS **UNPREDICTABLE** DISPLAY The value for FAR/EDWAR for hitting this watchpoint may be anywhere in the range %{LOWER_BOUND} to %{UPPER_BOUND}. Fields:

LOWER_BOUND unsigned int

the lowest address accessed by the instruction that triggered the watchpoint.

UPPER_BOUND unsigned int

the highest watchpointed address accessed by that instruction.

ArchMsg.Warning.unpredictable_watchpoint_far_isscmp

DISPLAY The value for FAR/EDWAR for hitting this watchpoint may be anywhere in the range %{LOWER_BOUND} to %{UPPER_BOUND}. Fields:

LOWER_BOUND unsigned int

the lowest address accessed by the instruction that triggered the watchpoint.

UPPER_BOUND unsigned int

the highest watchpointed address accessed by that instruction.

ArchMsg.Warning.warning_AdvSIMDExpandImmUnexpectedZero

TAGS UNPREDICTABLE DISPLAY AdvSIMDExpandImm may treat this immediate value as UNPREDICTABLE.

ArchMsg.Warning.warning_ConditionalSMC

TAGS UNPREDICTABLE DISPLAY SMC instruction has UNPREDICTABLE effects when conditional and when combined with traps.

ArchMsg.Warning.warning_TLBID_out_of_range

TLBID out of range in TLBI according to the configuration of NIS, NVIS, NOS and NVOS. Fields:

NIS unsigned int

NIS.

NOS unsigned int

NOS.

NVIS unsigned int

NVIS.

NVOS unsigned int

NVOS.

TLBID unsigned int

The TLBID value used in the TLBI instruction.

ArchMsg.Warning.warning_TTL_or_TG_mismatches_current_translation_granule

TTL or TG mismatches current translation granule in TLBI VA/RVA/IPA/RIPA. Fields:

CURRENT_TG unsigned int

Current TG as indicated by TCR_ELx or VTCR_EL2.

TLBI_TG unsigned int

TG (or TTL[3:2]) used in TLBI instruction.

ArchMsg.Warning.warning_access_crosses_page_boundary

TAGS UNPREDICTABLE DISPLAY Access crosses page boundary.

ArchMsg.Warning.warning_access_crosses_page_boundary_has_fault

TAGS UNPREDICTABLE DISPLAY Access that crosses page boundary has fault(s).

ArchMsg.Warning.warning_access_crosses_page_boundary_has_fault_on_both_pages

TAGS **UNPREDICTABLE** DISPLAY Access that crosses page boundary has a fault on each page.

Fields:

ADDR1 unsigned int

FAR address on first page.

ADDR2 unsigned int

FAR address on second page.

FSTATUS1 unsigned int

Fault status on first page.

FSTATUS2 unsigned int

Fault status on second page.

ArchMsg.Warning.warning_access_crosses_page_boundary_spanning_different_memory

TAGS **UNPREDICTABLE** DISPLAY Access at address %{ADDR} crosses page from MEM_TYPE %{MEMTYPE_PAGE1} to %{MEMTYPE_PAGE2}. Fields:

ADDR unsigned int

address of access crossing page.

MEMTYPE_PAGE1 signed int

First Page's memory type (0-Normal, 1-Non-Normal).

MEMTYPE_PAGE2 signed int

Second Page's memory type (0-Normal, 1-Non-Normal).

ArchMsg.Warning.warning_access_wraps_around_memory

TAGS **UNPREDICTABLE** DISPLAY Access wraps around memory in %{WIDTH:d}bit mode.

Fields:

WIDTH unsigned int

Address width.

ArchMsg.Warning.warning_bcr_linking_status

TAGS **UNPREDICTABLE** DISPLAY DBGBCR.{BT2,BT} combination gives unpredictable behaviour.

ArchMsg.Warning.warning_bcr_mask_reserved

TAGS **UNPREDICTABLE** DISPLAY DBGBCR_ELn.MASK is set to a reserved value.

ArchMsg.Warning.warning_bvr_masked_bit_not_zero

TAGS **UNPREDICTABLE** DISPLAY DBGBVRn_EL1.VA contains a bit masked by DBGBCRn_EL1.MASK that is not zero.

ArchMsg.Warning.warning_ccsidr_unimplemented_level

TAGS **UNPREDICTABLE** DISPLAY CCSIDR read while CSSELR %{CSSELR:x} points to an unimplemented cache level. Fields:

CSSELR unsigned int

current effective value of CSSELR.

ArchMsg.Warning.warning_change_to_ns_when_tge_set

TAGS **UNPREDICTABLE** DISPLAY Attempting to change NS to 1 when HCR.TGE = 1.

ArchMsg.Warning.warning_contiguous_bit_check_abort

TAGS **UNPREDICTABLE** DISPLAY An abort occurred whilst attempting to check TLB entry is contiguous at %{ENTRY_ADDR}. Fields:

ENTRY_ADDR unsigned int

address of conflicting TLB entry.

ArchMsg.Warning.warning_contiguous_bit_error

TAGS **UNPREDICTABLE** DISPLAY TLB entry at %{ENTRY_ADDR} was not contiguous with entry at %{CONTIG_ADDR} Contents are %{ENTRY_DATA} but expected %{CONTIG_DATA}. Fields:

CONTIG_ADDR unsigned int

address of the TLB first read, with which this entry is expected to be contiguous.

CONTIG_DATA unsigned int

expected contents based on the entry at CONTIG_ADDR.

ENTRY_ADDR unsigned int

address of conflicting TLB entry.

ENTRY_DATA unsigned int

contents of conflicting TLB entry.

ArchMsg.Warning.warning_contiguous_bit_error_D128

TAGS **UNPREDICTABLE** DISPLAY TLB entry at %{ENTRY_ADDR} was not contiguous with entry at %{CONTIG_ADDR} Contents are %{ENTRY_DATA_HI} %{ENTRY_DATA_LO} but expected %{CONTIG_DATA}. Fields:

CONTIG_ADDR unsigned int

address of the TLB first read, with which this entry is expected to be contiguous.

CONTIG_DATA unsigned int

expected contents based on the entry at CONTIG_ADDR.

ENTRY_ADDR unsigned int

address of conflicting TLB entry.

ENTRY_DATA_HI unsigned int

Higher bit contents of conflicting TLB entry.

ENTRY_DATA_LO unsigned int

Lower bit contents of conflicting TLB entry.

ArchMsg.Warning.warning_cp10_cp11_mismatch

TAGS **UNPREDICTABLE** DISPLAY Unpredictable configuration of CP10 and CP11 controls at EL %{EL:d}. Fields:

EL unsigned int

EL owning the control.

ArchMsg.Warning.warning_cp10_cp11_reserved_value

TAGS **UNPREDICTABLE** DISPLAY Unpredictable configuration of CPACR CP10 ({CP10}) or CP11 ({CP11}). Fields:

CP10 unsigned int

CP10 access.

CP11 unsigned int

CP11 access.

ArchMsg.Warning.warning_csselr_level_out_of_range

TAGS **UNPREDICTABLE** DISPLAY CSSELR.Level out of range, written value {LEVEL} with only {IMPLEMENTED:d} levels implemented. Fields:

IMPLEMENTED unsigned int

number of cache levels implemented.

LEVEL unsigned int

written value of CSSELR.Level.

TYPE enum

The type of cache selected by CSSELR.InD and TnD.

ArchMsg.Warning.warning_debug_flow_control_bits_not_obeyed

TAGS **UNPREDICTABLE** DISPLAY Flow control bits for {REGISTER} were not obeyed, causing {ERROR}. Fields:

ERROR enum

type of error resulting.

REGISTER enum

register under consideration.

ArchMsg.Warning.warning_debug_register_access_during_reset

TAGS **UNPREDICTABLE** DISPLAY Unpredictable {IS_WRITE:(read|write)} access to debug register offset {OFFSET} during reset. Fields:

IS_WRITE unsigned int

Write Not Read.

OFFSET unsigned int

Register Offset.

ArchMsg.Warning.warning_decode_cps_inconsistent_fields

TAGS **UNPREDICTABLE** DISPLAY The mode setting fields in the CPS instruction are inconsistent.

ArchMsg.Warning.warning_decode_invalid_state

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable in current exception-level/security state.

ArchMsg.Warning.warning_default_cacheable_mmu_on

TAGS **UNPREDICTABLE** DISPLAY Default cacheable is enabled (HCR.DC = 1) while MMU is enabled.

ArchMsg.Warning.warning_default_cacheable_vmmu_off

TAGS **UNPREDICTABLE** DISPLAY Default cacheable is enabled (HCR.DC = 1) while HCR.VM = 0.

ArchMsg.Warning.warning_deprecated_wvr_bit_2_set

TAGS **UNPREDICTABLE** DISPLAY DBGWVR%{N:d}_EL1=%{WVR:h} has bit 2 set, deprecated in ARMv8. Fields:

N unsigned int

index.

WVR unsigned int

DBGWVR.

ArchMsg.Warning.warning_eret_while_software_step_active_pending

Missing ISB between setting MDSCR_EL1.SS and ERET TAGS **UNPREDICTABLE**.

ArchMsg.Warning.warning_exclusive_to_non_normal

TAGS **UNPREDICTABLE** DISPLAY Exclusive Access to Non-Normal Memory %{ADDR}. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_exclusive_to_non_writeback

TAGS **UNPREDICTABLE** DISPLAY Exclusive Access to Non-Writeback-Cacheable Memory %{ADDR}. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_execute_from_device_memory

TAGS **UNPREDICTABLE** DISPLAY An attempt was made to execute from Device memory.

ArchMsg.Warning.warning_illegal_cpsr_mode

TAGS **UNPREDICTABLE** DISPLAY Writing an illegal or unimplement mode (%{NEW_MODE:h}) to CPSR was unpredictable. Fields:

NEW_MODE unsigned int

New value of CPSR.M.

ArchMsg.Warning.warning_illegal_srs_mode

TAGS **UNPREDICTABLE** DISPLAY Illegal or **UNPREDICTABLE** mode (%{MODE:h}) used for SRS instruction. Fields:

MODE unsigned int

mode for Banked SP.

ArchMsg.Warning.warning_implementation_defined_read_debug_register_in_SCS

Read Access to Debug registers from the processor is IMPLEMENTATION DEFINED TAGS IMP_DEF. Fields:

PPB_OFFSET unsigned int
debug register offset.

ArchMsg.Warning.warning_implementation_defined_sequential_security_transitions_supported

The behavior of sequential instruction fetches that cross from non-secure to secure memory and contain an SG instruction is CONSTRAINED UNPREDICTABLE TAGS UNPREDICTABLE.

ArchMsg.Warning.warning_implementation_defined_stack_limit_check_supported

It is IMPLEMENTATION DEFINED whether stack pointer limit checking is performed for this instruction TAGS IMP_DEF. Fields:

ADDRESS unsigned int
Address of instruction.

OPCODE unsigned int
instruction opcode.

ArchMsg.Warning.warning_implementation_defined_taggednonsharedmemorytype

TAGS IMP_DEF DISPLAY Memory attribute has Tagged, non shared memory type; it is IMPLEMENTATION DEFINED whether the memory location is treated as Tagged or non-Tagged.

ArchMsg.Warning.warning_implementation_defined_write_debug_register_in_SCS

Write Access to Debug registers from the processor is IMPLEMENTATION DEFINED TAGS IMP_DEF. Fields:

DATA unsigned int
data attempted to be written.

PPB_OFFSET unsigned int
debug register offset.

ArchMsg.Warning.warning_invalid_tcr_granule

TAGS UNPREDICTABLE DISPLAY TCR.TG%{TG_ID:(0|1)} bits have been set to a granule size %{{REQUEST:d}}K not implemented - using %{{SUBSTITUTE:d}}K. Fields:

REQUEST signed int
page size requested (or 0 for reserved).

SUBSTITUTE signed int
best guess available page size.

TG_ID bool
bits TG0 or TG1.

ArchMsg.Warning.warning_load_multiple_user_registers_from_user_mode

TAGS UNPREDICTABLE DISPLAY An LDM(user registers) instruction executed from user mode.

ArchMsg.Warning.warning_mte_in_mem_attr_not_supported

TAGS UNPREDICTABLE DISPLAY Memory attribute %{ATTR} has Tagged memory type but Armv8.5-MTE is not implemented. Fields:

ATTR unsigned int

Memory attribute.

ArchMsg.Warning.warning_reserved_breakpoint_state_match

TAGS UNPREDICTABLE DISPLAY %{IS_BREAKPOINT:(Watchpoint|Breakpoint)} programmed with a reserved combination of HMC, SSC and %{IS_BREAKPOINT:(PAC|PMC)}. Fields:

IS_BREAKPOINT unsigned int

Is a breakpoint, not a watchpoint.

ArchMsg.Warning.warning_shareability

TAGS UNPREDICTABLE DISPLAY Unpredictable: combination of the 1st and 2nd stages of translation is Normal Inner Non-Cacheable, Outer Non-Cacheable.

ArchMsg.Warning.warning_software_step_set_while_enabled

TAGS UNPREDICTABLE DISPLAY MDSCR_EL1.SS set to 1 while software step debug exceptions are enabled.

ArchMsg.Warning.warning_thumb_instruction_wraps_around_memory

TAGS UNPREDICTABLE DISPLAY Thumb instruction wraps around memory.

ArchMsg.Warning.warning_ttbr_sbz_bits_are_not_zero

TAGS UNPREDICTABLE DISPLAY %{TTBR}[%{MSB:d}:0] should be zero, but are not. Fields:

MSB unsigned int

x-1.

TTBR enum

which TTBR register.

ArchMsg.Warning.warning_unaligned_address_dbgdtrrx_write

TAGS UNPREDICTABLE DISPLAY Unaligned address %{ADDR} in X0/R0 when DBGDTRRX written in memory access mode. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unaligned_address_dbgdtrtx_read

TAGS UNPREDICTABLE DISPLAY Unaligned address %{ADDR} in X0/R0 when DBGDTRTX read in memory access mode. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unaligned_to_device

TAGS UNPREDICTABLE DISPLAY Unaligned Access to Device Memory %{ADDR}. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unaligned_to_strongly_ordered

TAGS **UNPREDICTABLE** DISPLAY Unaligned Access to Strongly Ordered Memory %{ADDR}.

Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unknown_sau_rnr

TAGS **UNPREDICTABLE** DISPLAY SAU_RNR was set to an unsupported value.

ArchMsg.Warning.warning_unpredictable_AIRCR_PRIS_and_BFHFNMINs

TAGS **UNPREDICTABLE** DISPLAY The effect of setting both AIRCR.BFHFNMINs and AIRCR.PRIS to 1 is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_AIRCR_VECTCLRACTIVE_when_not_in_debug

TAGS **UNPREDICTABLE** DISPLAY The effect of writing a 1 to AIRCR.VECTCLRACTIVE if the processor is not halted in Debug state is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_AIRCR_VECTRESET_and_SYSRESETREQ

TAGS **UNPREDICTABLE** DISPLAY When the processor is halted in Debug state, if a write to the register writes a 1 to both VECTRESET and SYSRESETREQ, the behavior is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_AIRCR_VECTRESET_when_not_in_debug

TAGS **UNPREDICTABLE** DISPLAY The effect of writing a 1 to AIRCR.VECTRESET if the processor is not halted in Debug state is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_AIRCR_incorrect_VKEY

TAGS **UNPREDICTABLE** DISPLAY The value 0x05FA must be written to AIRCR.VKEY, otherwise the register write is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_EXC_RETURN_Reserved_Bit

EXC_RETURN[23:7] are reserved with the special condition that all bits should be written as one. Values other than all 1s are **UNPREDICTABLE**. TAGS **UNPREDICTABLE**. Fields:

EXC_RETURN unsigned int

Special PC value.

SBO_MASK unsigned int

Which bits Should Be One to not UNPRED.

ArchMsg.Warning.warning_unpredictable_PMEVTYPEPERn_ELO_VS_reserved_value

TAGS **UNPREDICTABLE** DISPLAY Unpredictable reserved value of PMEVTYPEPER<n>_ELO.VS.

ArchMsg.Warning.warning_unpredictable_change_to_DEMCR_MON_STEP_at_insufficient_priority

TAGS **UNPREDICTABLE** DISPLAY The effect of changing DEMCR.MON_STEP at an execution priority that is lower than the priority of the DebugMonitor exception is **CONSTRAINED UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_change_to_priority_of_active_exception

TAGS UNPREDICTABLE DISPLAY Changing the priority of an active exception is UNPREDICTABLE.

ArchMsg.Warning.warning_unpredictable_clear_lspact

TAGS UNPREDICTABLE DISPLAY UNPREDICTABLE state on context switch. Fields:

DRVEXC_TAKEN unsigned int

Derived exception taken, not pended.

REACHED_LAST_STORE unsigned int

Only the last faulted.

ArchMsg.Warning.warning_unpredictable_dbg_exit_unaligned_dlr

TAGS UNPREDICTABLE DISPLAY Debug exit to AArch32 with DLR[0] set to 1. PC[0] can be set to 0 or to DLR[0].

ArchMsg.Warning.warning_unpredictable_exception_catch

TAGS UNPREDICTABLE DISPLAY Generation of an exception catch event is unpredictable.

ArchMsg.Warning.warning_unpredictable_exception_return_inconsistent_state

TAGS UNPREDICTABLE DISPLAY state on exception return is unpredictable.

ArchMsg.Warning.warning_unpredictable_exception_return_instruction

TAGS UNPREDICTABLE DISPLAY exception return instruction is unpredictable.

ArchMsg.Warning.warning_unpredictable_in_debug_state

TAGS UNPREDICTABLE DISPLAY Instruction is UNPREDICTABLE when executed in debug state.

ArchMsg.Warning.warning_unpredictable_mair_encoding

TAGS UNPREDICTABLE DISPLAY Memory attribute %{ATTR} has invalid MAIR encoding; this may lead to unpredictable results. Fields:

ATTR unsigned int

Memory attribute.

ArchMsg.Warning.warning_unpredictable_pmu_counter_access

TAGS UNPREDICTABLE DISPLAY Access to PMU counters from non-secure EL0 or EL1 is UNPREDICTABLE with MDCR_EL2.HPMN set to 0.

ArchMsg.Warning.warning_unpredictable_prioritization_breakpoint_match_vector_catch

TAGS UNPREDICTABLE DISPLAY Two events with the same priority occurred at the same instruction: Address Matching Vector Catch debug event, and Breakpoint debug event. It is constrained unpredictable which is taken.

ArchMsg.Warning.warning_unpredictable_stack_selection

TAGS UNPREDICTABLE DISPLAY UNPREDICTABLE state on context switch.

ArchMsg.Warning.warning_unpredictable_tsize_out_of_range

TAGS UNPREDICTABLE DISPLAY %{TCR}.T%{TTBR:d}SZ (%{TSIZE:d} bits) is out of range. Must be between 25 and %{TSIZE_MAX:d}. Fields:

TCR enum

Which TCR.

TSize signed int

Value of TSize.

TSize_MAX signed int

The maximum allowed value.

TTBR unsigned int

Which TTBR region.

ArchMsg.Warning.warning_unpredictable_unaligned_pc_as_base_register

TAGS **UNPREDICTABLE** DISPLAY The PC value must be word-aligned, otherwise the behavior of the instruction is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_unaligned_pop_stack

TAGS **UNPREDICTABLE** DISPLAY **CONSTRAINED UNPREDICTABLE** unaligned stack frame `%{FRAME_ADDR:x}` in exception-return. Fields:

FRAME_ADDR unsigned int

unaligned stack frame address.

ArchMsg.Warning.warning_unpredictable_unaligned_to_non_writeback

TAGS **UNPREDICTABLE** DISPLAY Unaligned Access to Non-Writeback-Cacheable Memory.

ArchMsg.Warning.warning_unpredictable_vmsa_memattrib

TAGS **UNPREDICTABLE** DISPLAY Unpredictable vmsa memory attribute with texcb value programmed to `%{TEXCB}` when tex remap is `%{TEX_REMAP}`. Fields:

TEXCB unsigned int

texcb value.

TEX_REMAP bool

use tex remap.

ArchMsg.Warning.warning_unpredictable_vtcr_t0sz_sl0

TAGS **UNPREDICTABLE** DISPLAY The combination of VTCR.TOSZ=`%{TOSZ:d}` and VTCR.SL0=`%{SL0:d}` is unpredictable. Fields:

SL0 signed int

VTCR.SL0, starting level for stage 2 translation table walks.

TOSZ signed int

VTCR.TOSZ, required input address range.

ArchMsg.Warning.warning_unpredictable_write_DHCSR

TAGS **UNPREDICTABLE** DISPLAY **UNPREDICTABLE** set of bit-changes in DHCSR. Fields:

NEW unsigned int

data attempted to be written.

OLD unsigned int

previous value.

ArchMsg.Warning.warning_unpredictable_write_SHCSR

TAGS UNPREDICTABLE DISPLAY UNPREDICTABLE setting a pending bit to 1 for an exception with priority \geq execution priority.

ArchMsg.Warning.warning_unsupported_access_to_memory_mapped_register

TAGS UNPREDICTABLE DISPLAY Access to memory mapped register is unsupported.

ArchMsg.Warning.warning_user_jmcr_access

TAGS UNPREDICTABLE DISPLAY User %{WRITE:(write|read)} access to JMCR is unpredictable. Fields:

WRITE unsigned int
access is write.

ArchMsg.Warning.warning_wcr_mask_and_bas

TAGS UNPREDICTABLE DISPLAY DBGWCR_EL1.MASK is non-zero and BAS is not 0xff.

ArchMsg.Warning.warning_wcr_mask_reserved

TAGS UNPREDICTABLE DISPLAY DBGWCR_ELn.MASK is set to a reserved value.

ArchMsg.Warning.warning_wcr_non_configuous_bas

TAGS UNPREDICTABLE DISPLAY DBGWCR_EL1.BAS has non-contiguous set of ones.

ArchMsg.Warning.warning_wvr_masked_bit_not_zero

TAGS UNPREDICTABLE DISPLAY DBGWVRn_EL1.VA contains a bit masked by DBGWCRn_EL1.MASK that is not zero.

ArchMsg.Why.why_branch_target_exception

A branch target exception has occurred. DISPLAY branch target exception: target pc=%{PC}, target opcode=%{OPCODE}. PRIMARY KEY PC. Fields:

BTYPE unsigned int
pstate.BTYPE at time branch taken.

OPCODE unsigned int
opcode at branch target.

PC unsigned int
Virtual Address of branch target.

ArchMsg.Why.why_illegal_state

An illegal mode change has occurred. DISPLAY illegal mode change: %{MESSAGE}. Fields:

MESSAGE string
The message.

BRANCH_MISPREDICT

Simulating branch mispredict. Fields:

PC unsigned int
Origin address (or 0 if unavailable).

BRA_DIR

Direct branches, to immediate address. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISET enum

The instruction set of the branch instruction.

IS_COND bool

Indicates if this is a conditional branch.

IS_HINTED bool

Whether the conditional instruction is hinted.

IS_ISB bool

Whether the instruction is ISB.

IS_LINK bool

Indicates if this is a branch with link.

PC unsigned int

The address of the branch instruction.

TARGET_ISET enum

The instruction set after the branch.

TARGET_PC unsigned int

The address the instruction branches to.

BRA_INDIR

Indirect branches, perhaps to a register. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISET enum

The instruction set of the branch instruction.

IS_COND bool

Indicates if this is a conditional branch.

IS_HINTED bool

Whether the conditional instruction is hinted.

IS_ISB bool

Whether the instruction is ISB.

IS_LINK bool

Indicates if this is a branch with link.

PC unsigned int

The address of the branch instruction.

TARGET_ISET enum

The instruction set after the branch.

TARGET_PC unsigned int

The address the instruction branches to.

BRA_INDIR_CCFAIL

Indirect branches fail, perhaps to a register.

BRA_RET_CCFAIL

Branch return fail, perhaps to a register.

CACHE_MAINTENANCE_OP

Cache Maintenance Operation. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

DATA unsigned int

Specified MVA or Set/way.

FUNCTION enum

Clean / Invalidate.

SCOPE enum

Affected region.

SIDE enum

Inst / Data.

CCFAIL

Conditional instruction condition check fail. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CCFAIL_UNC

Unconditional instruction condition check fail. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CCPASS

Conditional instruction condition check pass. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CCPASS_UNC

Unconditional instruction condition check pass. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CODE_CACHE_MAINT

Code cache maintenance. Fields:

END_INCL unsigned int

Inclusive end address for VA or PA requests.

NS bool

Non-secure world for PA.

START unsigned int

Start address for VA or PA requests.

TYPE enum

Request type.

COMPILE_BLOCK_END

Last instruction of basic block translated. Fields:

VADDR unsigned int

Address of next instruction after this basic block.

COMPILE_BLOCK_START

First instruction of basic block translated. Fields:

VADDR **unsigned int**

Address of first instruction in basic block.

COMPILE_INST

ARM instruction compiled. Fields:

DISASS **string**

Disassembly of the instruction.

ISET **enum**

The instruction set of this instruction.

ITSTATE **unsigned int**

The ITSTATE current for the instruction.

OPCODE **unsigned int**

The opcode of the instruction.

PC **unsigned int**

The address of the instruction.

SIZE **unsigned int**

The size of the instruction in bytes.

CONTEXTIDR

Context ID Register write. Fields:

CORE_NUM **unsigned int**

Core number in a multi processor.

NS **bool**

Secure or nonsecure banked register is accessed.

UNDEF **bool**

The register accessed is undefined.

VALUE **unsigned int**

The new value written.

CONTEXT_SYNC

Called for every context synchronization event. Fields:

ARCH_CSE **bool**

Architectural Context Synchronization Event.

PC **unsigned int**

The address of the synchronization instruction.

SYSREG_SYNC **bool**

System register synchronization event.

CORE_ENDIAN

Core BE8 Big-Endian state changed. Fields:

BE8 bool

Core BE8 Big-Endian state.

CORE_INFO

Static processor attributes. Only triggered by a call to DumpState(). Fields:

ARCH_PROFILE enum

The architecture profile of the core.

CLUSTER_ID unsigned int

The cluster ID of this processor.

CORE_NUM unsigned int

The number of this core in an MP processor.

DETAILED_CORE_FEATURES enum

Detailed core features as a bitfield. Enum values are bitmasks to extract the info from the field.

FPU_VERSION enum

The VFP version implemented by the core.

MEM_ARCH enum

The memory architecture of the core.

MIDR_EL1 unsigned int

MIDR_EL1 value.

NUM_CORES unsigned int

The number of cores in this MP processor.

QUANTUM_SIZE unsigned int

The default quantum size of the core (in ticks).

SECURITY_FEATURES bool

Does the core have security features?.

CORE_LOADS

Processor load accesses. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ACQREL enum

Is this an acquire/release.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DATA unsigned int

The data read or written.

ELEMENT_SIZE unsigned int

Width of each element.

ENDIANNESS enum

Byte ordering of this access.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

LOCK enum

Normal, exclusive or locked access.

MECID unsigned int

Memory Encryption Context ID, if applicable.

MECID2 unsigned int

Memory Encryption Context ID of the second page, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PADDR_TAG unsigned int

If tag checked, the physical address tag for the access.

PAGE_SIZE unsigned int

Page size as \log_2 (size in bytes).

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK, 2=Exclusive Failed, 3=Unsupported.

SIZE unsigned int

Width of the access in bytes. Only required if DATA is not traced.

TAG_CHECKED bool

Is this access tag checked.

TAG_CHECKED2 bool

Is the second page access tag checked.

TRANS bool

Is this a translated access.

VADDR unsigned int

The virtual address of the access.

CORE_REGS

Changes of the core registers R0 to R14. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number, 0 to 14.

MODE enum

Bank of the register accessed.

OLD_VALUE unsigned int

The old value overwritten.

PHYS_ID enum

The physical register accessed.

VALUE unsigned int

The new value written to the register.

CORE_REGS64

Changes of the core registers X0..X30, SP_ELn. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ID enum

The register number, 0-30 for X0-X30, >=32 for SP_ELn.

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

CORE_REGS64_READ

General purpose AArch64 register has been read X0..X30, SP_ELn. Fields:

ID enum

The register number, 0-30 for X0-X30, >=32 for SP_ELn.

VALUE unsigned int

The value read from the register.

CORE_STALLED_TRANSITION

Detect when a core is prevented from progressing PChannel states. Fields:

CURRENT_STATE enum

Current power mode of core.

REASON string

The reason a transition was stalled, and what must be done to unblock it.

TARGET_STATE enum

Desired power mode of core after transition.

CORE_STORES

Processor store accesses. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ACQREL enum

Is this an acquire/release.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DATA unsigned int

The data read or written.

ELEMENT_SIZE unsigned int

Width of each element.

ENDIANNESS enum

Byte ordering of this access.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

LOCK enum

Normal, exclusive or locked access.

MECID unsigned int

Memory Encryption Context ID, if applicable.

MECID2 unsigned int

Memory Encryption Context ID of the second page, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PADDR_TAG unsigned int

If tag checked, the physical address tag for the access.

PAGE_SIZE unsigned int

Page size as $\log_2(\text{size in bytes})$.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK, 2=Exclusive Failed, 3=Unsupported.

SIZE unsigned int

Width of the access in bytes. Only required if DATA is not traced.

TAG_CHECKED bool

Is this access tag checked.

TAG_CHECKED2 bool

Is the second page access tag checked.

TRANS bool

Is this a translated access.

VADDR unsigned int

The virtual address of the access.

CP14_READ

System Coprocessor register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP14 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc1 unsigned int

opcode 1.

opc2 unsigned int

opcode 2.

CP14_WRITE

System Coprocessor register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP14 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc1 unsigned int

opcode 1.

opc2 unsigned int

opcode 2.

CP15_READ

System Coprocessor register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP15 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc1 unsigned int

opcode 1.

opc2 unsigned int

opcode 2.

CP15_READ64

System Coprocessor register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP15 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc unsigned int

opcode 1.

CP15_WRITE

System Control Coprocessor register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP15 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc1 unsigned int

opcode 1.

opc2 unsigned int

opcode 2.

CP15_WRITE64

System Coprocessor register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP15 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc unsigned int

opcode 1.

CPSR

CPSR change. (In AArch64 PSTATE represented in SPSR format). Fields:

CORE_NUM unsigned int

Core number in a multi processor.

OLD_VALUE unsigned int

The old CPSR value.

UNKNOWN unsigned int

Bits within the register that have unknown value.

VALUE unsigned int

The new CPSR value.

CRYPTO_SPEC

Every crypto instruction speculatively executed.

CorePowerStateChange

Core power-up state has changed. DISPLAY Core powered %{PU:(up|down)}. Fields:

PU bool

Whether the core powered up.

DATA_CACHE_SET_ALLOC_TAGS

Set allocation tags from DC GVA instruction. Fields:

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

NSDESC unsigned int

The security state of the access.

PADDR unsigned int

The physical address of the access.

PAS enum

The physical address space of the page.

RESPONSE enum

0=Aborted, 1=OK.

SIZE unsigned int

The size of memory in bytes.

TAG unsigned int

The value of the tag being set.

TAGGING_TYPE enum

Type of Memory tagging (Physical or Virtual).

TAG_VADDR unsigned int

The virtual address of the Virtual Tag if Virtual tagging is used; otherwise 0.

VADDR unsigned int

Virtual address used for setting allocation tags.

DATA_CACHE_ZERO

Zero and invalidate cache line from DC ZVA instruction. Fields:

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

NSDESC unsigned int

The security state of the access.

PADDR unsigned int

The physical address of the access.

PADDR_TAG unsigned int

If tag checked, the physical address tag for the access.

PAS enum

The physical address space of the page.

RESPONSE enum

0=Aborted, 1=OK.

SIZE unsigned int

The size of zero'd memory in bytes.

TAG_CHECKED bool

Is this access tag checked.

VADDR unsigned int

Virtual address of the zero'd cache line.

DEBUG_EVENT

Hardware debug support event. Fields:

EVENT enum

Description of event.

VALUE unsigned int

data value.

DEBUG_TARGET_ISET

Hardware debug target ISA. Fields:

TARGET_ISET enum

The target instruction set of the debug state.

DMI_ALLOCATE

DMI allocated. Fields:

ACCESS_DIRECTION enum

READ / WRITE.

PADDR unsigned int

The physical address of the DMI.

PTR unsigned int

DMI range pointer.

SIDE enum

DSIDE / ISIDE.

SIZE unsigned int

The size of DMI.

VADDR unsigned int

The virtual address of the DMI..

DMI_HIT

DMI hit, access used DMI (only fire for small DMI regions i.e. less than a full page). Fields:

ACCESS_DIRECTION enum

READ / WRITE.

PADDR unsigned int

The physical address of the DMI.

SIDE enum

DSIDE / ISIDE.

SIZE unsigned int

The size of the DMI.

VADDR unsigned int

The virtual address of the DMI..

DMI_REVOKE

DMI revoked. Fields:

ACCESS_DIRECTION enum

READ / WRITE.

PADDR unsigned int

The physical address of the DMI.

SIDE enum

DSIDE / ISIDE.

SIZE unsigned int

The size of DMI.

VADDR unsigned int

The virtual address of the DMI..

END_COMPILE

Compilation end. Fields:

END_OF_PAGE_COUNT unsigned int

Number of basic blocks exited due to page boundary since START_COMPILE.

FETCHFAIL_COUNT unsigned int

Number of basic blocks exited due to ifetch failure START_COMPILE.

INST_COUNT unsigned int

Number of instructions compiled since START_COMPILE.

NONSEQ_COUNT unsigned int

Number of basic blocks exited due to non-sequential instructions since START_COMPILE.

PAGE_STRADDLE_COUNT unsigned int

Number of basic blocks exited due to unaligned instructions crossing page since START_COMPILE.

ESB

Error Synchronization Barrier hit. Fields:

ASYNC bool

ESB is async.

IMPLICIT bool

ESB is implicit.

EXCEPTION

Exception taken. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ESR unsigned int

The value assigned to ESR when exception is taken.

ESR_EC unsigned int

The Value assigned to ESR.EC when exception is taken.

IS_PHYSICAL bool

Physical or Virtual exception.

IS_TRAP bool

Exception coming from a trap.

LR unsigned int

The value assigned to the link register.

PC unsigned int

The location where the exception occurred.

PREFERRED_RETURN unsigned int

The preferred return address for the exception.

TAKEN_LOCALLY bool

Whether exception is taken locally(same exception level).

TARGET_ISET enum

The instruction set of the exception handler code.

TARGET_PC unsigned int

The address the exception branches to.

VECTOR enum

The exception vector.

VECTOR_OFFSET unsigned int

The offset of an exception vector from the base address.

EXCEPTION_END

Every exception completed.

EXCEPTION_INFO

Trace human-readable reason of the current exception for data/instruction aborts, floating point exceptions and unknown exceptions. Fields:

DESCRIPTION string

Reason for current exception.

ESR_EC unsigned int

The value of ESR.EC when exception is taken.

EXCEPTION_RAISE

Every exception raised.

EXCEPTION_RETURN

Branches on leaving exception. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISET enum

The instruction set of the branch instruction.

IS_COND bool

Indicates if this is a conditional branch.

IS_HINTED bool

Whether the conditional instruction is hinted.

IS_ISB bool

Whether the instruction is ISB.

IS_LINK bool

Indicates if this is a branch with link.

PC unsigned int

The address of the branch instruction.

TARGET_ISET enum

The instruction set after the branch.

TARGET_PC unsigned int

The address the instruction branches to.

EXCEPTION_RETURN_PREBRANCH

Exception return event that must occur before PC and processor state have updated.

EXCEPTION_START

Every exception started.

FIQ_TAKEN

FIQ taken exception.

FP_EXCEPTIONS_SIGNALLED

Trace which FP exceptions were signalled as a part of the FP operations performed by an instruction. Fields:

FP_EXCEPTIONS_SIGNALLED_MASK unsigned int

Indicates which FP exceptions were signalled by an instruction. This uses the same encoding as the cumulative floating-point exception bitfields in FPSR.

FP_STATE

Floating point state. Fields:

FPSR unsigned int

FPSR register value.

MASK unsigned int

The mask indicating updated cumulative bits.

FREQ_CHANGED

Core frequency changed. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The instruction count of this core.

NEW_FREQ signed int

The new frequency of this core (expressed in Hz).

OLD_FREQ signed int

The old frequency of this core (expressed in Hz).

HLT

HLT instruction occurred. Fields:

IMM16 unsigned int

The 16-bit immediate value encoded in HLT instruction.

INST

Every instruction executed. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DEBUG_STATE bool

The instruction is executed in debug state.

DISASS string

Disassembly of instruction.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISSET enum

The current instruction set.

ITSTATE unsigned int

The current ITSTATE.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

MODE enum

The mode the core is in.

NS unsigned int

The core's non-secure bit.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

OPCODE unsigned int

The opcode of the instruction.

PADDR unsigned int

The physical address of the instruction.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the instruction.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the page.

PC unsigned int

The address of the instruction.

SECURITY_STATE enum

The core's security state.

SIZE unsigned int

The size of the instruction in bytes.

INST_END

Every instruction completed.

INST_START

Every instruction started. Fields:

DEBUG_STATE bool

The instruction is executed in debug state.

ISSET enum

The current instruction set.

MODE enum

The mode the core is in.

NS enum

The current Secure State.

PC unsigned int

The address of the conditional instruction.

SECURITY_STATE enum

The current Security State.

INST_STRADDLE

Instruction straddles boundary. Fields:

DEBUG_STATE bool

The instruction is executed in debug state.

ISSET enum

The current instruction set.

MODE enum

The mode the core is in.

NS enum

The current Secure State.

PC unsigned int

The address of the conditional instruction.

SECURITY_STATE enum

The current Security State.

IRQ_TAKEN

IRQ taken exception.

ISB

ISB instruction occurred. Fields:

PC unsigned int

The address of the ISB instruction.

LOAD_MULTIPLE_REGS_LIST

List of destination registers (not source) of a multiple registers load instruction. Fields:

REG string

Register to add into the list.

LOCAL_MONITOR

Local monitor activity. Fields:

PADDR unsigned int

Local Monitor Address.

State enum

State of the monitor (Open/Exclusive).

MAPPING_CREATE

A memory mapping has been created. This is an implementation detail of the model and so may change without notice. Fields:

ASID unsigned int

The ASID used during translation, or 0 if no ASID was used.

IS_VALID bool

Whether the mapping is valid or not.

PADDR unsigned int

Base physical address of region, or 0 if not known.

SIDE enum

I-side or D-side.

SIZE unsigned int

Size of the page, or 0 if not known.

VADDR unsigned int

Base virtual address of region.

VMID unsigned int

The VMID used during translation, or 0 if no VMID was used.

MAPPING_DESTROY

A memory mapping has been destroyed. This is an implementation detail of the model and so may change without notice. Fields:

ASID unsigned int

The ASID used during translation.

CAUSE enum

The cause of invalidation.

PADDR unsigned int

Base physical address of page.

SIDE enum

I-side or D-side.

SIZE unsigned int

Size of the page.

VADDR unsigned int

Base virtual address of page.

VMID unsigned int

The VMID used during translation.

MEMMAP_DEBUG_READ

Memory mapped reads to the debug registers. Fields:

ADDR unsigned int

address.

CORE_NUM unsigned int

Core number in a multi processor.

EXT bool

Whether access is from an external device (such as the DAP).

NS enum

Secure state of the access.

REG_NAME string

Name of the debug register accessed.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

MEMMAP_DEBUG_WRITE

Memory mapped writes to the debug registers. Fields:

ADDR **unsigned int**

address.

CORE_NUM **unsigned int**

Core number in a multi processor.

EXT **bool**

Whether access is from an external device (such as the DAP).

NS **enum**

Secure state of the access.

REG_NAME **string**

Name of the debug register accessed.

UNDEF **bool**

The register accessed is undefined.

VALUE **unsigned int**

The new value written.

MEMTAG_LOADS

Allocation tag memory read; when attributes and system configuration allow tag accesses.
Fields:

ACCESS_TYPE **enum**

The type of instruction performing the access.

ATTR **unsigned int**

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 **unsigned int**

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

NSDESC **unsigned int**

The physical address non-secure bit.

NSDESC2 **unsigned int**

The second page physical address non-secure bit.

NUM_GRANULES **unsigned int**

Number of tags to be Loaded/stored.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK.

TAGGING_TYPE enum

Type of Memory tagging (Physical or Virtual).

TAGS unsigned int

The tags read or written. Tags for subsequent granules would be found at subsequent bytes in this field (One byte for each tag).

TAG_VADDR unsigned int

The virtual address of the Virtual Tag if Virtual tagging is used; otherwise 0.

VADDR unsigned int

The virtual address of the access.

MEMTAG_LOAD_INST

Allocation tag load instruction; regardless of tags being actually accessed. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

NUM_GRANULES unsigned int

Number of tags to be Loaded/stored.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK.

TAGGING_TYPE enum

Type of Memory tagging (Physical or Virtual).

TAGS unsigned int

The tags read or written. Tags for subsequent granules would be found at subsequent bytes in this field (One byte for each tag).

TAG_VADDR unsigned int

The virtual address of the Virtual Tag if Virtual tagging is used; otherwise 0.

VADDR unsigned int

The virtual address of the access.

MEMTAG_STORES

Allocation tag memory write; when attributes and system configuration allow tag accesses.

Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

NUM_GRANULES unsigned int

Number of tags to be Loaded/stored.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK.

TAGGING_TYPE enum

Type of Memory tagging (Physical or Virtual).

TAGS unsigned int

The tags read or written. Tags for subsequent granules would be found at subsequent bytes in this field (One byte for each tag).

TAG_VADDR unsigned int

The virtual address of the Virtual Tag if Virtual tagging is used; otherwise 0.

VADDR unsigned int

The virtual address of the access.

MEMTAG_STORE_INST

Allocation tag store instruction; regardless of tags being actually accessed. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

NUM_GRANULES unsigned int

Number of tags to be Loaded/stored.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK.

TAGGING_TYPE enum

Type of Memory tagging (Physical or Virtual).

TAGS unsigned int

The tags read or written. Tags for subsequent granules would be found at subsequent bytes in this field (One byte for each tag).

TAG_VADDR unsigned int

The virtual address of the Virtual Tag if Virtual tagging is used; otherwise 0.

VADDR unsigned int

The virtual address of the access.

MMU_D128_TTB_READ

This event is triggered by reads caused by a D128 translation table walk. Fields:

ABORTED bool

Was the walk successful.

DATA_HIGH unsigned int

The data upper bits read.

DATA_LOW unsigned int

The data lower bits read.

IPA unsigned int

For stage 1, the IPA of the read.

LEVEL unsigned int

Translation table level.

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

OUTADDR unsigned int

The output physical address.

PADDR unsigned int

The physical address of the read.

PASmem enum

Physical Address Space of the accessed memory.

PASreq enum

Physical Address Space of the requested lookup.

REGIME_EL enum

Entry matches in this translation regime.

REQUESTER enum

The requester of this table walk.

SIDE enum

Inst / Data.

STAGE unsigned int

Translation stage.

MMU_D128_TTB_WRITE

This event is triggered by writes caused by a D128 translation table walk. Fields:

ABORTED bool

Was the walk successful.

DATA_HIGH unsigned int

The data upper bits written.

DATA_LOW unsigned int

The data lower bits written.

IPA unsigned int

For stage 1, the IPA of the write.

IS_HACDBS_ACCESS bool

If the access was due to HACDBS_CLEAN.

LEVEL unsigned int

Translation table level.

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

OUTADDR unsigned int

The output physical address.

PADDR unsigned int

The physical address of the write.

PASmem enum

Physical Address Space of the accessed memory.

PASreq enum

Physical Address Space of the requested lookup.

REGIME_EL enum

Entry matches in this translation regime.

SIDE enum

Inst / Data.

STAGE unsigned int

Translation stage.

MMU_TRANS

Address translation information. Fields:

ASID unsigned int

Address space identifier.

CORE_NUM unsigned int

Core number in a multi processor.

Hyp bool

Entry matches in Hyp state only.

INNERCACHE_RA bool

Is the inner cache allocate on read.

INNERCACHE_TRANSIENT bool

Is the inner write-through transient.

INNERCACHE_TYPE enum

Inner Caching scheme (NC/MB/WA).

INNERCACHE_WA bool

Is the inner cache allocate on write.

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MEMTYPE enum

Memory type.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC enum

Is secure side supposed to access secure or nonsecure memory.

OUTERCACHE_RA bool

Is the outer cache allocate on read.

OUTERCACHE_TRANSIENT bool

Is the outer write-through transient.

OUTERCACHE_TYPE enum

Outer Caching scheme (NC/MB/WA).

OUTERCACHE_WA bool

Is the outer cache allocate on write.

PADDR unsigned int

Physical address of the access.

PAGESIZE unsigned int

Page size as $\log_2(\text{size})$.

PAS enum

PAS supposed to be accessed is secure or nonsecure memory or Root/Realm if RME is implemented.

SH enum

Shareability.

SIDE enum

Inst / Data.

STAGE1_PERM unsigned int

Stage 1 permissions, [5:3] Privileged access (XWR) [2:0] User access (XWR).

STAGE2_PERM unsigned int

Stage 2 permission mask [5:3] Privileged access (XWR) [2:0] User access (XWR).

VADDR unsigned int

Virtual address of the access.

VMID unsigned int

Virtual machine identifier.

XS bool

XS Attribute.

nG enum

Flag indicating whether ASID will be matched.

MMU_TTB_READ

This event is triggered by reads caused by a translation table walk. Fields:

ABORTED bool

Was the walk successful.

DATA unsigned int

The data read.

DESC_KIND enum

The encoding scheme of the descriptors used for this walk.

IPA unsigned int

For stage 1, the IPA of the read.

LEVEL unsigned int

Translation table level.

LPAE bool

Is this for an LPAE translation (*DEPRECATED*, use DESC_KIND).

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSmem bool

Non secure memory access.

NSreq bool

Non secure request.

OUTADDR **unsigned int**

The output physical address.

PADDR **unsigned int**

The physical address of the read.

PASmem **enum**

Physical Address Space of the accessed memory.

PASreq **enum**

Physical Address Space of the requested lookup.

REGIME_EL **enum**

Entry matches in this translation regime.

REQUESTER **enum**

The requester of this table walk.

SIDE **enum**

Inst / Data.

STAGE **unsigned int**

Translation stage.

MMU_TTB_WRITE

This event is triggered by writes caused by a translation table walk. Fields:

ABORTED **bool**

Was the walk successful.

DATA **unsigned int**

The data written.

DESC_KIND **enum**

The encoding scheme of the descriptors used for this walk.

IPA **unsigned int**

For stage 1, the IPA of the write.

IS_HACDBS_ACCESS **bool**

If the access was due to HACDBS_CLEAN.

LEVEL **unsigned int**

Translation table level.

LPAE **bool**

Is this for an LPAE translation (*DEPRECATED*, use DESC_KIND).

MECID **unsigned int**

Memory Encryption Context(MEC) ID, if applicable.

MPAM_PARTID **unsigned int**

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSmem bool

Non secure memory access.

NSreq bool

Non secure request.

OUTADDR unsigned int

The output physical address.

PADDR unsigned int

The physical address of the write.

PASmem enum

Physical Address Space of the accessed memory.

PASreq enum

Physical Address Space of the requested lookup.

REGIME_EL enum

Entry matches in this translation regime.

SIDE enum

Inst / Data.

STAGE unsigned int

Translation stage.

MODE_CHANGE

Mode change. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

MODE enum

The new mode.

NON_SECURE enum

The core's new non-secure bit.

OLD_MODE enum

The old mode.

SECURITY_STATE enum

The new security state.

NOT_TAG_CHECKED_INST

MTE Tag not checked for current instruction.

PERIODIC

Called for every quantum. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The instruction count of this CPU.

PC unsigned int

The address of the next instruction to be executed on this CPU.

PMU_COUNTER_OVERFLOW

PMU counter overflow. Fields:

COUNTER_GROUP enum

Counter group.

COUNTER_RANGE enum

Counter range.

EVENT_ID unsigned int

Event Identifier, field is valid only for Event counter group.

INDEX unsigned int

Counter index (as selected by PMSELR), field is valid only for Event/Cycle counter groups.

INTERRUPT bool

Is interrupt enabled on overflow for this counter.

POINTER_AUTH

Called when execution hits a pointer authentication check. Fields:

AUTH_STATE bool

Authentication result.

KEY_USED enum

Pointer authentication key.

POTINDEX_UPDATE

Value of POT index for current pc changed. Fields:

NEW_POTINDEX unsigned int

New POTIndex.

OLD_POTINDEX unsigned int

Old POTIndex.

PREFETCH_MEMORY64

Prefetch from PRFM or PRFUM instructions. Fields:

PRFOP enum

Prefetch hint.

VADDR unsigned int

Virtual address of the location that should be prefetched.

PRELOAD64_REQUEST

Preload requests from PRFM or PRFUM instructions. Fields:

PRFOP enum

Prefetch hint.

VADDR unsigned int

Virtual address of the location that should be prefetched.

PRELOAD_DATA

Data preload from PLD instruction. Fields:

VADDR unsigned int

Virtual address of the data that should be preloaded.

PRELOAD_INST

Instruction preload from PLI instruction. Fields:

VADDR unsigned int

Virtual address of the instruction that should be preloaded.

PRE_CORE_LOAD

Trace just before a core load. Fields:

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

PRE_CORE_STORE

Trace just before a core store. Fields:

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

PRE_TTB_READ

This event is triggered before reads caused by a translation table walk. Fields:

ADDR unsigned int

The physical address of the read.

PSTATE

PSTATE fields update. Fields:

ALLINT unsigned int

All interrupt mask bit.

BTYPE unsigned int

Branch target identification bit.

CORE_NUM unsigned int

Core number in a multi processor.

DAIF unsigned int

Exception mask bits.

DIT unsigned int

Data Independent Timing bit.

EL unsigned int

Current Exception level field.

EXLOCK unsigned int

Exception return state lock bit.

IL unsigned int

Illegal Execution state bit.

NZCV unsigned int

Condition flags.

PACM unsigned int

PACM bit.

PAN unsigned int

Privileged Access Never state bit.

PM unsigned int

PMU exception mask bit.

PPEND unsigned int

PMU exception pending bit.

SM unsigned int

StreamingMode status bit.

SP unsigned int

Stack pointer register selection bit.

SS unsigned int

Software Step bit.

SSBS unsigned int

Speculative Store Bypass Safe bit.

TCO unsigned int

Tag Check Override bit.

TOUCHED unsigned int

Mask of the fields that have been touched since this trace has last been emitted. Note that touched does not necessarily mean the value changed. The index of the field is the bit index in the mask plus the index of this field plus one.

UAO unsigned int

User Access Override bit.

UINJ unsigned int

Undefined Instruction exceptions bit.

ZA unsigned int

ZA storage and ZT0 register enable bit.

nRW unsigned int

Current Execution state bit.

PSTATE_PACTIVE_OPMODE_TRANSITION

Trace pactive and Opmode mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Operating mode after the transition.

OLD_STATE enum

Operating mode before the transition.

PSTATE_PACTIVE_OPMODE_TRANSITION_DENIED

Trace the denied pactive and Opmode mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Operating mode after the transition.

OLD_STATE enum

Operating mode before the transition.

REASON string

Reason for the transition being denied.

PSTATE_PACTIVE_POWER_TRANSITION

Trace pactive and power mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Power mode after the transition.

OLD_STATE enum

Power mode before the transition.

PSTATE_PACTIVE_POWER_TRANSITION_DENIED

Trace the denied pactive and power mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Power mode after the transition.

OLD_STATE enum

Power mode before the transition.

REASON string

Reason for the transition being denied.

PSTATE_PACTIVE_THREAD_OPMODE_TRANSITION

Trace pactive and Opmode mode transitions for threads on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Operating mode after the transition.

OLD_STATE enum

Operating mode before the transition.

QUANTUM_START

Called at the start of every quantum.

RANGE_PRELOAD_REQUEST

Preload requests from RPRFM. Fields:

COUNT unsigned int

Number of blocks of data to access.

LENGTH unsigned int

Number of contiguous bytes to be accessed in each block.

OP unsigned int

Range prefetch operation .

REUSE unsigned int

Maximum number of bytes to be accessed before next RPRFM .

STRIDE signed int

Stride to advance after accessing LENGTH bytes.

VADDR unsigned int

Base virtual address of range prefetch request.

RUN_STATE

Run state transition. Fields:

INST_COUNT **unsigned int**

Ticks count at point of transition.

NEW **enum**

New run state.

OLD **enum**

Old run state.

SEMIHOSTING_CALL

Call of a semihost function occurred. Fields:

PC **unsigned int**

The program counter after the semihosting call.

SEMIHOSTING_PRECALL

About to call semihost. Fields:

EL2 **bool**

Use EL2 translation regime.

NS **bool**

Is Non-Secure.

REG_WIDTH **unsigned int**

The current register width in bytes.

SIGNAL

External signal state change. Fields:

SIGNAL **enum**

Signal that changed.

STATE **bool**

Signal asserted state.

SOFTWARE_STEP

Return the Debug Software Step state. Fields:

STATE **enum**

Software Step state.

SPSR

SPSR change. Fields:

CORE_NUM **unsigned int**

Core number in a multi processor.

MODE **enum**

Which of the banked SPSR registers is written.

OLD_VALUE unsigned int

The old SPSR value.

VALUE unsigned int

The new SPSR value.

START_COMPILE

Compilation started. Fields:

VADDR unsigned int

Instruction where compilation begins.

STG_MULTIPLE_MICRO_OPS_ADDRESS_RANGE

Address range for MTE stores with multiple uops. Fields:

ADDR_END unsigned int

Final address.

ADDR_START unsigned int

Starting address.

STORE_MULTIPLE_REGS_ADDRESS_RANGE

Address range for stores with multiple registers. Fields:

ADDR_END unsigned int

Final address.

ADDR_START unsigned int

Starting address.

SUBOPTIMAL_LDST_RETIRED

suboptimal load/store. Fields:

ACCESS_SIZE unsigned int

Log2 of the access width in bytes used for alignment checking (1: half-word, 2: word ...).

OPTIMAL_ALIGN_SELECTOR unsigned int

Selector to decide the optimal alignment size (1: access width, 2: 4-bytes, 3: 8-bytes, ..., 12: 4-kbytes).

VADDR unsigned int

The virtual address of the access.

SVE_INST_SPEC

SVE operations speculatively executed.

SVE_LD_RETIRED

SVE load instructions architecturally executed.

SVE_LOADS

SVE load accesses. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ACQREL enum

Is this an acquire/release.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DATA unsigned int

The data read or written.

ELEMENT_SIZE unsigned int

Width of each element.

ENDIANNESS enum

Byte ordering of this access.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

LOCK enum

Normal, exclusive or locked access.

MECID unsigned int

Memory Encryption Context ID, if applicable.

MECID2 unsigned int

Memory Encryption Context ID of the second page, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PADDR_TAG unsigned int

If tag checked, the physical address tag for the access.

PAGE_SIZE unsigned int

Page size as $\log_2(\text{size in bytes})$.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK, 2=Exclusive Failed, 3=Unsupported.

SIZE unsigned int

Width of the access in bytes. Only required if DATA is not traced.

TAG_CHECKED bool

Is this access tag checked.

TAG_CHECKED2 bool

Is the second page access tag checked.

TRANS bool

Is this a translated access.

VADDR unsigned int

The virtual address of the access.

SVE_MEM_PREFETCH

Scalable Vector Extension memory prefetch. Fields:

CACHE_LEVEL enum

The preferred cache level.

HINT enum

What the memory is expected to be used for.

SIZE unsigned int

The prefetch size in bytes.

TEMPORAL bool

Whether the data is temporal.

VADDR unsigned int

The virtual address of the prefetch.

SVE_REG_READ

Scalable Vector Extension register read. Fields:

ID enum

The register identifier.

SM bool

Whether the PE is in Streaming Mode.

VALUE unsigned int

The read data.

SVE_REG_UPDATE

Scalable Vector Extension register update. Fields:

ID enum

The register identifier.

SM bool

Whether the PE is in Streaming Mode.

VALUE unsigned int

The written data.

SVE_REG_WRITE

Scalable Vector Extension register write. Fields:

ID enum

The register identifier.

SM bool

Whether the PE is in Streaming Mode.

VALUE unsigned int

The written data.

SVE_STORES

SVE store accesses. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ACQREL enum

Is this an acquire/release.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DATA unsigned int

The data read or written.

ELEMENT_SIZE unsigned int

Width of each element.

ENDIANNESS enum

Byte ordering of this access.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

LOCK enum

Normal, exclusive or locked access.

MECID unsigned int

Memory Encryption Context ID, if applicable.

MECID2 unsigned int

Memory Encryption Context ID of the second page, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PADDR_TAG unsigned int

If tag checked, the physical address tag for the access.

PAGE_SIZE unsigned intPage size as $\log_2(\text{size in bytes})$.**PAS enum**

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK, 2=Exclusive Failed, 3=Unsupported.

SIZE unsigned int

Width of the access in bytes. Only required if DATA is not traced.

TAG_CHECKED bool

Is this access tag checked.

TAG_CHECKED2 bool

Is the second page access tag checked.

TRANS bool

Is this a translated access.

VADDR unsigned int

The virtual address of the access.

SVE_ST_RETIRED

SVE store instructions architecturally executed.

SVE_UNALIGNED_LDST_RETIRED

SVE load or store instructions architecturally executed that would cause alignment fault if A bit is set.

SVE_Z_PRED_MEM_ACCESS_INFO

Information about SVE/SME predicated accesses to Z registers (non Gather/Scatter). Fields:

MEM_ELEM_SIZE unsigned int

Width of elements accessed in memory (in bytes).

RnW bool

Whether this is a Read not a Write.

VA unsigned int

Virtual base address computed for accesses.

SYNC

Called for every synchronization, that is every quantum break. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The instruction count of this CPU.

LOCAL_QUANTUM unsigned int

The local quantum of this CPU.

LOCAL_TIME unsigned int

The local time of this CPU.

SYNC_REASON enum

Reason for sync.

SYSCALL

System call instruction executed. Fields:

IMM unsigned int

Immediate value of the system call instruction.

TYPE enum

System call type.

VADDR unsigned int

Instruction that caused the system call.

SYSREG_READ128

System Coprocessor 128 bit register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

REGNUM enum

Internal register number.

REGNUM_PHYS enum

Internal register number physically accessed (e.g. in presense of VHE redirection).

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_READ64

System Coprocessor register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REGNUM enum

Internal register number.

REGNUM_OPERAND enum

Register number corresponding to the instruction operand.

REGNUM_PHYS enum

Internal register number physically accessed (e.g. in presense of VHE redirection).

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_UPDATE128

Triggers when the system updates a register. Fields:

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG enum

Register number.

UNKNOWN unsigned int

Bits of the register which became unknown.

VALUE unsigned int

Value written to the register.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_UPDATE32

Triggers when the system updates a register. Fields:

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG enum

Register number.

UNKNOWN unsigned int

Bits of the register which became unknown.

VALUE unsigned int

Value written to the register.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_UPDATE64

Triggers when the system updates a register. Fields:

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG enum

Register number.

UNKNOWN unsigned int

Bits of the register which became unknown.

VALUE unsigned int

Value written to the register.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_WRITE128

System Coprocessor 128 bit register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

REGNUM enum

Internal register number.

REGNUM_PHYS enum

Internal register number physically accessed (e.g. in presense of VHE redirection).

UNDEF bool

The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int
opcode 2.

SYSREG_WRITE64

System Coprocessor register write. Fields:

CORE_NUM unsigned int
Core number in a multi processor.

CRm unsigned int
CRm.

CRn unsigned int
CRn.

NS bool
Secure or nonsecure banked register is accessed.

REGNUM enum
Internal register number.

REGNUM_OPERAND enum
Register number corresponding to the instruction operand.

REGNUM_PHYS enum
Internal register number physically accessed (e.g. in presense of VHE redirection).

UNDEF bool
The register accessed is undefined.

UPDATED_VALUE unsigned int
Updated value of the register now it has been written.

VALUE unsigned int
The new value written.

opc unsigned int
opcode 1.

opc0 unsigned int
opcode 0.

opc2 unsigned int
opcode 2.

TRBU_BUFFER_WRAP

DISPLAY Trace buffer current write pointer wrapped.

TRBU_TRIGGER_EVENT

DISPLAY Trace buffer Trigger Event.

TRBU_WRITE

DISPLAY PETU trace data has been written to the TRBU memory buffer at %{BUFFER_ADDRESS} with size %{SIZE} and data %{DATA}. Fields:

BUFFER_ADDRESS unsigned int

Virtual Address of start of buffer write.

DATA unsigned int

Data written to the buffer.

SIZE unsigned int

Size of sample written to buffer, in bytes.

UNALIGNED_LDST_RETIRED

Processor unaligned load/store. Fields:

ACCESS_SIZE unsigned int

Log2 of the access width in bytes used for alignment checking (1=>half-word, 2->word ...).

NUMBER_OF_BEATS unsigned int

Number of accesses (beats) in this burst (Total size in bytes is ACCESS_SIZE * NUMBER_OF_BEATS).

VADDR unsigned int

The virtual address of the access.

VFP_D_REGS

VFP/NEON D 64 bit register write. Fields:

ALIASING enum

Type of register aliasing used.

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a write to the corresponding bit in the register. For example, 0x1 would indicate a write to VALUE[0].

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

VFP_D_REGS_READ

VFP/NEON D 64 bit register read. Fields:

ID unsigned int

The register number.

VALUE unsigned int

The current value of the register.

VFP_Q_REGS

VFP/NEON Q 128 bit register write. Fields:

ALIASING enum

Type of register aliasing used.

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a write to the corresponding byte in the register. For example, 0x1 would indicate a write to VALUE[7:0].

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

VFP_Q_REGS_READ

VFP/NEON Q 128 bit register read. Fields:

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a read to the corresponding byte in the register. For example, 0x1 would indicate a read to VALUE[7:0].

VALUE unsigned int

The current value of the register.

VFP_SYS_REGS

Writes to the VFP/NEON units system registers. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ID enum

Which VFP system register is written.

MASK unsigned int

The mask indicating updated cumulative bits.

OLD_VALUE unsigned int

The register's old value overwritten.

VALUE unsigned int

The new value written to the VFP system register.

VFP_S_REGS

VFP/NEON S 32 bit register write. Fields:

ALIASING enum

Type of register aliasing used.

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a write to the corresponding bit in the register. For example, 0x1 would indicate a write to VALUE[0].

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

VFP_S_REGS_READ

VFP/NEON S 32 bit register read. Fields:

ID unsigned int

The register number.

VALUE unsigned int

The current value of the register.

VIRTUAL_TAG_ALLOCATED

Virtual tag information when allocated. Fields:

TAG_VADDR unsigned int

Virtual tag virtual address.

VADDR unsigned int

Data Virtual address.

VIRTUAL_TAG unsigned int

Virtual tag value.

WAYPOINT

Signals end of basic block execution. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ISSET enum

Origin instruction set.

IS_COND bool

Indicates if this is a conditional waypoint.

PC unsigned int

Origin address (or 0 if unavailable).

TAKEN bool

Indicates if this waypoint was taken.

TARGET unsigned int

Destination address.

TARGET_ISET enum

Destination instruction set.

WFE_END

WFE ended. Fields:

INST_COUNT unsigned int

Ticks count when leaving WFE.

WFE_EVENT_REGISTER

WFE event register status: set/clear, reason. Fields:

INST_COUNT unsigned int

Ticks count.

REASON enum

Reason for set/clear. Only REASON==0 (Cleared by WFE) clears the bit, all other reasons set it.

WFE_IGNORED

WFE ignored. Fields:

EVENT bool

This WFE was ignored because the event register was set.

INST_COUNT unsigned int

Ticks count when ignoring WFE.

TRAPPED bool

This WFE was trapped.

WFE_START

WFE entered. Fields:

INST_COUNT unsigned int

Ticks count when entering WFE.

WFI_END

WFI ended. Fields:

INST_COUNT unsigned int

Ticks count when leaving WFI.

WFI_IGNORED

WFI ignored. Fields:

DISABLED bool

This WFI was ignored because WFI is disabled.

INST_COUNT unsigned int

Ticks count when ignoring WFI.

REASON enum

specifies reason why WFI trace was ignored.

TRAPPED bool

This WFI was trapped.

WFI_START

WFI entered. Fields:

INST_COUNT unsigned int

Ticks count when entering WFI.

WFI_WAKEUP

WFI wakeup. Fields:

INST_COUNT unsigned int

Ticks count when WFI wakeup occurred.

REASON enum

Reason for wakeup.

2.18 ARM_Cortex-A520AE

This section describes the trace sources.

AA64_ASE_SVE_REGS

Modification of AArch64 Advanced SIMD or SVE Vector Registers (via SIMD or Scalar operations), SVE Predicate Registers and ZT0. Fields:

ID enum

The register identifier.

MASK unsigned int

Bitmask of the register to signal the modified bits in the VALUE field.

SM bool

Whether the PE is in Streaming Mode.

VALUE unsigned int

Value of AArch64 vector register.

ASYNC_MEMORY_FAULT

Context ID Register write. Fields:

FAULT unsigned int

Fault status present in ESR.ISS encoding for Serror in ARCH64 and present in DFSR encoding for external abort format in ARCH32.

PADDR unsigned int

Physical Address (or 0 if unavailable).

VADDR signed int

Virtual Address (or 0 if unavailable).

ATOMIC_END_ACCESS

Bus trace access for atomics. Fields:

ACCESS_FAIL bool

Memory access failed.

ADDR unsigned int

The virtual address of the access.

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

COMPARE_VALUE unsigned int

Compare value for CAS.

LOAD_VALUE unsigned int

Loaded values.

NSDESC unsigned int

The security state of the access.

OPERAND_VALUE unsigned int

Operation's operand.

OPERATION enum

Operation type.

PADDR unsigned int

The physical address of the access.

PRIV bool

Is this a privileged access?.

ATOMIC_START_ACCESS

Bus trace access for atomics. Fields:

ADDR unsigned int

The virtual address of the access.

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

COMPARE_VALUE unsigned int

Compare value for CAS.

NSDESC unsigned int

The security state of the access.

OPERAND_VALUE unsigned int

Operation's operand.

OPERATION enum

Operation type.

PADDR unsigned int

The physical address of the access.

PRIV bool

Is this a privileged access?.

ArchMsg.Error.exit_code

DISPLAY exit code: %{EXITCODE} component kind: %{KIND} reason: %{REASON}. Fields:

EXITCODE signed int

Exit code to be returned.

KIND string

Component kind that invoked the exit code trace.

REASON string

Reason why the exit code trace was invoked (optional).

ArchMsg.Warning.IgnoredWarmResetRequest

Event fires when software attempts to request warm reset through write to deprecated DBGPRCR.CWRR.

ArchMsg.Warning.RAS_ErrselrUnpredictableState

DISPLAY ERRSELR is in unpredictable state due to value greater than number_of_error_records written to it.

ArchMsg.Warning.RAS_UnpredictableStateWrite

DISPLAY %{REG_NAME} write ignored due to unpredictable ERRSELR state. Fields:

REG_NAME string

Name of register being written.

ArchMsg.Warning.ReportedRevisionRegisterRead

DISPLAY Reported revision and patch do not match that of the cpu. Fields:

ORIGINAL_VALUE unsigned int

The value the register would have based on the modelled revision.

REGISTER string

The name of the register that has ben read.

REPORTED_VALUE unsigned int

The value the register has been overridden to.

ArchMsg.Warning.Unpredictably Indexed PM Event Register

TAGS **UNPREDICTABLE** DISPLAY Event %{IsDirect:(Selector value|Register)} %{SEL} is >= %{N}. Fields:

IsDirect unsigned int

Direct Access rather than indirect.

N unsigned int

Number of Accessible Registers.

SEL unsigned int

Selector.

ArchMsg.Warning.cache_contents_unknown

A cache has been activated without being invalidated and may contain **UNKNOWN** content. DISPLAY %{SIDE:(D|I)}-Cache was enabled in %{REGIME:(SECURE|NON_SECURE|HYP)} regime but was not invalidated since power-on: cache lines could contain **UNKNOWN** content %{SIDE:(Note: D-side caches must invalidate SECURE lines even if the cache is only used by NON_SECURE code in order to ensure that unexpected dirty lines tagged SECURE are not naturally evicted)}. Fields:

REGIME unsigned int

Current translation regime.

SIDE unsigned int

0 d-side cache or unified cache, 1 i-side.

ArchMsg.Warning.dcimvac_matches_watchpoint

TAGS **UNPREDICTABLE** DISPLAY Watchpoints matching an AArch32 DCIMVA are implementation defined.

ArchMsg.Warning.decode_fieldmismatch

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with different fields %{FIELD1} and %{FIELD2}. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_frachitsnegative

TAGS **UNPREDICTABLE** DISPLAY VCVT Instruction is unpredictable with a negative number of fraction bits.

ArchMsg.Warning.decode_initblock

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable in an IT block.

ArchMsg.Warning.decode_initblocknotlast

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable in an IT block when not the last.

ArchMsg.Warning.decode_invalid_condition

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable when the condition is not AL.

ArchMsg.Warning.decode_invalidfieldcombination

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with the values in %{FIELD1} and %{FIELD2}. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_invalidvalue

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with the value in %{FIELD}. Fields:

FIELD enum

Field name.

ArchMsg.Warning.decode_mem_hint_unallocated

TAGS **UNPREDICTABLE** DISPLAY There is no memory hint allocated to this bit pattern.

ArchMsg.Warning.decode_registermatch

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with %{FIELD1} and %{FIELD2} as the same register. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_registermismatch

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with %{FIELD1} and %{FIELD2} as different registers. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_registeroutofrange

TAGS **UNPREDICTABLE** DISPLAY Floating-point instruction is unpredictable indexing beyond the end of the register bank.

ArchMsg.Warning.decode_sbzsbo

TAGS **UNPREDICTABLE** DISPLAY Reserved bits in the instruction are not canonical.

ArchMsg.Warning.decode_transactiontoobig

TAGS **UNPREDICTABLE** DISPLAY Attempt to perform a memory transaction of more than 128 bytes.

ArchMsg.Warning.decode_unpred_other

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable.

ArchMsg.Warning.decode_unpreduseofpc

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable when %{FIELD} is PC. Fields:

FIELD enum

Field name.

ArchMsg.Warning.decode_unpreduseofr13

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable when %{FIELD} is R13. Fields:

FIELD enum

Field name.

ArchMsg.Warning.decode_unpreduseofr13inreglist

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with R13 in register-list.

ArchMsg.Warning.decode_useofd16orhigher

TAGS **UNPREDICTABLE** DISPLAY Attempt to use register D16 or higher where it is not allowed.

ArchMsg.Warning.decode_writebackandbaseinlist

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with write back when the base register is in the transfer list.

ArchMsg.Warning.decode_zeroregistersinlist

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable when no registers are to be transferred.

ArchMsg.Warning.misaligned_ptrs

DISPLAY TRBPTR_EL1(%{TRBPTR_EL1}) or TRBTRG_EL1(%{TRBTRG_EL1}) are not aligned as per TRBIDR_EL1.Align(%{TRBIDR_EL1}). Fields:

TRBIDR_EL1 unsigned int

TRBIDR_EL1.Align value.

TRBPTR_EL1 unsigned int

Current write pointer.

TRBTRG_EL1 unsigned int

Trigger pointer.

ArchMsg.Warning.out_of_range_ptrs

DISPLAY TRBPTR_EL1(%{TRBPTR_EL1}) is out of range as per TRBBASER_EL1(%{TRBBASER_EL1}) or TRBLIMITR_EL1.LIMIT(%{TRBLIMITR_EL1}). Fields:

TRBBASER_EL1 unsigned int

Base pointer.

TRBLIMITR_EL1 unsigned int

Limit pointer.

TRBPTR_EL1 unsigned int

Current write pointer.

ArchMsg.Warning.recursive_branch

An instruction is performing a branch that targets the same instruction. If this was intended then a WFI might be more effective. DISPLAY recursive branch detected at PC=%{PC}. PRIMARY KEY PC. Fields:

PC unsigned int

Address of instruction causing the branch.

ArchMsg.Warning.recursive_exception

An instruction has generated an exception that targets the same instruction. DISPLAY recursive %{TYPE} exception detected at PC=%{PC} (CPSR=%{CPSR}). PRIMARY KEY PC. Fields:

CPSR unsigned int

Processor state.

PC unsigned int

Address of instruction causing the exception.

TYPE enum

Type of exception.

ArchMsg.Warning.reentrant_vector_catch

TAGS UNPREDICTABLE DISPLAY Vector catch debug event on Prefetch or Data abort vector %{ADDR}. Fields:

ADDR unsigned int

vector address.

ArchMsg.Warning.reserved_it_state

TAGS UNPREDICTABLE DISPLAY Execution with reserved IT state %{ITSTATE}. Fields:

ITSTATE unsigned int

reserved IT state.

ArchMsg.Warning.tlb_contents_unknown

Translation has been enabled without invalidating the TLB. DISPLAY Translation was enabled in the %{PAS:(Secure|Non-secure|Root|Realm)} EL%{EL:d} translation regime (vmid=

{VMID:x}) without invalidating {INVALIDITY:(|d-side |i-side |)} entries since power-on: {INVALIDITY:(|D||I)} TLB could contain **UNKNOWN** content. Fields:

EL unsigned int

Exception level of translation regime.

INVALIDITY unsigned int

Which TLB may be invalid (if TLBs are separate).

NON_SECURE bool

Whether the translation regime is non-secure.

PAS unsigned int

Physical address space of translation regime.

VMID unsigned int

Current VMID, if applicable.

ArchMsg.Warning.unknown_DLR_DSPSR

Exiting debug state with unknown DLR or DSPSR.

ArchMsg.Warning.unknown_ELR_SPSR

Returning from debug exception with unknown ELR or SPSR.

ArchMsg.Warning.unpred_FAR_on_LRCPC3_non_atomic_mem_access

TAGS **UNPREDICTABLE** Unpredictable FAR/FSTATUS on LRCPC3 pair instructions resulting in two separate memory accesses
DISPLAY Unpredictable FAR on LRCPC3 non atomic memory access at {ADDR1} with FSTATUS {FSTATUS1} or {ADDR2} with FSTATUS {FSTATUS2}. Fields:

ADDR1 unsigned int

Possible FAR address 1.

ADDR2 unsigned int

Possible FAR address 2.

FSTATUS1 unsigned int

Possible fault status value 1.

FSTATUS2 unsigned int

Possible fault status value 2.

ArchMsg.Warning.unpred_async_tag_check_fail_on_sve_speculative_load_do_update_ffr

Unpredictable set to FALSE of the FFR element associated to SVE speculative load with an async tag check fail TAGS **UNPREDICTABLE**.

ArchMsg.Warning.unpredictable_a32_breakpoint

TAGS **UNPREDICTABLE** DISPLAY Unpredictable breakpoint on a A32 instruction at {ADDR1} Offset {OFFSET} BAS {BAS} for a {IS_ADDRESS_MISMATCH_BREAKPOINT:(Mismatch|Match)} Breakpoint number {N:d} Breakpoint address {ADDR2}. Fields:

ADDR1 unsigned int

Address.

ADDR2 unsigned int

Breakpoint address.

BAS unsigned int

BAS field.

IS_ADDRESS_MISMATCH_BREAKPOINT bool

Is a Mismatch breakpoint, not a Match breakpoint.

N unsigned int

Breakpoint number.

OFFSET unsigned int

Always has a value of zero.

ArchMsg.Warning.unpredictable_banked_register_access

TAGS **UNPREDICTABLE** DISPLAY Access to banked register (R=%{R:d} SYSm=%{SYSM:2x}) is unpredictable. Fields:

R unsigned int

R bit.

SYSM unsigned int

SYSm.

ArchMsg.Warning.unpredictable_ccfail_undef

TAGS **UNPREDICTABLE** DISPLAY An instruction %{OPCODE} at VA=%{VADDR} failed its condition codes check but would otherwise have caused a non-data-dependent trap or undefined exception. Behaviour is implementation-defined choice of exception or nop. Fields:

OPCODE unsigned int

opcode of the instruction.

VADDR unsigned int

Virtual address of the instruction being executed.

ArchMsg.Warning.unpredictable_ici_bits

TAGS **UNPREDICTABLE** DISPLAY ICI bits do not apply or are architecturally unpredictable for instruction.

ArchMsg.Warning.unpredictable_t32_breakpoint

TAGS **UNPREDICTABLE** DISPLAY Unpredictable breakpoint on a T32 instruction at %{ADDR1} Offset %{OFFSET} BAS %{BAS} for a %{IS_ADDRESS_MISMATCH_BREAKPOINT:(Mismatch|Match)} Breakpoint number %{N:d} Breakpoint address %{ADDR2}. Fields:

ADDR1 unsigned int

Address.

ADDR2 unsigned int

Breakpoint address.

BAS unsigned int

BAS field.

IS_ADDRESS_MISMATCH_BREAKPOINT bool

Is a Mismatch breakpoint, not a Match breakpoint.

N unsigned int

Breakpoint number.

OFFSET unsigned int

Indicates which halfword of a 32-bit Thumb instruction the breakpoint is set on. 0 for the first halfword or if the instruction is 16 bits wide, or 1 for the second halfword.

ArchMsg.Warning.unpredictable_t32_vector_catch

TAGS **UNPREDICTABLE** DISPLAY Unpredictable Vector catch exception on a T32 instruction at %{ADDR} Offset %{OFFSET}. Fields:

ADDR unsigned int

Address.

OFFSET unsigned int

Indicates which halfword of a 32-bit Thumb instruction the vector catch exception is set on. 0 for the first halfword or if the instruction is 16 bits wide, or 1 for the second halfword.

ArchMsg.Warning.unpredictable_watchpoint_far

TAGS **UNPREDICTABLE** DISPLAY The value for FAR/EDWAR for hitting this watchpoint may be anywhere in the range %{LOWER_BOUND} to %{UPPER_BOUND}. Fields:

LOWER_BOUND unsigned int

the lowest address accessed by the instruction that triggered the watchpoint.

UPPER_BOUND unsigned int

the highest watchpointed address accessed by that instruction.

ArchMsg.Warning.unpredictable_watchpoint_far_isscmp

DISPLAY The value for FAR/EDWAR for hitting this watchpoint may be anywhere in the range %{LOWER_BOUND} to %{UPPER_BOUND}. Fields:

LOWER_BOUND unsigned int

the lowest address accessed by the instruction that triggered the watchpoint.

UPPER_BOUND unsigned int

the highest watchpointed address accessed by that instruction.

ArchMsg.Warning.warning_AdvSIMDExpandImmUnexpectedZero

TAGS **UNPREDICTABLE** DISPLAY AdvSIMDExpandImm may treat this immediate value as **UNPREDICTABLE**.

ArchMsg.Warning.warning_ConditionalSMC

TAGS **UNPREDICTABLE** DISPLAY SMC instruction has **UNPREDICTABLE** effects when conditional and when combined with traps.

ArchMsg.Warning.warning_TLBID_out_of_range

TLBID out of range in TLBI according to the configuration of NIS, NVIS, NOS and NVOS.
Fields:

NIS unsigned int

NIS.

NOS unsigned int

NOS.

NVIS unsigned int

NVIS.

NVOS unsigned int

NVOS.

TLBID unsigned int

The TLBID value used in the TLBI instruction.

ArchMsg.Warning.warning_TTL_or_TG_mismatches_current_translation_granule

TTL or TG mismatches current translation granule in TLBI VA/RVA/IPA/RIPA. Fields:

CURRENT_TG unsigned int

Current TG as indicated by TCR_ELx or VTCR_EL2.

TLBI_TG unsigned int

TG (or TTL[3:2]) used in TLBI instruction.

ArchMsg.Warning.warning_access_crosses_page_boundary

TAGS UNPREDICTABLE DISPLAY Access crosses page boundary.

ArchMsg.Warning.warning_access_crosses_page_boundary_has_fault

TAGS UNPREDICTABLE DISPLAY Access that crosses page boundary has fault(s).

ArchMsg.Warning.warning_access_crosses_page_boundary_has_fault_on_both_pages

TAGS UNPREDICTABLE DISPLAY Access that crosses page boundary has a fault on each page.
Fields:

ADDR1 unsigned int

FAR address on first page.

ADDR2 unsigned int

FAR address on second page.

FSTATUS1 unsigned int

Fault status on first page.

FSTATUS2 unsigned int

Fault status on second page.

ArchMsg.Warning.warning_access_crosses_page_boundary_spanning_different_memory

TAGS UNPREDICTABLE DISPLAY Access at address %{ADDR} crosses page from MEM_TYPE %{MEMTYPE_PAGE1} to %{MEMTYPE_PAGE2}. Fields:

ADDR unsigned int

address of access crossing page.

MEMTYPE_PAGE1 signed int

First Page's memory type (0-Normal, 1-Non-Normal).

MEMTYPE_PAGE2 signed int

Second Page's memory type (0-Normal, 1-Non-Normal).

ArchMsg.Warning.warning_access_wraps_around_memory

TAGS **UNPREDICTABLE** DISPLAY Access wraps around memory in %{WIDTH:d}bit mode.
Fields:

WIDTH unsigned int

Address width.

ArchMsg.Warning.warning_bcr_linking_status

TAGS **UNPREDICTABLE** DISPLAY DBGBCR.{BT2,BT} combination gives unpredictable behaviour.

ArchMsg.Warning.warning_bcr_mask_reserved

TAGS **UNPREDICTABLE** DISPLAY DBGBCR_ELn.MASK is set to a reserved value.

ArchMsg.Warning.warning_bvr_masked_bit_not_zero

TAGS **UNPREDICTABLE** DISPLAY DBGBVRn_EL1.VA contains a bit masked by DBGBCRn_EL1.MASK that is not zero.

ArchMsg.Warning.warning_ccsidr_unimplemented_level

TAGS **UNPREDICTABLE** DISPLAY CCSIDR read while CSSELR %{CSSELR:x} points to an unimplemented cache level. Fields:

CSSELR unsigned int

current effective value of CSSELR.

ArchMsg.Warning.warning_change_to_ns_when_tge_set

TAGS **UNPREDICTABLE** DISPLAY Attempting to change NS to 1 when HCR.TGE = 1.

ArchMsg.Warning.warning_contiguous_bit_check_abort

TAGS **UNPREDICTABLE** DISPLAY An abort occurred whilst attempting to check TLB entry is contiguous at %{ENTRY_ADDR}. Fields:

ENTRY_ADDR unsigned int

address of conflicting TLB entry.

ArchMsg.Warning.warning_contiguous_bit_error

TAGS **UNPREDICTABLE** DISPLAY TLB entry at %{ENTRY_ADDR} was not contiguous with entry at %{CONTIG_ADDR} Contents are %{ENTRY_DATA} but expected %{CONTIG_DATA}.
Fields:

CONTIG_ADDR unsigned int

address of the TLB first read, with which this entry is expected to be contiguous.

CONTIG_DATA unsigned int

expected contents based on the entry at CONTIG_ADDR.

ENTRY_ADDR unsigned int

address of conflicting TLB entry.

ENTRY_DATA unsigned int

contents of conflicting TLB entry.

ArchMsg.Warning.warning_contiguous_bit_error_D128

TAGS **UNPREDICTABLE** DISPLAY TLB entry at %{ENTRY_ADDR} was not contiguous with entry at %{CONTIG_ADDR} Contents are %{ENTRY_DATA_HI} %{ENTRY_DATA_LO} but expected %{CONTIG_DATA}. Fields:

CONTIG_ADDR unsigned int

address of the TLB first read, with which this entry is expected to be contiguous.

CONTIG_DATA unsigned int

expected contents based on the entry at CONTIG_ADDR.

ENTRY_ADDR unsigned int

address of conflicting TLB entry.

ENTRY_DATA_HI unsigned int

Higher bit contents of conflicting TLB entry.

ENTRY_DATA_LO unsigned int

Lower bit contents of conflicting TLB entry.

ArchMsg.Warning.warning_cp10_cp11_mismatch

TAGS **UNPREDICTABLE** DISPLAY Unpredictable configuration of CP10 and CP11 controls at EL %{EL:d}. Fields:

EL unsigned int

EL owning the control.

ArchMsg.Warning.warning_cp10_cp11_reserved_value

TAGS **UNPREDICTABLE** DISPLAY Unpredictable configuration of CPACR CP10 (%{CP10}) or CP11 (%{CP11}). Fields:

CP10 unsigned int

CP10 access.

CP11 unsigned int

CP11 access.

ArchMsg.Warning.warning_csselr_level_out_of_range

TAGS **UNPREDICTABLE** DISPLAY CSSELR.Level out of range, written value %{LEVEL} with only %{IMPLEMENTED:d} levels implemented. Fields:

IMPLEMENTED unsigned int

number of cache levels implemented.

LEVEL unsigned int

written value of CSSELR.Level.

TYPE enum

The type of cache selected by CSSELR.InD and TnD.

ArchMsg.Warning.warning_debug_flow_control_bits_not_obeyed

TAGS **UNPREDICTABLE** DISPLAY Flow control bits for %{REGISTER} were not obeyed, causing %{ERROR}. Fields:

ERROR enum

type of error resulting.

REGISTER enum

register under consideration.

ArchMsg.Warning.warning_debug_register_access_during_reset

TAGS **UNPREDICTABLE** DISPLAY Unpredictable %{IS_WRITE:(read|write)} access to debug register offset %{OFFSET} during reset. Fields:

IS_WRITE unsigned int

Write Not Read.

OFFSET unsigned int

Register Offset.

ArchMsg.Warning.warning_decode_cps_inconsistent_fields

TAGS **UNPREDICTABLE** DISPLAY The mode setting fields in the CPS instruction are inconsistent.

ArchMsg.Warning.warning_decode_invalid_state

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable in current exception-level/security state.

ArchMsg.Warning.warning_default_cacheable_mmu_on

TAGS **UNPREDICTABLE** DISPLAY Default cacheable is enabled (HCR.DC = 1) while MMU is enabled.

ArchMsg.Warning.warning_default_cacheable_vmmu_off

TAGS **UNPREDICTABLE** DISPLAY Default cacheable is enabled (HCR.DC = 1) while HCR.VM = 0.

ArchMsg.Warning.warning_deprecated_wvr_bit_2_set

TAGS **UNPREDICTABLE** DISPLAY DBGWVR%{N:d}_EL1=%{WVR:h} has bit 2 set, deprecated in ARMv8. Fields:

N unsigned int

index.

WVR unsigned int

DBGWVR.

ArchMsg.Warning.warning_eret_while_software_step_active_pending

Missing ISB between setting MDSCR_EL1.SS and ERET TAGS **UNPREDICTABLE**.

ArchMsg.Warning.warning_exclusive_to_non_normal

TAGS **UNPREDICTABLE** DISPLAY Exclusive Access to Non-Normal Memory %{ADDR}. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_exclusive_to_non_writeback

TAGS **UNPREDICTABLE** DISPLAY Exclusive Access to Non-Writeback-Cacheable Memory %{ADDR}. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_execute_from_device_memory

TAGS **UNPREDICTABLE** DISPLAY An attempt was made to execute from Device memory.

ArchMsg.Warning.warning_illegal_cpsr_mode

TAGS **UNPREDICTABLE** DISPLAY Writing an illegal or unimplement mode (%{NEW_MODE:h}) to CPSR was unpredictable. Fields:

NEW_MODE unsigned int

New value of CPSR.M.

ArchMsg.Warning.warning_illegal_srs_mode

TAGS **UNPREDICTABLE** DISPLAY Illegal or **UNPREDICTABLE** mode (%{MODE:h}) used for SRS instruction. Fields:

MODE unsigned int

mode for Banked SP.

ArchMsg.Warning.warning_implementation_defined_read_debug_register_in_SCS

Read Access to Debug registers from the processor is IMPLEMENTATION DEFINED TAGS IMP_DEF. Fields:

PPB_OFFSET unsigned int

debug register offset.

ArchMsg.Warning.warning_implementation_defined_sequential_security_transitions_supported

The behavior of sequential instruction fetches that cross from non-secure to secure memory and contain an SG instruction is CONSTRAINED **UNPREDICTABLE** TAGS **UNPREDICTABLE**.

ArchMsg.Warning.warning_implementation_defined_stack_limit_check_supported

It is **IMPLEMENTATION DEFINED** whether stack pointer limit checking is performed for this instruction TAGS IMP_DEF. Fields:

ADDRESS unsigned int

Address of instruction.

OPCODE unsigned int

instruction opcode.

ArchMsg.Warning.warning_implementation_defined_taggednonsharedmemorytype

TAGS IMP_DEF DISPLAY Memory attribute has Tagged, non shared memory type; it is IMPLEMENTATION DEFINED whether the memory location is treated as Tagged or non-Tagged.

ArchMsg.Warning.warning_implementation_defined_write_debug_register_in_SCS

Write Access to Debug registers from the processor is IMPLEMENTATION DEFINED TAGS IMP_DEF. Fields:

DATA unsigned int

data attempted to be written.

PPB_OFFSET unsigned int

debug register offset.

ArchMsg.Warning.warning_invalid_tcr_granule

TAGS UNPREDICTABLE DISPLAY TCR.TG%{TG_ID:(0|1)} bits have been set to a granule size %{{REQUEST:d}K not implemented - using %{{SUBSTITUTE:d}K. Fields:

REQUEST signed int

page size requested (or 0 for reserved).

SUBSTITUTE signed int

best guess available page size.

TG_ID bool

bits TG0 or TG1.

ArchMsg.Warning.warning_load_multiple_user_registers_from_user_mode

TAGS UNPREDICTABLE DISPLAY An LDM(user registers) instruction executed from user mode.

ArchMsg.Warning.warning_mte_in_mem_attr_not_supported

TAGS UNPREDICTABLE DISPLAY Memory attribute %{{ATTR}} has Tagged memory type but Armv8.5-MTE is not implemented. Fields:

ATTR unsigned int

Memory attribute.

ArchMsg.Warning.warning_reserved_breakpoint_state_match

TAGS UNPREDICTABLE DISPLAY %{{IS_BREAKPOINT:(Watchpoint|Breakpoint)}} programmed with a reserved combination of HMC, SSC and %{{IS_BREAKPOINT:(PAC|PMC)}}. Fields:

IS_BREAKPOINT unsigned int

Is a breakpoint, not a watchpoint.

ArchMsg.Warning.warning_shareability

TAGS UNPREDICTABLE DISPLAY Unpredictable: combination of the 1st and 2nd stages of translation is Normal Inner Non-Cacheable, Outer Non-Cacheable.

ArchMsg.Warning.warning_software_step_set_while_enabled

TAGS UNPREDICTABLE DISPLAY MDSCR_EL1.SS set to 1 while software step debug exceptions are enabled.

ArchMsg.Warning.warning_thumb_instruction_wraps_around_memory

TAGS UNPREDICTABLE DISPLAY Thumb instruction wraps around memory.

ArchMsg.Warning.warning_ttbr_sbz_bits_are_not_zero

TAGS UNPREDICTABLE DISPLAY %{TTBR}[%{MSB:d}:0] should be zero, but are not. Fields:

MSB unsigned int

x-1.

TTBR enum

which TTBR register.

ArchMsg.Warning.warning_unaligned_address_dbgdtrrx_write

TAGS UNPREDICTABLE DISPLAY Unaligned address %{ADDR} in X0/R0 when DBGDTRRX written in memory access mode. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unaligned_address_dbgdtrtx_read

TAGS UNPREDICTABLE DISPLAY Unaligned address %{ADDR} in X0/R0 when DBGDTRTX read in memory access mode. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unaligned_to_device

TAGS UNPREDICTABLE DISPLAY Unaligned Access to Device Memory %{ADDR}. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unaligned_to_strongly_ordered

TAGS UNPREDICTABLE DISPLAY Unaligned Access to Strongly Ordered Memory %{ADDR}. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unknown_sau_rnr

TAGS UNPREDICTABLE DISPLAY SAU_RNR was set to an unsupported value.

ArchMsg.Warning.warning_unpredictable_AIRCR_PRIS_and_BFHFNMINs

TAGS UNPREDICTABLE DISPLAY The effect of setting both AIRCR.BFHFNMINs and AIRCR.PRIS to 1 is UNPREDICTABLE.

ArchMsg.Warning.warning_unpredictable_AIRCR_VECTCLRACTIVE_when_not_in_debug

TAGS **UNPREDICTABLE** DISPLAY The effect of writing a 1 to AIRCR.VECTCLRACTIVE if the processor is not halted in Debug state is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_AIRCR_VECTRESET_and_SYSRESETREQ

TAGS **UNPREDICTABLE** DISPLAY When the processor is halted in Debug state, if a write to the register writes a 1 to both VECTRESET and SYSRESETREQ, the behavior is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_AIRCR_VECTRESET_when_not_in_debug

TAGS **UNPREDICTABLE** DISPLAY The effect of writing a 1 to AIRCR.VECTRESET if the processor is not halted in Debug state is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_AIRCR_incorrect_VKEY

TAGS **UNPREDICTABLE** DISPLAY The value 0x05FA must be written to AIRCR.VKEY, otherwise the register write is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_EXC_RETURN_Reserved_Bit

EXC_RETURN[23:7] are reserved with the special condition that all bits should be written as one. Values other than all 1s are **UNPREDICTABLE**. TAGS **UNPREDICTABLE**. Fields:

EXC_RETURN unsigned int

Special PC value.

SBO_MASK unsigned int

Which bits Should Be One to not UNPRED.

ArchMsg.Warning.warning_unpredictable_PMEVTYPEPn_ELO_VS_reserved_value

TAGS **UNPREDICTABLE** DISPLAY Unpredictable reserved value of PMEVTYPEN<n>_ELO.VS.

ArchMsg.Warning.warning_unpredictable_change_to_DEMCR_MON_STEP_at_insufficient_priority

TAGS **UNPREDICTABLE** DISPLAY The effect of changing DEMCR.MON_STEP at an execution priority that is lower than the priority of the DebugMonitor exception is **CONSTRAINED UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_change_to_priority_of_active_exception

TAGS **UNPREDICTABLE** DISPLAY Changing the priority of an active exception is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_clear_lspact

TAGS **UNPREDICTABLE** DISPLAY **UNPREDICTABLE** state on context switch. Fields:

DRVEXC_TAKEN unsigned int

Derived exception taken, not pended.

REACHED_LAST_STORE unsigned int

Only the last faulted.

ArchMsg.Warning.warning_unpredictable_dbg_exit_unaligned_dlr

TAGS **UNPREDICTABLE** DISPLAY Debug exit to AArch32 with DLR[0] set to 1. PC[0] can be set to 0 or to DLR[0].

ArchMsg.Warning.warning_unpredictable_exception_catch

TAGS **UNPREDICTABLE** DISPLAY Generation of an exception catch event is unpredictable.

ArchMsg.Warning.warning_unpredictable_exception_return_inconsistent_state

TAGS **UNPREDICTABLE** DISPLAY state on exception return is unpredictable.

ArchMsg.Warning.warning_unpredictable_exception_return_instruction

TAGS **UNPREDICTABLE** DISPLAY exception return instruction is unpredictable.

ArchMsg.Warning.warning_unpredictable_in_debug_state

TAGS **UNPREDICTABLE** DISPLAY Instruction is **UNPREDICTABLE** when executed in debug state.

ArchMsg.Warning.warning_unpredictable_mair_encoding

TAGS **UNPREDICTABLE** DISPLAY Memory attribute %{ATTR} has invalid MAIR encoding; this may lead to unpredictable results. Fields:

ATTR unsigned int

Memory attribute.

ArchMsg.Warning.warning_unpredictable_pmu_counter_access

TAGS **UNPREDICTABLE** DISPLAY Access to PMU counters from non-secure EL0 or EL1 is **UNPREDICTABLE** with MDCR_EL2.HPMN set to 0.

ArchMsg.Warning.warning_unpredictable_prioritization_breakpoint_match_vector_catch

TAGS **UNPREDICTABLE** DISPLAY Two events with the same priority occurred at the same instruction: Address Matching Vector Catch debug event, and Breakpoint debug event. It is constrained unpredictable which is taken.

ArchMsg.Warning.warning_unpredictable_stack_selection

TAGS **UNPREDICTABLE** DISPLAY **UNPREDICTABLE** state on context switch.

ArchMsg.Warning.warning_unpredictable_tsize_out_of_range

TAGS **UNPREDICTABLE** DISPLAY %{TCR}.T%{TTBR:d}SZ (%{TSIZE:d} bits) is out of range. Must be between 25 and %{TSIZE_MAX:d}. Fields:

TCR enum

Which TCR.

TSIZE signed int

Value of TSize.

TSIZE_MAX signed int

The maximum allowed value.

TTBR unsigned int

Which TTBR region.

ArchMsg.Warning.warning_unpredictable_unaligned_pc_as_base_register

TAGS **UNPREDICTABLE** DISPLAY The PC value must be word-aligned, otherwise the behavior of the instruction is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_unaligned_pop_stack

TAGS **UNPREDICTABLE** DISPLAY **CONSTRAINED UNPREDICTABLE** unaligned stack frame %{FRAME_ADDR:x} in exception-return. Fields:

FRAME_ADDR unsigned int
unaligned stack frame address.

ArchMsg.Warning.warning_unpredictable_unaligned_to_non_writeback
TAGS **UNPREDICTABLE** DISPLAY Unaligned Access to Non-Writeback-Cacheable Memory.

ArchMsg.Warning.warning_unpredictable_vmsa_memattrib
TAGS **UNPREDICTABLE** DISPLAY Unpredictable vmsa memory attribute with texcb value programmed to %{TEXCB} when tex remap is %{TEX_REMAP}. Fields:

TEXCB unsigned int
texcb value.

TEX_REMAP bool
use tex remap.

ArchMsg.Warning.warning_unpredictable_vtcr_t0sz_sl0
TAGS **UNPREDICTABLE** DISPLAY The combination of VTCR.TOSZ=%{TOSZ:d} and VTCR.SLO=%{SLO:d} is unpredictable. Fields:

SLO signed int
VTCR.SLO, starting level for stage 2 translation table walks.

TOSZ signed int
VTCR.TOSZ, required input address range.

ArchMsg.Warning.warning_unpredictable_write_DHCSR
TAGS **UNPREDICTABLE** DISPLAY **UNPREDICTABLE** set of bit-changes in DHCSR. Fields:

NEW unsigned int
data attempted to be written.

OLD unsigned int
previous value.

ArchMsg.Warning.warning_unpredictable_write_SHCSR
TAGS **UNPREDICTABLE** DISPLAY **UNPREDICTABLE** setting a pending bit to 1 for an exception with priority >= execution priority.

ArchMsg.Warning.warning_unsupported_access_to_memory_mapped_register
TAGS **UNPREDICTABLE** DISPLAY Access to memory mapped register is unsupported.

ArchMsg.Warning.warning_user_jmcr_access
TAGS **UNPREDICTABLE** DISPLAY User %{WRITE:(write|read)} access to JMCR is unpredictable. Fields:

WRITE unsigned int
access is write.

ArchMsg.Warning.warning_wcr_mask_and_bas
TAGS **UNPREDICTABLE** DISPLAY DBGWCR_EL1.MASK is non-zero and BAS is not 0xff.

ArchMsg.Warning.warning_wcr_mask_reserved

TAGS **UNPREDICTABLE** DISPLAY DBGWCRn_EL1.MASK is set to a reserved value.

ArchMsg.Warning.warning_wcr_non_configuous_bas

TAGS **UNPREDICTABLE** DISPLAY DBGWCRn_EL1.BAS has non-contiguous set of ones.

ArchMsg.Warning.warning_wvr_masked_bit_not_zero

TAGS **UNPREDICTABLE** DISPLAY DBGWVRn_EL1.VA contains a bit masked by DBGWCRn_EL1.MASK that is not zero.

ArchMsg.Why.why_branch_target_exception

A branch target exception has occurred. DISPLAY branch target exception: target pc={PC}, target opcode={OPCODE}. PRIMARY KEY PC. Fields:

BTYPE unsigned int

pstate.BTYPE at time branch taken.

OPCODE unsigned int

opcode at branch target.

PC unsigned int

Virtual Address of branch target.

ArchMsg.Why.why_illegal_state

An illegal mode change has occurred. DISPLAY illegal mode change: {MESSAGE}. Fields:

MESSAGE string

The message.

BRANCH_MISPREDICT

Simulating branch mispredict. Fields:

PC unsigned int

Origin address (or 0 if unavailable).

BRA_DIR

Direct branches, to immediate address. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISSET enum

The instruction set of the branch instruction.

IS_COND bool

Indicates if this is a conditional branch.

IS_HINTED bool

Whether the conditional instruction is hinted.

IS_ISB bool

Whether the instruction is ISB.

IS_LINK bool

Indicates if this is a branch with link.

PC unsigned int

The address of the branch instruction.

TARGET_ISET enum

The instruction set after the branch.

TARGET_PC unsigned int

The address the instruction branches to.

BRA_INDIR

Indirect branches, perhaps to a register. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISET enum

The instruction set of the branch instruction.

IS_COND bool

Indicates if this is a conditional branch.

IS_HINTED bool

Whether the conditional instruction is hinted.

IS_ISB bool

Whether the instruction is ISB.

IS_LINK bool

Indicates if this is a branch with link.

PC unsigned int

The address of the branch instruction.

TARGET_ISET enum

The instruction set after the branch.

TARGET_PC unsigned int

The address the instruction branches to.

BRA_INDIR_CCFAIL

Indirect branches fail, perhaps to a register.

BRA_RET_CCFAIL

Branch return fail, perhaps to a register.

CACHE_MAINTENANCE_OP

Cache Maintenance Operation. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

DATA unsigned int

Specified MVA or Set/way.

FUNCTION enum

Clean / Invalidate.

SCOPE enum

Affected region.

SIDE enum

Inst / Data.

CCFAIL

Conditional instruction condition check fail. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CCFAIL_UNC

Unconditional instruction condition check fail. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CCPASS

Conditional instruction condition check pass. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CCPASS_UNC

Unconditional instruction condition check pass. Fields:

COND **enum**

The condition of the conditional instruction.

CORE_NUM **unsigned int**

Core number in a multi processor.

PC **unsigned int**

The address of the conditional instruction.

CODE_CACHE_MAINT

Code cache maintenance. Fields:

END_INCL **unsigned int**

Inclusive end address for VA or PA requests.

NS **bool**

Non-secure world for PA.

START **unsigned int**

Start address for VA or PA requests.

TYPE **enum**

Request type.

COMPILE_BLOCK_END

Last instruction of basic block translated. Fields:

VADDR **unsigned int**

Address of next instruction after this basic block.

COMPILE_BLOCK_START

First instruction of basic block translated. Fields:

VADDR **unsigned int**

Address of first instruction in basic block.

COMPILE_INST

ARM instruction compiled. Fields:

DISASS **string**

Disassembly of the instruction.

ISSET **enum**

The instruction set of this instruction.

ITSTATE **unsigned int**

The ITSTATE current for the instruction.

OPCODE **unsigned int**

The opcode of the instruction.

PC unsigned int

The address of the instruction.

SIZE unsigned int

The size of the instruction in bytes.

CONTEXTIDR

Context ID Register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

NS bool

Secure or nonsecure banked register is accessed.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The new value written.

CONTEXT_SYNC

Called for every context synchronization event. Fields:

ARCH_CSE bool

Architectural Context Synchronization Event.

PC unsigned int

The address of the synchronization instruction.

SYSREG_SYNC bool

System register synchronization event.

CORE_ENDIAN

Core BE8 Big-Endian state changed. Fields:

BE8 bool

Core BE8 Big-Endian state.

CORE_INFO

Static processor attributes. Only triggered by a call to DumpState(). Fields:

ARCH_PROFILE enum

The architecture profile of the core.

CLUSTER_ID unsigned int

The cluster ID of this processor.

CORE_NUM unsigned int

The number of this core in an MP processor.

DETAILED_CORE_FEATURES enum

Detailed core features as a bitfield. Enum values are bitmasks to extract the info from the field.

FPU_VERSION enum

The VFP version implemented by the core.

MEM_ARCH enum

The memory architecture of the core.

MIDR_EL1 unsigned int

MIDR_EL1 value.

NUM_CORES unsigned int

The number of cores in this MP processor.

QUANTUM_SIZE unsigned int

The default quantum size of the core (in ticks).

SECURITY_FEATURES bool

Does the core have security features?.

CORE_LOADS

Processor load accesses. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ACQREL enum

Is this an acquire/release.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DATA unsigned int

The data read or written.

ELEMENT_SIZE unsigned int

Width of each element.

ENDIANNESS enum

Byte ordering of this access.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

LOCK enum

Normal, exclusive or locked access.

MECID unsigned int

Memory Encryption Context ID, if applicable.

MECID2 unsigned int

Memory Encryption Context ID of the second page, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PADDR_TAG unsigned int

If tag checked, the physical address tag for the access.

PAGE_SIZE unsigned int

Page size as $\log_2(\text{size in bytes})$.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK, 2=Exclusive Failed, 3=Unsupported.

SIZE unsigned int

Width of the access in bytes. Only required if DATA is not traced.

TAG_CHECKED bool

Is this access tag checked.

TAG_CHECKED2 bool

Is the second page access tag checked.

TRANS bool

Is this a translated access.

VADDR unsigned int

The virtual address of the access.

CORE_REGS

Changes of the core registers R0 to R14. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number, 0 to 14.

MODE enum

Bank of the register accessed.

OLD_VALUE unsigned int

The old value overwritten.

PHYS_ID enum

The physical register accessed.

VALUE unsigned int

The new value written to the register.

CORE_REGS64

Changes of the core registers X0..X30, SP_ELn. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ID enum

The register number, 0-30 for X0-X30, >=32 for SP_ELn.

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

CORE_REGS64_READ

General purpose AArch64 register has been read X0..X30, SP_ELn. Fields:

ID enum

The register number, 0-30 for X0-X30, >=32 for SP_ELn.

VALUE unsigned int

The value read from the register.

CORE_STALLED_TRANSITION

Detect when a core is prevented from progressing PChannel states. Fields:

CURRENT_STATE enum

Current power mode of core.

REASON string

The reason a transition was stalled, and what must be done to unblock it.

TARGET_STATE enum

Desired power mode of core after transition.

CORE_STORES

Processor store accesses. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ACQREL enum

Is this an acquire/release.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DATA unsigned int

The data read or written.

ELEMENT_SIZE unsigned int

Width of each element.

ENDIANNESS enum

Byte ordering of this access.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

LOCK enum

Normal, exclusive or locked access.

MECID unsigned int

Memory Encryption Context ID, if applicable.

MECID2 unsigned int

Memory Encryption Context ID of the second page, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PADDR_TAG unsigned int

If tag checked, the physical address tag for the access.

PAGE_SIZE unsigned int

Page size as $\log_2(\text{size in bytes})$.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK, 2=Exclusive Failed, 3=Unsupported.

SIZE unsigned int

Width of the access in bytes. Only required if DATA is not traced.

TAG_CHECKED bool

Is this access tag checked.

TAG_CHECKED2 bool

Is the second page access tag checked.

TRANS bool

Is this a translated access.

VADDR unsigned int

The virtual address of the access.

CP14_READ

System Coprocessor register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP14 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc1 unsigned int

opcode 1.

opc2 unsigned int

opcode 2.

CP14_WRITE

System Coprocessor register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP14 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc1 unsigned int

opcode 1.

opc2 unsigned int

opcode 2.

CP15_READ

System Coprocessor register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP15 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc1 unsigned int

opcode 1.

opc2 unsigned int

opcode 2.

CP15_READ64

System Coprocessor register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP15 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc unsigned int

opcode 1.

CP15_WRITE

System Control Coprocessor register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP15 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc1 unsigned int

opcode 1.

opc2 unsigned int
opcode 2.

CP15_WRITE64

System Coprocessor register write. Fields:

CORE_NUM unsigned int
Core number in a multi processor.

CRm unsigned int
CRm.

NS bool
Secure or nonsecure banked register is accessed.

REG_NAME string
Name of the CP15 register accessed.

SECURITY_STATE enum
The current security state.

UNDEF bool
The register accessed is undefined.

UPDATED_VALUE unsigned int
Updated value of the register now it has been written.

VALUE unsigned int
The new value written.

opc unsigned int
opcode 1.

CPSR

CPSR change. (In AArch64 PSTATE represented in SPSR format). Fields:

CORE_NUM unsigned int
Core number in a multi processor.

OLD_VALUE unsigned int
The old CPSR value.

UNKNOWN unsigned int
Bits within the register that have unknown value.

VALUE unsigned int
The new CPSR value.

CRYPTO_SPEC

Every crypto instruction speculatively executed.

CorePowerStateChange

Core power-up state has changed. DISPLAY Core powered %{PU:(up|down)}. Fields:

PU bool

Whether the core powered up.

DATA_CACHE_SET_ALLOC_TAGS

Set allocation tags from DC GVA instruction. Fields:

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

NSDESC unsigned int

The security state of the access.

PADDR unsigned int

The physical address of the access.

PAS enum

The physical address space of the page.

RESPONSE enum

0=Aborted, 1=OK.

SIZE unsigned int

The size of memory in bytes.

TAG unsigned int

The value of the tag being set.

TAGGING_TYPE enum

Type of Memory tagging (Physical or Virtual).

TAG_VADDR unsigned int

The virtual address of the Virtual Tag if Virtual tagging is used; otherwise 0.

VADDR unsigned int

Virtual address used for setting allocation tags.

DATA_CACHE_ZERO

Zero and invalidate cache line from DC ZVA instruction. Fields:

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

NSDESC unsigned int

The security state of the access.

PADDR unsigned int

The physical address of the access.

PADDR_TAG unsigned int

If tag checked, the physical address tag for the access.

PAS enum

The physical address space of the page.

RESPONSE enum

0=Aborted, 1=OK.

SIZE unsigned int

The size of zero'd memory in bytes.

TAG_CHECKED bool

Is this access tag checked.

VADDR unsigned int

Virtual address of the zero'd cache line.

DEBUG_EVENT

Hardware debug support event. Fields:

EVENT enum

Description of event.

VALUE unsigned int

data value.

DEBUG_TARGET_ISET

Hardware debug target ISA. Fields:

TARGET_ISET enum

The target instruction set of the debug state.

DMI_ALLOCATE

DMI allocated. Fields:

ACCESS_DIRECTION enum

READ / WRITE.

PADDR unsigned int

The physical address of the DMI.

PTR unsigned int

DMI range pointer.

SIDE enum

DSIDE / ISIDE.

SIZE unsigned int

The size of DMI.

VADDR unsigned int

The virtual address of the DMI..

DMI_HIT

DMI hit, access used DMI (only fire for small DMI regions i.e. less than a full page). Fields:

ACCESS_DIRECTION enum

READ / WRITE.

PADDR unsigned int

The physical address of the DMI.

SIDE enum

DSIDE / ISIDE.

SIZE unsigned int

The size of the DMI.

VADDR unsigned int

The virtual address of the DMI..

DMI_REVOKE

DMI revoked. Fields:

ACCESS_DIRECTION enum

READ / WRITE.

PADDR unsigned int

The physical address of the DMI.

SIDE enum

DSIDE / ISIDE.

SIZE unsigned int

The size of DMI.

VADDR unsigned int

The virtual address of the DMI..

END_COMPILE

Compilation end. Fields:

END_OF_PAGE_COUNT unsigned int

Number of basic blocks exited due to page boundary since START_COMPILE.

FETCHFAIL_COUNT unsigned int

Number of basic blocks exited due to ifetch failure START_COMPILE.

INST_COUNT unsigned int

Number of instructions compiled since START_COMPILE.

NONSEQ_COUNT unsigned int

Number of basic blocks exited due to non-sequential instructions since START_COMPILE.

PAGE_STRADDLE_COUNT unsigned int

Number of basic blocks exited due to unaligned instructions crossing page since START_COMPILE.

ESB

Error Synchronization Barrier hit. Fields:

ASYNC bool

ESB is async.

IMPLICIT bool

ESB is implicit.

EXCEPTION

Exception taken. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ESR unsigned int

The value assigned to ESR when exception is taken.

ESR_EC unsigned int

The Value assigned to ESR.EC when exception is taken.

IS_PHYSICAL bool

Physical or Virtual exception.

IS_TRAP bool

Exception coming from a trap.

LR unsigned int

The value assigned to the link register.

PC unsigned int

The location where the exception occurred.

PREFERRED_RETURN unsigned int

The preferred return address for the exception.

TAKEN_LOCALLY bool

Whether exception is taken locally(same exception level).

TARGET_ISET enum

The instruction set of the exception handler code.

TARGET_PC unsigned int

The address the exception branches to.

VECTOR enum

The exception vector.

VECTOR_OFFSET unsigned int

The offset of an exception vector from the base address.

EXCEPTION_END

Every exception completed.

EXCEPTION_INFO

Trace human-readable reason of the current exception for data/instruction aborts, floating point exceptions and unknown exceptions. Fields:

DESCRIPTION string

Reason for current exception.

ESR_EC unsigned int

The value of ESR.EC when exception is taken.

EXCEPTION_RAISE

Every exception raised.

EXCEPTION_RETURN

Branches on leaving exception. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISSET enum

The instruction set of the branch instruction.

IS_COND bool

Indicates if this is a conditional branch.

IS_HINTED bool

Whether the conditional instruction is hinted.

IS_ISB bool

Whether the instruction is ISB.

IS_LINK bool

Indicates if this is a branch with link.

PC unsigned int

The address of the branch instruction.

TARGET_ISET enum

The instruction set after the branch.

TARGET_PC unsigned int

The address the instruction branches to.

EXCEPTION_RETURN_PREBRANCH

Exception return event that must occur before PC and processor state have updated.

EXCEPTION_START

Every exception started.

FIQ_TAKEN

FIQ taken exception.

FP_EXCEPTIONS_SIGNALLED

Trace which FP exceptions were signalled as a part of the FP operations performed by an instruction. Fields:

FP_EXCEPTIONS_SIGNALLED_MASK unsigned int

Indicates which FP exceptions were signalled by an instruction. This uses the same encoding as the cumulative floating-point exception bitfields in FPSR.

FP_STATE

Floating point state. Fields:

FPSR unsigned int

FPSR register value.

MASK unsigned int

The mask indicating updated cumulative bits.

FREQ_CHANGED

Core frequency changed. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The instruction count of this core.

NEW_FREQ signed int

The new frequency of this core (expressed in Hz).

OLD_FREQ signed int

The old frequency of this core (expressed in Hz).

HLT

HLT instruction occurred. Fields:

IMM16 unsigned int

The 16-bit immediate value encoded in HLT instruction.

INST

Every instruction executed. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DEBUG_STATE bool

The instruction is executed in debug state.

DISASS string

Disassembly of instruction.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISET enum

The current instruction set.

ITSTATE unsigned int

The current ITSTATE.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

MODE enum

The mode the core is in.

NS unsigned int

The core's non-secure bit.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

OPCODE unsigned int

The opcode of the instruction.

PADDR unsigned int

The physical address of the instruction.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the instruction.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the page.

PC unsigned int

The address of the instruction.

SECURITY_STATE enum

The core's security state.

SIZE unsigned int

The size of the instruction in bytes.

INST_END

Every instruction completed.

INST_START

Every instruction started. Fields:

DEBUG_STATE **bool**

The instruction is executed in debug state.

ISET **enum**

The current instruction set.

MODE **enum**

The mode the core is in.

NS **enum**

The current Secure State.

PC **unsigned int**

The address of the conditional instruction.

SECURITY_STATE **enum**

The current Security State.

INST_STRADDLE

Instruction straddles boundary. Fields:

DEBUG_STATE **bool**

The instruction is executed in debug state.

ISET **enum**

The current instruction set.

MODE **enum**

The mode the core is in.

NS **enum**

The current Secure State.

PC **unsigned int**

The address of the conditional instruction.

SECURITY_STATE **enum**

The current Security State.

IRQ_TAKEN

IRQ taken exception.

ISB

ISB instruction occurred. Fields:

PC **unsigned int**

The address of the ISB instruction.

LOAD_MULTIPLE_REGS_LIST

List of destination registers (not source) of a multiple registers load instruction. Fields:

REG string

Register to add into the list.

LOCAL_MONITOR

Local monitor activity. Fields:

PADDR unsigned int

Local Monitor Address.

State enum

State of the monitor (Open/Exclusive).

MAPPING_CREATE

A memory mapping has been created. This is an implementation detail of the model and so may change without notice. Fields:

ASID unsigned int

The ASID used during translation, or 0 if no ASID was used.

IS_VALID bool

Whether the mapping is valid or not.

PADDR unsigned int

Base physical address of region, or 0 if not known.

SIDE enum

I-side or D-side.

SIZE unsigned int

Size of the page, or 0 if not known.

VADDR unsigned int

Base virtual address of region.

VMID unsigned int

The VMID used during translation, or 0 if no VMID was used.

MAPPING_DESTROY

A memory mapping has been destroyed. This is an implementation detail of the model and so may change without notice. Fields:

ASID unsigned int

The ASID used during translation.

CAUSE enum

The cause of invalidation.

PADDR unsigned int

Base physical address of page.

SIDE enum

I-side or D-side.

SIZE unsigned int

Size of the page.

VADDR unsigned int

Base virtual address of page.

VMID unsigned int

The VMID used during translation.

MEMMAP_DEBUG_READ

Memory mapped reads to the debug registers. Fields:

ADDR unsigned int

address.

CORE_NUM unsigned int

Core number in a multi processor.

EXT bool

Whether access is from an external device (such as the DAP).

NS enum

Secure state of the access.

REG_NAME string

Name of the debug register accessed.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

MEMMAP_DEBUG_WRITE

Memory mapped writes to the debug registers. Fields:

ADDR unsigned int

address.

CORE_NUM unsigned int

Core number in a multi processor.

EXT bool

Whether access is from an external device (such as the DAP).

NS enum

Secure state of the access.

REG_NAME string

Name of the debug register accessed.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The new value written.

MEMTAG_LOADS

Allocation tag memory read; when attributes and system configuration allow tag accesses.

Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

NUM_GRANULES unsigned int

Number of tags to be Loaded/stored.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK.

TAGGING_TYPE enum

Type of Memory tagging (Physical or Virtual).

TAGS unsigned int

The tags read or written. Tags for subsequent granules would be found at subsequent bytes in this field (One byte for each tag).

TAG_VADDR unsigned int

The virtual address of the Virtual Tag if Virtual tagging is used; otherwise 0.

VADDR unsigned int

The virtual address of the access.

MEMTAG_LOAD_INST

Allocation tag load instruction; regardless of tags being actually accessed. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

NUM_GRANULES unsigned int

Number of tags to be Loaded/stored.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK.

TAGGING_TYPE enum

Type of Memory tagging (Physical or Virtual).

TAGS unsigned int

The tags read or written. Tags for subsequent granules would be found at subsequent bytes in this field (One byte for each tag).

TAG_VADDR unsigned int

The virtual address of the Virtual Tag if Virtual tagging is used; otherwise 0.

VADDR unsigned int

The virtual address of the access.

MENTAG_STORES

Allocation tag memory write; when attributes and system configuration allow tag accesses.

Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

NUM_GRANULES unsigned int

Number of tags to be Loaded/stored.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK.

TAGGING_TYPE enum

Type of Memory tagging (Physical or Virtual).

TAGS unsigned int

The tags read or written. Tags for subsequent granules would be found at subsequent bytes in this field (One byte for each tag).

TAG_VADDR unsigned int

The virtual address of the Virtual Tag if Virtual tagging is used; otherwise 0.

VADDR unsigned int

The virtual address of the access.

MEMTAG_STORE_INST

Allocation tag store instruction; regardless of tags being actually accessed. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

NUM_GRANULES unsigned int

Number of tags to be Loaded/stored.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK.

TAGGING_TYPE enum

Type of Memory tagging (Physical or Virtual).

TAGS unsigned int

The tags read or written. Tags for subsequent granules would be found at subsequent bytes in this field (One byte for each tag).

TAG_VADDR unsigned int

The virtual address of the Virtual Tag if Virtual tagging is used; otherwise 0.

VADDR unsigned int

The virtual address of the access.

MMU_D128_TTB_READ

This event is triggered by reads caused by a D128 translation table walk. Fields:

ABORTED bool

Was the walk successful.

DATA_HIGH unsigned int

The data upper bits read.

DATA_LOW unsigned int

The data lower bits read.

IPA unsigned int

For stage 1, the IPA of the read.

LEVEL unsigned int

Translation table level.

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

OUTADDR unsigned int

The output physical address.

PADDR unsigned int

The physical address of the read.

PASmem enum

Physical Address Space of the accessed memory.

PASreq enum

Physical Address Space of the requested lookup.

REGIME_EL enum

Entry matches in this translation regime.

REQUESTER enum

The requester of this table walk.

SIDE enum

Inst / Data.

STAGE unsigned int

Translation stage.

MMU_D128_TTB_WRITE

This event is triggered by writes caused by a D128 translation table walk. Fields:

ABORTED bool

Was the walk successful.

DATA_HIGH unsigned int

The data upper bits written.

DATA_LOW unsigned int

The data lower bits written.

IPA unsigned int

For stage 1, the IPA of the write.

IS_HACDBS_ACCESS bool

If the access was due to HACDBS_CLEAN.

LEVEL unsigned int

Translation table level.

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

OUTADDR unsigned int

The output physical address.

PADDR unsigned int

The physical address of the write.

PASmem enum

Physical Address Space of the accessed memory.

PASreq enum

Physical Address Space of the requested lookup.

REGIME_EL enum

Entry matches in this translation regime.

SIDE enum

Inst / Data.

STAGE unsigned int

Translation stage.

MMU_TRANS

Address translation information. Fields:

ASID unsigned int

Address space identifier.

CORE_NUM unsigned int

Core number in a multi processor.

Hyp bool

Entry matches in Hyp state only.

INNERCACHE_RA bool

Is the inner cache allocate on read.

INNERCACHE_TRANSIENT bool

Is the inner write-through transient.

INNERCACHE_TYPE enum

Inner Caching scheme (NC/MB/WA).

INNERCACHE_WA bool

Is the inner cache allocate on write.

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MEMTYPE enum

Memory type.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC enum

Is secure side supposed to access secure or nonsecure memory.

OUTERCACHE_RA bool

Is the outer cache allocate on read.

OUTERCACHE_TRANSIENT bool

Is the outer write-through transient.

OUTERCACHE_TYPE enum

Outer Caching scheme (NC/MB/WA).

OUTERCACHE_WA bool

Is the outer cache allocate on write.

PADDR unsigned int

Physical address of the access.

PAGESIZE unsigned int

Page size as $\log_2(\text{size})$.

PAS enum

PAS supposed to be accessed is secure or nonsecure memory or Root/Realm if RME is implemented.

SH enum

Shareability.

SIDE enum

Inst / Data.

STAGE1_PERM unsigned int

Stage 1 permissions, [5:3] Privileged access (XWR) [2:0] User access (XWR).

STAGE2_PERM unsigned int

Stage 2 permission mask [5:3] Privileged access (XWR) [2:0] User access (XWR).

VADDR unsigned int

Virtual address of the access.

VMID unsigned int

Virtual machine identifier.

XS bool

XS Attribute.

nG enum

Flag indicating whether ASID will be matched.

MMU_TTB_READ

This event is triggered by reads caused by a translation table walk. Fields:

ABORTED bool

Was the walk successful.

DATA unsigned int

The data read.

DESC_KIND enum

The encoding scheme of the descriptors used for this walk.

IPA unsigned int

For stage 1, the IPA of the read.

LEVEL unsigned int

Translation table level.

LPAE bool

Is this for an LPAE translation (*DEPRECATED*, use DESC_KIND).

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSmem bool

Non secure memory access.

NSreq bool

Non secure request.

OUTADDR unsigned int

The output physical address.

PADDR unsigned int

The physical address of the read.

PASmem enum

Physical Address Space of the accessed memory.

PASreq enum

Physical Address Space of the requested lookup.

REGIME_EL enum

Entry matches in this translation regime.

REQUESTER enum

The requester of this table walk.

SIDE enum

Inst / Data.

STAGE unsigned int

Translation stage.

MMU_TTB_WRITE

This event is triggered by writes caused by a translation table walk. Fields:

ABORTED bool

Was the walk successful.

DATA unsigned int

The data written.

DESC_KIND enum

The encoding scheme of the descriptors used for this walk.

IPA unsigned int

For stage 1, the IPA of the write.

IS_HACDBS_ACCESS bool

If the access was due to HACDBS_CLEAN.

LEVEL unsigned int

Translation table level.

LPAE bool

Is this for an LPAE translation (*DEPRECATED*, use DESC_KIND).

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSmem bool

Non secure memory access.

NSreq bool

Non secure request.

OUTADDR unsigned int

The output physical address.

PADDR unsigned int

The physical address of the write.

PASmem enum

Physical Address Space of the accessed memory.

PASreq enum

Physical Address Space of the requested lookup.

REGIME_EL enum

Entry matches in this translation regime.

SIDE enum

Inst / Data.

STAGE unsigned int

Translation stage.

MODE_CHANGE

Mode change. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

MODE enum

The new mode.

NON_SECURE enum

The core's new non-secure bit.

OLD_MODE enum

The old mode.

SECURITY_STATE enum

The new security state.

NOT_TAG_CHECKED_INST

MTE Tag not checked for current instruction.

PERIODIC

Called for every quantum. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The instruction count of this CPU.

PC unsigned int

The address of the next instruction to be executed on this CPU.

PMU_COUNTER_OVERFLOW

PMU counter overflow. Fields:

COUNTER_GROUP enum

Counter group.

COUNTER_RANGE enum

Counter range.

EVENT_ID unsigned int

Event Identifier, field is valid only for Event counter group.

INDEX unsigned int

Counter index (as selected by PMSELR), field is valid only for Event/Cycle counter groups.

INTERRUPT bool

Is interrupt enabled on overflow for this counter.

POINTER_AUTH

Called when execution hits a pointer authentication check. Fields:

AUTH_STATE bool

Authentication result.

KEY_USED enum

Pointer authentication key.

POTINDEX_UPDATE

Value of POT index for current pc changed. Fields:

NEW_POTINDEX unsigned int

New POTIndex.

OLD_POTINDEX unsigned int

Old POTIndex.

PREFETCH_MEMORY64

Prefetch from PRFM or PRFUM instructions. Fields:

PRFOP enum

Prefetch hint.

VADDR unsigned int

Virtual address of the location that should be prefetched.

PRELOAD64_REQUEST

Preload requests from PRFM or PRFUM instructions. Fields:

PRFOP enum

Prefetch hint.

VADDR unsigned int

Virtual address of the location that should be prefetched.

PRELOAD_DATA

Data preload from PLD instruction. Fields:

VADDR unsigned int

Virtual address of the data that should be preloaded.

PRELOAD_INST

Instruction preload from PLI instruction. Fields:

VADDR unsigned int

Virtual address of the instruction that should be preloaded.

PRE_CORE_LOAD

Trace just before a core load. Fields:

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

PRE_CORE_STORE

Trace just before a core store. Fields:

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

PRE_TTB_READ

This event is triggered before reads caused by a translation table walk. Fields:

ADDR unsigned int

The physical address of the read.

PSTATE

PSTATE fields update. Fields:

ALLINT unsigned int

All interrupt mask bit.

BTYPE unsigned int

Branch target identification bit.

CORE_NUM unsigned int

Core number in a multi processor.

DAIF unsigned int

Exception mask bits.

DIT unsigned int

Data Independent Timing bit.

EL unsigned int

Current Exception level field.

EXLOCK unsigned int

Exception return state lock bit.

IL unsigned int

Illegal Execution state bit.

NZCV unsigned int

Condition flags.

PACM unsigned int

PACM bit.

PAN unsigned int

Privileged Access Never state bit.

PM unsigned int

PMU exception mask bit.

PPEND unsigned int

PMU exception pending bit.

SM unsigned int

StreamingMode status bit.

SP unsigned int

Stack pointer register selection bit.

SS unsigned int

Software Step bit.

SSBS unsigned int

Speculative Store Bypass Safe bit.

TCO unsigned int

Tag Check Override bit.

TOUCHED unsigned int

Mask of the fields that have been touched since this trace has last been emitted. Note that touched does not necessarily mean the value changed. The index of the field is the bit index in the mask plus the index of this field plus one.

UAO unsigned int

User Access Override bit.

UINJ unsigned int

Undefined Instruction exceptions bit.

ZA unsigned int

ZA storage and ZT0 register enable bit.

nRW unsigned int

Current Execution state bit.

PSTATE_PACTIVE_OPMODE_TRANSITION

Trace pactive and Opmode mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Operating mode after the transition.

OLD_STATE enum

Operating mode before the transition.

PSTATE_PACTIVE_OPMODE_TRANSITION_DENIED

Trace the denied pactive and Opmode mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Operating mode after the transition.

OLD_STATE enum

Operating mode before the transition.

REASON string

Reason for the transition being denied.

PSTATE_PACTIVE_POWER_TRANSITION

Trace pactive and power mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Power mode after the transition.

OLD_STATE enum

Power mode before the transition.

PSTATE_PACTIVE_POWER_TRANSITION_DENIED

Trace the denied pactive and power mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Power mode after the transition.

OLD_STATE enum

Power mode before the transition.

REASON string

Reason for the transition being denied.

PSTATE_PACTIVE_THREAD_OPMODE_TRANSITION

Trace pactive and Opmode mode transitions for threads on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Operating mode after the transition.

OLD_STATE enum

Operating mode before the transition.

QUANTUM_START

Called at the start of every quantum.

RANGE_PRELOAD_REQUEST

Preload requests from RPRFM. Fields:

COUNT unsigned int

Number of blocks of data to access.

LENGTH unsigned int

Number of contiguous bytes to be accessed in each block.

OP unsigned int

Range prefetch operation .

REUSE unsigned int

Maximum number of bytes to be accessed before next RPRFM .

STRIDE signed int

Stride to advance after accessing LENGTH bytes.

VADDR unsigned int

Base virtual address of range prefetch request.

RUN_STATE

Run state transition. Fields:

INST_COUNT unsigned int

Ticks count at point of transition.

NEW enum

New run state.

OLD enum

Old run state.

SEMIHOSTING_CALL

Call of a semihost function occurred. Fields:

PC unsigned int

The program counter after the semihosting call.

SEMIHOSTING_PRECALL

About to call semihost. Fields:

EL2 bool

Use EL2 translation regime.

NS bool

Is Non-Secure.

REG_WIDTH unsigned int

The current register width in bytes.

SIGNAL

External signal state change. Fields:

SIGNAL *enum*

Signal that changed.

STATE *bool*

Signal asserted state.

SOFTWARE_STEP

Return the Debug Software Step state. Fields:

STATE *enum*

Software Step state.

SPSR

SPSR change. Fields:

CORE_NUM *unsigned int*

Core number in a multi processor.

MODE *enum*

Which of the banked SPSR registers is written.

OLD_VALUE *unsigned int*

The old SPSR value.

VALUE *unsigned int*

The new SPSR value.

START_COMPILE

Compilation started. Fields:

VADDR *unsigned int*

Instruction where compilation begins.

STG_MULTIPLE_MICRO_OPS_ADDRESS_RANGE

Address range for MTE stores with multiple uops. Fields:

ADDR_END *unsigned int*

Final address.

ADDR_START *unsigned int*

Starting address.

STORE_MULTIPLE_REGS_ADDRESS_RANGE

Address range for stores with multiple registers. Fields:

ADDR_END *unsigned int*

Final address.

ADDR_START *unsigned int*

Starting address.

SUBOPTIMAL_LDST_RETIRED

suboptimal load/store. Fields:

ACCESS_SIZE unsigned int

Log2 of the access width in bytes used for alignment checking (1: half-word, 2: word ...).

OPTIMAL_ALIGN_SELECTOR unsigned int

Selector to decide the optimal alignment size (1: access width, 2: 4-bytes, 3: 8-bytes, ..., 12: 4-kbytes).

VADDR unsigned int

The virtual address of the access.

SVE_INST_SPEC

SVE operations speculatively executed.

SVE_LD_RETIRED

SVE load instructions architecturally executed.

SVE_LOADS

SVE load accesses. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ACQREL enum

Is this an acquire/release.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DATA unsigned int

The data read or written.

ELEMENT_SIZE unsigned int

Width of each element.

ENDIANNESS enum

Byte ordering of this access.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

LOCK enum

Normal, exclusive or locked access.

MECID unsigned int

Memory Encryption Context ID, if applicable.

MECID2 unsigned int

Memory Encryption Context ID of the second page, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PADDR_TAG unsigned int

If tag checked, the physical address tag for the access.

PAGE_SIZE unsigned int

Page size as $\log_2(\text{size in bytes})$.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK, 2=Exclusive Failed, 3=Unsupported.

SIZE unsigned int

Width of the access in bytes. Only required if DATA is not traced.

TAG_CHECKED bool

Is this access tag checked.

TAG_CHECKED2 bool

Is the second page access tag checked.

TRANS bool

Is this a translated access.

VADDR unsigned int

The virtual address of the access.

SVE_MEM_PREFETCH

Scalable Vector Extension memory prefetch. Fields:

CACHE_LEVEL enum

The preferred cache level.

HINT enum

What the memory is expected to be used for.

SIZE unsigned int

The prefetch size in bytes.

TEMPORAL bool

Whether the data is temporal.

VADDR unsigned int

The virtual address of the prefetch.

SVE_REG_READ

Scalable Vector Extension register read. Fields:

ID enum

The register identifier.

SM bool

Whether the PE is in Streaming Mode.

VALUE unsigned int

The read data.

SVE_REG_UPDATE

Scalable Vector Extension register update. Fields:

ID enum

The register identifier.

SM bool

Whether the PE is in Streaming Mode.

VALUE unsigned int

The written data.

SVE_REG_WRITE

Scalable Vector Extension register write. Fields:

ID enum

The register identifier.

SM bool

Whether the PE is in Streaming Mode.

VALUE unsigned int

The written data.

SVE_STORES

SVE store accesses. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ACQREL enum

Is this an acquire/release.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DATA unsigned int

The data read or written.

ELEMENT_SIZE unsigned int

Width of each element.

ENDIANNESS enum

Byte ordering of this access.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

LOCK enum

Normal, exclusive or locked access.

MECID unsigned int

Memory Encryption Context ID, if applicable.

MECID2 unsigned int

Memory Encryption Context ID of the second page, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PADDR_TAG unsigned int

If tag checked, the physical address tag for the access.

PAGE_SIZE unsigned int

Page size as $\log_2(\text{size in bytes})$.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK, 2=Exclusive Failed, 3=Unsupported.

SIZE unsigned int

Width of the access in bytes. Only required if DATA is not traced.

TAG_CHECKED bool

Is this access tag checked.

TAG_CHECKED2 bool

Is the second page access tag checked.

TRANS bool

Is this a translated access.

VADDR unsigned int

The virtual address of the access.

SVE_ST_RETIRED

SVE store instructions architecturally executed.

SVE_UNALIGNED_LDST_RETIRED

SVE load or store instructions architecturally executed that would cause alignment fault if A bit is set.

SVE_Z_PRED_MEM_ACCESS_INFO

Information about SVE/SME predicated accesses to Z registers (non Gather/Scatter). Fields:

MEM_ELEM_SIZE unsigned int

Width of elements accessed in memory (in bytes).

RnW bool

Whether this is a Read not a Write.

VA unsigned int

Virtual base address computed for accesses.

SYNC

Called for every synchronization, that is every quantum break. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The instruction count of this CPU.

LOCAL_QUANTUM unsigned int

The local quantum of this CPU.

LOCAL_TIME unsigned int

The local time of this CPU.

SYNC_REASON enum

Reason for sync.

SYSCALL

System call instruction executed. Fields:

IMM unsigned int

Immediate value of the system call instruction.

TYPE enum

System call type.

VADDR unsigned int

Instruction that caused the system call.

SYSREG_READ128

System Coprocessor 128 bit register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

REGNUM enum

Internal register number.

REGNUM_PHYS enum

Internal register number physically accessed (e.g. in presense of VHE redirection).

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_READ64

System Coprocessor register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REGNUM enum

Internal register number.

REGNUM_OPERAND enum

Register number corresponding to the instruction operand.

REGNUM_PHYS enum

Internal register number physically accessed (e.g. in presense of VHE redirection).

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_UPDATE128

Triggers when the system updates a register. Fields:

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG enum

Register number.

UNKNOWN unsigned int

Bits of the register which became unknown.

VALUE unsigned int

Value written to the register.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_UPDATE32

Triggers when the system updates a register. Fields:

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG enum

Register number.

UNKNOWN unsigned int

Bits of the register which became unknown.

VALUE unsigned int

Value written to the register.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_UPDATE64

Triggers when the system updates a register. Fields:

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG enum

Register number.

UNKNOWN unsigned int

Bits of the register which became unknown.

VALUE unsigned int

Value written to the register.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_WRITE128

System Coprocessor 128 bit register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

REGNUM enum

Internal register number.

REGNUM_PHYS enum

Internal register number physically accessed (e.g. in presense of VHE redirection).

UNDEF bool

The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_WRITE64

System Coprocessor register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REGNUM enum

Internal register number.

REGNUM_OPERAND enum

Register number corresponding to the instruction operand.

REGNUM_PHYS enum

Internal register number physically accessed (e.g. in presense of VHE redirection).

UNDEF bool

The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

TRBU_BUFFER_WRAP

DISPLAY Trace buffer current write pointer wrapped.

TRBU_TRIGGER_EVENT

DISPLAY Trace buffer Trigger Event.

TRBU_WRITE

DISPLAY PETU trace data has been written to the TRBU memory buffer at `{BUFFER_ADDRESS}` with size `{SIZE}` and data `{DATA}`. Fields:

BUFFER_ADDRESS unsigned int

Virtual Address of start of buffer write.

DATA unsigned int

Data written to the buffer.

SIZE unsigned int

Size of sample written to buffer, in bytes.

UNALIGNED_LDST_RETIRED

Processor unaligned load/store. Fields:

ACCESS_SIZE unsigned int

Log2 of the access width in bytes used for alignment checking (1=>half-word, 2->word ...).

NUMBER_OF_BEATS unsigned int

Number of accesses (beats) in this burst (Total size in bytes is `ACCESS_SIZE * NUMBER_OF_BEATS`).

VADDR unsigned int

The virtual address of the access.

VFP_D_REGS

VFP/NEON D 64 bit register write. Fields:

ALIASING enum

Type of register aliasing used.

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a write to the corresponding bit in the register. For example, 0x1 would indicate a write to VALUE[0].

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

VFP_D_REGS_READ

VFP/NEON D 64 bit register read. Fields:

ID unsigned int

The register number.

VALUE unsigned int

The current value of the register.

VFP_Q_REGS

VFP/NEON Q 128 bit register write. Fields:

ALIASING enum

Type of register aliasing used.

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a write to the corresponding byte in the register. For example, 0x1 would indicate a write to VALUE[7:0].

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

VFP_Q_REGS_READ

VFP/NEON Q 128 bit register read. Fields:

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a read to the corresponding byte in the register. For example, 0x1 would indicate a read to VALUE[7:0].

VALUE unsigned int

The current value of the register.

VFP_SYS_REGS

Writes to the VFP/NEON units system registers. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ID enum

Which VFP system register is written.

MASK unsigned int

The mask indicating updated cumulative bits.

OLD_VALUE unsigned int

The register's old value overwritten.

VALUE unsigned int

The new value written to the VFP system register.

VFP_S_REGS

VFP/NEON S 32 bit register write. Fields:

ALIASING enum

Type of register aliasing used.

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a write to the corresponding bit in the register. For example, 0x1 would indicate a write to VALUE[0].

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

VFP_S_REGS_READ

VFP/NEON S 32 bit register read. Fields:

ID unsigned int

The register number.

VALUE unsigned int

The current value of the register.

VIRTUAL_TAG_ALLOCATED

Virtual tag information when allocated. Fields:

TAG_VADDR **unsigned int**

Virtual tag virtual address.

VADDR **unsigned int**

Data Virtual address.

VIRTUAL_TAG **unsigned int**

Virtual tag value.

WAYPOINT

Signals end of basic block execution. Fields:

CORE_NUM **unsigned int**

Core number in a multi processor.

ISET **enum**

Origin instruction set.

IS_COND **bool**

Indicates if this is a conditional waypoint.

PC **unsigned int**

Origin address (or 0 if unavailable).

TAKEN **bool**

Indicates if this waypoint was taken.

TARGET **unsigned int**

Destination address.

TARGET_ISET **enum**

Destination instruction set.

WFE_END

WFE ended. Fields:

INST_COUNT **unsigned int**

Ticks count when leaving WFE.

WFE_EVENT_REGISTER

WFE event register status: set/clear, reason. Fields:

INST_COUNT **unsigned int**

Ticks count.

REASON **enum**

Reason for set/clear. Only REASON==0 (Cleared by WFE) clears the bit, all other reasons set it.

WFE_IGNORED

WFE ignored. Fields:

EVENT bool

This WFE was ignored because the event register was set.

INST_COUNT unsigned int

Ticks count when ignoring WFE.

TRAPPED bool

This WFE was trapped.

WFE_START

WFE entered. Fields:

INST_COUNT unsigned int

Ticks count when entering WFE.

WFI_END

WFI ended. Fields:

INST_COUNT unsigned int

Ticks count when leaving WFI.

WFI_IGNORED

WFI ignored. Fields:

DISABLED bool

This WFI was ignored because WFI is disabled.

INST_COUNT unsigned int

Ticks count when ignoring WFI.

REASON enum

specifies reason why WFI trace was ignored.

TRAPPED bool

This WFI was trapped.

WFI_START

WFI entered. Fields:

INST_COUNT unsigned int

Ticks count when entering WFI.

WFI_WAKEUP

WFI wakeup. Fields:

INST_COUNT unsigned int

Ticks count when WFI wakeup occurred.

REASON enum

Reason for wakeup.

2.19 ARM_Cortex-A53

This section describes the trace sources.

AA64_ASE_SVE_REGS

Modification of AArch64 Advanced SIMD or SVE Vector Registers (via SIMD or Scalar operations), SVE Predicate Registers and ZT0. Fields:

ID **enum**

The register identifier.

MASK **unsigned int**

Bitmask of the register to signal the modified bits in the VALUE field.

SM **bool**

Whether the PE is in Streaming Mode.

VALUE **unsigned int**

Value of AArch64 vector register.

ASYNC_MEMORY_FAULT

Context ID Register write. Fields:

FAULT **unsigned int**

Fault status present in ESR.ISS encoding for Serror in ARCH64 and present in DFSR encoding for external abort format in ARCH32.

PADDR **unsigned int**

Physical Address (or 0 if unavailable).

VADDR **signed int**

Virtual Address (or 0 if unavailable).

ATOMIC_END_ACCESS

Bus trace access for atomics. Fields:

ACCESS_FAIL **bool**

Memory access failed.

ADDR **unsigned int**

The virtual address of the access.

ATTR **unsigned int**

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

COMPARE_VALUE **unsigned int**

Compare value for CAS.

LOAD_VALUE unsigned int

Loaded values.

NSDESC unsigned int

The security state of the access.

OPERAND_VALUE unsigned int

Operation's operand.

OPERATION enum

Operation type.

PADDR unsigned int

The physical address of the access.

PRIV bool

Is this a privileged access?.

ATOMIC_START_ACCESS

Bus trace access for atomics. Fields:

ADDR unsigned int

The virtual address of the access.

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

COMPARE_VALUE unsigned int

Compare value for CAS.

NSDESC unsigned int

The security state of the access.

OPERAND_VALUE unsigned int

Operation's operand.

OPERATION enum

Operation type.

PADDR unsigned int

The physical address of the access.

PRIV bool

Is this a privileged access?.

ArchMsg.Error.exit_code

DISPLAY exit code: %{EXITCODE} component kind: %{KIND} reason: %{REASON}. Fields:

EXITCODE signed int

Exit code to be returned.

KIND string

Component kind that invoked the exit code trace.

REASON string

Reason why the exit code trace was invoked (optional).

ArchMsg.Warning.IgnoredWarmResetRequest

Event fires when software attempts to request warm reset through write to deprecated DBGPRCR.CWRR.

ArchMsg.Warning.ReportedRevisionRegisterRead

DISPLAY Reported revision and patch do not match that of the cpu. Fields:

ORIGINAL_VALUE unsigned int

The value the register would have based on the modelled revision.

REGISTER string

The name of the register that has ben read.

REPORTED_VALUE unsigned int

The value the register has been overridden to.

ArchMsg.Warning.Unpredictably Indexed PM Event Register

TAGS **UNPREDICTABLE** DISPLAY Event %{IsDirect:(Selector value|Register)} %{SEL} is >= %{N}. Fields:

IsDirect unsigned int

Direct Access rather than indirect.

N unsigned int

Number of Accessible Registers.

SEL unsigned int

Selector.

ArchMsg.Warning.cache_contents_unknown

A cache has been activated without being invalidated and may contain **UNKNOWN** content. DISPLAY %{SIDE:(D|I)}-Cache was enabled in %{REGIME:(SECURE|NON_SECURE|HYP)} regime but was not invalidated since power-on: cache lines could contain **UNKNOWN** content %{SIDE:(Note: D-side caches must invalidate SECURE lines even if the cache is only used by NON_SECURE code in order to ensure that unexpected dirty lines tagged SECURE are not naturally evicted)}. Fields:

REGIME unsigned int

Current translation regime.

SIDE unsigned int

0 d-side cache or unified cache, 1 i-side.

ArchMsg.Warning.dcimvac_matches_watchpoint

TAGS **UNPREDICTABLE** DISPLAY Watchpoints matching an AArch32 DCIMVA are implementation defined.

ArchMsg.Warning.decode_fieldmismatch

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with different fields %{FIELD1} and %{FIELD2}. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_fracbitsnegative

TAGS **UNPREDICTABLE** DISPLAY VCVT Instruction is unpredictable with a negative number of fraction bits.

ArchMsg.Warning.decode_initblock

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable in an IT block.

ArchMsg.Warning.decode_initblocknotlast

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable in an IT block when not the last.

ArchMsg.Warning.decode_invalid_condition

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable when the condition is not AL.

ArchMsg.Warning.decode_invalidfieldcombination

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with the values in %{FIELD1} and %{FIELD2}. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_invalidvalue

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with the value in %{FIELD}. Fields:

FIELD enum

Field name.

ArchMsg.Warning.decode_mem_hint_unallocated

TAGS **UNPREDICTABLE** DISPLAY There is no memory hint allocated to this bit pattern.

ArchMsg.Warning.decode_registermatch

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with %{FIELD1} and %{FIELD2} as the same register. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_registermismatch

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with %{FIELD1} and %{FIELD2} as different registers. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_registeroutofrange

TAGS **UNPREDICTABLE** DISPLAY Floating-point instruction is unpredictable indexing beyond the end of the register bank.

ArchMsg.Warning.decode_sbzsbo

TAGS **UNPREDICTABLE** DISPLAY Reserved bits in the instruction are not canonical.

ArchMsg.Warning.decode_transactiontoobig

TAGS **UNPREDICTABLE** DISPLAY Attempt to perform a memory transaction of more than 128 bytes.

ArchMsg.Warning.decode_unpred_other

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable.

ArchMsg.Warning.decode_unpreduseofpc

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable when %{FIELD} is PC. Fields:

FIELD enum

Field name.

ArchMsg.Warning.decode_unpreduseofr13

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable when %{FIELD} is R13. Fields:

FIELD enum

Field name.

ArchMsg.Warning.decode_unpreduseofr13inreglist

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with R13 in register-list.

ArchMsg.Warning.decode_useofd16orhigher

TAGS **UNPREDICTABLE** DISPLAY Attempt to use register D16 or higher where it is not allowed.

ArchMsg.Warning.decode_writebackandbaseinlist

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with write back when the base register is in the transfer list.

ArchMsg.Warning.decode_zeroregistersinlist

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable when no registers are to be transferred.

ArchMsg.Warning.recursive_branch

An instruction is performing a branch that targets the same instruction. If this was intended then a WFI might be more effective. DISPLAY recursive branch detected at PC=%{PC}.

PRIMARY KEY PC. Fields:

PC unsigned int

Address of instruction causing the branch.

ArchMsg.Warning.recursive_exception

An instruction has generated an exception that targets the same instruction. DISPLAY recursive %{TYPE} exception detected at PC=%{PC} (CPSR=%{CPSR}). PRIMARY KEY PC. Fields:

CPSR unsigned int

Processor state.

PC unsigned int

Address of instruction causing the exception.

TYPE enum

Type of exception.

ArchMsg.Warning.reentrant_vector_catch

TAGS **UNPREDICTABLE** DISPLAY Vector catch debug event on Prefetch or Data abort vector %{ADDR}. Fields:

ADDR unsigned int

vector address.

ArchMsg.Warning.reserved_it_state

TAGS **UNPREDICTABLE** DISPLAY Execution with reserved IT state %{ITSTATE}. Fields:

ITSTATE unsigned int

reserved IT state.

ArchMsg.Warning.tlb_contents_unknown

Translation has been enabled without invalidating the TLB. DISPLAY Translation was enabled in the %{PAS:(Secure|Non-secure|Root|Realm)} EL%{EL:d} translation regime (vmid=%{VMID:x}) without invalidating %{INVALIDITY:(|d-side |i-side |)}entries since power-on: %{INVALIDITY:(|D||I)}TLB could contain **UNKNOWN** content. Fields:

EL unsigned int

Exception level of translation regime.

INVALIDITY unsigned int

Which TLB may be invalid (if TLBs are separate).

NON_SECURE bool

Whether the translation regime is non-secure.

PAS unsigned int

Physical address space of translation regime.

VMID unsigned int

Current VMID, if applicable.

ArchMsg.Warning.unknown_DLR_DSPSR

Exiting debug state with unknown DLR or DSPSR.

ArchMsg.Warning.unknown_ELR_SPSR

Returning from debug exception with unknown ELR or SPSR.

ArchMsg.Warning.unpred_FAR_on_LRCPC3_non_atomic_mem_access

TAGS **UNPREDICTABLE** Unpredictable FAR/FSTATUS on LRCPC3 pair instructions resulting in two separate memory accesses
DISPLAY Unpredictable FAR on LRCPC3 non atomic memory access at %{ADDR1} with FSTATUS %{FSTATUS1} or %{ADDR2} with FSTATUS %{FSTATUS2}. Fields:

ADDR1 unsigned int

Possible FAR address 1.

ADDR2 unsigned int

Possible FAR address 2.

FSTATUS1 unsigned int

Possible fault status value 1.

FSTATUS2 unsigned int

Possible fault status value 2.

ArchMsg.Warning.unpred_async_tag_check_fail_on_sve_speculative_load_do_update_ffr

Unpredictable set to FALSE of the FFR element associated to SVE speculative load with an async tag check fail TAGS **UNPREDICTABLE**.

ArchMsg.Warning.unpredictable_a32_breakpoint

TAGS **UNPREDICTABLE** DISPLAY Unpredictable breakpoint on a A32 instruction at %{ADDR1} Offset %{OFFSET} BAS %{BAS} for a %{IS_ADDRESS_MISMATCH_BREAKPOINT:(Mismatch|Match)} Breakpoint number %{N:d} Breakpoint address %{ADDR2}. Fields:

ADDR1 unsigned int

Address.

ADDR2 unsigned int

Breakpoint address.

BAS unsigned int

BAS field.

IS_ADDRESS_MISMATCH_BREAKPOINT bool

Is a Mismatch breakpoint, not a Match breakpoint.

N unsigned int

Breakpoint number.

OFFSET unsigned int

Always has a value of zero.

ArchMsg.Warning.unpredictable_banked_register_access

TAGS **UNPREDICTABLE** DISPLAY Access to banked register (R=%{R:d} SYSm=%{SYSM:2x}) is unpredictable. Fields:

R unsigned int

R bit.

SYSM unsigned int

SYSm.

ArchMsg.Warning.unpredictable_ccfail_undef

TAGS **UNPREDICTABLE** DISPLAY An instruction %{OPCODE} at VA=%{VADDR} failed its condition codes check but would otherwise have caused a non-data-dependent trap or undefined exception. Behaviour is implementation-defined choice of exception or nop. Fields:

OPCODE unsigned int

opcode of the instruction.

VADDR unsigned int

Virtual address of the instruction being executed.

ArchMsg.Warning.unpredictable_ici_bits

TAGS **UNPREDICTABLE** DISPLAY ICI bits do not apply or are architecturally unpredictable for instruction.

ArchMsg.Warning.unpredictable_t32_breakpoint

TAGS **UNPREDICTABLE** DISPLAY Unpredictable breakpoint on a T32 instruction at %{ADDR1} Offset %{OFFSET} BAS %{BAS} for a %{IS_ADDRESS_MISMATCH_BREAKPOINT:(Mismatch|Match)} Breakpoint number %{N:d} Breakpoint address %{ADDR2}. Fields:

ADDR1 unsigned int

Address.

ADDR2 unsigned int

Breakpoint address.

BAS unsigned int

BAS field.

IS_ADDRESS_MISMATCH_BREAKPOINT bool

Is a Mismatch breakpoint, not a Match breakpoint.

N unsigned int

Breakpoint number.

OFFSET unsigned int

Indicates which halfword of a 32-bit Thumb instruction the breakpoint is set on. 0 for the first halfword or if the instruction is 16 bits wide, or 1 for the second halfword.

ArchMsg.Warning.unpredictable_t32_vector_catch

TAGS **UNPREDICTABLE** DISPLAY Unpredictable Vector catch exception on a T32 instruction at %{ADDR} Offset %{OFFSET}. Fields:

ADDR unsigned int

Address.

OFFSET unsigned int

Indicates which halfword of a 32-bit Thumb instruction the vector catch exception is set on. 0 for the first halfword or if the instruction is 16 bits wide, or 1 for the second halfword.

ArchMsg.Warning.unpredictable_watchpoint_far

TAGS **UNPREDICTABLE** DISPLAY The value for FAR/EDWAR for hitting this watchpoint may be anywhere in the range %{LOWER_BOUND} to %{UPPER_BOUND}. Fields:

LOWER_BOUND unsigned int

the lowest address accessed by the instruction that triggered the watchpoint.

UPPER_BOUND unsigned int

the highest watchpointed address accessed by that instruction.

ArchMsg.Warning.unpredictable_watchpoint_far_isscmp

DISPLAY The value for FAR/EDWAR for hitting this watchpoint may be anywhere in the range %{LOWER_BOUND} to %{UPPER_BOUND}. Fields:

LOWER_BOUND unsigned int

the lowest address accessed by the instruction that triggered the watchpoint.

UPPER_BOUND unsigned int

the highest watchpointed address accessed by that instruction.

ArchMsg.Warning.warning_AdvSIMDExpandImmUnexpectedZero

TAGS **UNPREDICTABLE** DISPLAY AdvSIMDExpandImm may treat this immediate value as **UNPREDICTABLE**.

ArchMsg.Warning.warning_ConditionalSMC

TAGS **UNPREDICTABLE** DISPLAY SMC instruction has **UNPREDICTABLE** effects when conditional and when combined with traps.

ArchMsg.Warning.warning_TLBID_out_of_range

TLBID out of range in TLBI according to the configuration of NIS, NVIS, NOS and NVOS. Fields:

NIS unsigned int

NIS.

NOS unsigned int

NOS.

NVIS unsigned int

NVIS.

NVOS unsigned int

NVOS.

TLBID unsigned int

The TLBID value used in the TLBI instruction.

ArchMsg.Warning.warning_TTL_or_TG_mismatches_current_translation_granule

TTL or TG mismatches current translation granule in TLBI VA/RVA/IPA/RIPA. Fields:

CURRENT_TG unsigned int

Current TG as indicated by TCR_ELx or VTCR_EL2.

TLBI_TG unsigned int

TG (or TTL[3:2]) used in TLBI instruction.

ArchMsg.Warning.warning_access_crosses_page_boundary

TAGS **UNPREDICTABLE** DISPLAY Access crosses page boundary.

ArchMsg.Warning.warning_access_crosses_page_boundary_has_fault

TAGS **UNPREDICTABLE** DISPLAY Access that crosses page boundary has fault(s).

ArchMsg.Warning.warning_access_crosses_page_boundary_has_fault_on_both_pages

TAGS **UNPREDICTABLE** DISPLAY Access that crosses page boundary has a fault on each page.
Fields:

ADDR1 unsigned int

FAR address on first page.

ADDR2 unsigned int

FAR address on second page.

FSTATUS1 unsigned int

Fault status on first page.

FSTATUS2 unsigned int

Fault status on second page.

ArchMsg.Warning.warning_access_crosses_page_boundary_spanning_different_memory

TAGS **UNPREDICTABLE** DISPLAY Access at address %{ADDR} crosses page from MEM_TYPE %{MEMTYPE_PAGE1} to %{MEMTYPE_PAGE2}. Fields:

ADDR unsigned int

address of access crossing page.

MEMTYPE_PAGE1 signed int

First Page's memory type (0-Normal, 1-Non-Normal).

MEMTYPE_PAGE2 signed int

Second Page's memory type (0-Normal, 1-Non-Normal).

ArchMsg.Warning.warning_access_wraps_around_memory

TAGS **UNPREDICTABLE** DISPLAY Access wraps around memory in %{WIDTH:d}bit mode.
Fields:

WIDTH unsigned int

Address width.

ArchMsg.Warning.warning_bcr_linking_status

TAGS **UNPREDICTABLE** DISPLAY DBGBCR.{BT2,BT} combination gives unpredictable behaviour.

ArchMsg.Warning.warning_bcr_mask_reserved

TAGS **UNPREDICTABLE** DISPLAY DBGBCR_ELn.MASK is set to a reserved value.

ArchMsg.Warning.warning_bvr_masked_bit_not_zero

TAGS **UNPREDICTABLE** DISPLAY DBGBVRn_EL1.VA contains a bit masked by DBGBCRn_EL1.MASK that is not zero.

ArchMsg.Warning.warning_ccsidr_unimplemented_level

TAGS **UNPREDICTABLE** DISPLAY CCSIDR read while CSSELR %{CSSELR:x} points to an unimplemented cache level. Fields:

CSSELR unsigned int

current effective value of CSSELR.

ArchMsg.Warning.warning_change_to_ns_when_tge_set

TAGS **UNPREDICTABLE** DISPLAY Attempting to change NS to 1 when HCR.TGE = 1.

ArchMsg.Warning.warning_contiguous_bit_check_abort

TAGS **UNPREDICTABLE** DISPLAY An abort occurred whilst attempting to check TLB entry is contiguous at %{ENTRY_ADDR}. Fields:

ENTRY_ADDR unsigned int

address of conflicting TLB entry.

ArchMsg.Warning.warning_contiguous_bit_error

TAGS **UNPREDICTABLE** DISPLAY TLB entry at %{ENTRY_ADDR} was not contiguous with entry at %{CONTIG_ADDR} Contents are %{ENTRY_DATA} but expected %{CONTIG_DATA}. Fields:

CONTIG_ADDR unsigned int

address of the TLB first read, with which this entry is expected to be contiguous.

CONTIG_DATA unsigned int

expected contents based on the entry at CONTIG_ADDR.

ENTRY_ADDR unsigned int

address of conflicting TLB entry.

ENTRY_DATA unsigned int

contents of conflicting TLB entry.

ArchMsg.Warning.warning_contiguous_bit_error_D128

TAGS **UNPREDICTABLE** DISPLAY TLB entry at %{ENTRY_ADDR} was not contiguous with entry at %{CONTIG_ADDR} Contents are %{ENTRY_DATA_HI} %{ENTRY_DATA_LO} but expected %{CONTIG_DATA}. Fields:

CONTIG_ADDR unsigned int

address of the TLB first read, with which this entry is expected to be contiguous.

CONTIG_DATA unsigned int

expected contents based on the entry at CONTIG_ADDR.

ENTRY_ADDR unsigned int

address of conflicting TLB entry.

ENTRY_DATA_HI unsigned int

Higher bit contents of conflicting TLB entry.

ENTRY_DATA_LO unsigned int

Lower bit contents of conflicting TLB entry.

ArchMsg.Warning.warning_cp10_cp11_mismatch

TAGS **UNPREDICTABLE** DISPLAY Unpredictable configuration of CP10 and CP11 controls at EL
%{EL:d}. Fields:

EL unsigned int

EL owning the control.

ArchMsg.Warning.warning_cp10_cp11_reserved_value

TAGS **UNPREDICTABLE** DISPLAY Unpredictable configuration of CPACR CP10 (%{CP10}) or
CP11 (%{CP11}). Fields:

CP10 unsigned int

CP10 access.

CP11 unsigned int

CP11 access.

ArchMsg.Warning.warning_csselr_level_out_of_range

TAGS **UNPREDICTABLE** DISPLAY CSSELR.Level out of range, written value %{LEVEL} with only
%{IMPLEMENTED:d} levels implemented. Fields:

IMPLEMENTED unsigned int

number of cache levels implemented.

LEVEL unsigned int

written value of CSSELR.Level.

TYPE enum

The type of cache selected by CSSELR.InD and TnD.

ArchMsg.Warning.warning_debug_flow_control_bits_not_obeyed

TAGS **UNPREDICTABLE** DISPLAY Flow control bits for %{REGISTER} were not obeyed, causing
%{ERROR}. Fields:

ERROR enum

type of error resulting.

REGISTER enum

register under consideration.

ArchMsg.Warning.warning_debug_register_access_during_reset

TAGS **UNPREDICTABLE** DISPLAY Unpredictable %{IS_WRITE:(read|write)} access to debug register offset %{OFFSET} during reset. Fields:

IS_WRITE unsigned int

Write Not Read.

OFFSET unsigned int

Register Offset.

ArchMsg.Warning.warning_decode_cps_inconsistent_fields

TAGS **UNPREDICTABLE** DISPLAY The mode setting fields in the CPS instruction are inconsistent.

ArchMsg.Warning.warning_decode_invalid_state

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable in current exception-level/security state.

ArchMsg.Warning.warning_default_cacheable_mmu_on

TAGS **UNPREDICTABLE** DISPLAY Default cacheable is enabled (HCR.DC = 1) while MMU is enabled.

ArchMsg.Warning.warning_default_cacheable_vmmu_off

TAGS **UNPREDICTABLE** DISPLAY Default cacheable is enabled (HCR.DC = 1) while HCR.VM = 0.

ArchMsg.Warning.warning_deprecated_wvr_bit_2_set

TAGS **UNPREDICTABLE** DISPLAY DBGWVR%{N:d}_EL1=%{WVR:h} has bit 2 set, deprecated in ARMv8. Fields:

N unsigned int

index.

WVR unsigned int

DBGWVR.

ArchMsg.Warning.warning_eret_while_software_step_active_pending

Missing ISB between setting MDSCR_EL1.SS and ERET TAGS **UNPREDICTABLE**.

ArchMsg.Warning.warning_exclusive_to_non_normal

TAGS **UNPREDICTABLE** DISPLAY Exclusive Access to Non-Normal Memory %{ADDR}. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_exclusive_to_non_writeback

TAGS **UNPREDICTABLE** DISPLAY Exclusive Access to Non-Writeback-Cacheable Memory %{ADDR}. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_execute_from_device_memory

TAGS **UNPREDICTABLE** DISPLAY An attempt was made to execute from Device memory.

ArchMsg.Warning.warning_illegal_cpsr_mode

TAGS **UNPREDICTABLE** DISPLAY Writing an illegal or unimplement mode ({NEW_MODE:h}) to CPSR was unpredictable. Fields:

NEW_MODE unsigned int

New value of CPSR.M.

ArchMsg.Warning.warning_illegal_srs_mode

TAGS **UNPREDICTABLE** DISPLAY Illegal or **UNPREDICTABLE** mode ({MODE:h}) used for SRS instruction. Fields:

MODE unsigned int

mode for Banked SP.

ArchMsg.Warning.warning_implementation_defined_read_debug_register_in_SCS

Read Access to Debug registers from the processor is **IMPLEMENTATION DEFINED** TAGS **IMP_DEF**. Fields:

PPB_OFFSET unsigned int

debug register offset.

ArchMsg.Warning.warning_implementation_defined_sequential_security_transitions_supported

The behavior of sequential instruction fetches that cross from non-secure to secure memory and contain an SG instruction is **CONSTRAINED UNPREDICTABLE** TAGS **UNPREDICTABLE**.

ArchMsg.Warning.warning_implementation_defined_stack_limit_check_supported

It is **IMPLEMENTATION DEFINED** whether stack pointer limit checking is performed for this instruction TAGS **IMP_DEF**. Fields:

ADDRESS unsigned int

Address of instruction.

OPCODE unsigned int

instruction opcode.

ArchMsg.Warning.warning_implementation_defined_taggednonsharedmemorytype

TAGS **IMP_DEF** DISPLAY Memory attribute has Tagged, non shared memory type; it is **IMPLEMENTATION DEFINED** whether the memory location is treated as Tagged or non-Tagged.

ArchMsg.Warning.warning_implementation_defined_write_debug_register_in_SCS

Write Access to Debug registers from the processor is **IMPLEMENTATION DEFINED** TAGS **IMP_DEF**. Fields:

DATA unsigned int

data attempted to be written.

PPB_OFFSET unsigned int

debug register offset.

ArchMsg.Warning.warning_invalid_tcr_granule

TAGS UNPREDICTABLE DISPLAY TCR.TG{%TG_ID:(0|1)} bits have been set to a granule size {%REQUEST:d}K not implemented - using {%SUBSTITUTE:d}K. Fields:

REQUEST signed int

page size requested (or 0 for reserved).

SUBSTITUTE signed int

best guess available page size.

TG_ID bool

bits TG0 or TG1.

ArchMsg.Warning.warning_load_multiple_user_registers_from_user_mode

TAGS UNPREDICTABLE DISPLAY An LDM(user registers) instruction executed from user mode.

ArchMsg.Warning.warning_mte_in_mem_attr_not_supported

TAGS UNPREDICTABLE DISPLAY Memory attribute {%ATTR} has Tagged memory type but Armv8.5-MTE is not implemented. Fields:

ATTR unsigned int

Memory attribute.

ArchMsg.Warning.warning_reserved_breakpoint_state_match

TAGS UNPREDICTABLE DISPLAY {%IS_BREAKPOINT:(Watchpoint|Breakpoint)} programmed with a reserved combination of HMC, SSC and {%IS_BREAKPOINT:(PAC|PMC)}. Fields:

IS_BREAKPOINT unsigned int

Is a breakpoint, not a watchpoint.

ArchMsg.Warning.warning_shareability

TAGS UNPREDICTABLE DISPLAY Unpredictable: combination of the 1st and 2nd stages of translation is Normal Inner Non-Cacheable, Outer Non-Cacheable.

ArchMsg.Warning.warning_software_step_set_while_enabled

TAGS UNPREDICTABLE DISPLAY MDSCR_EL1.SS set to 1 while software step debug exceptions are enabled.

ArchMsg.Warning.warning_thumb_instruction_wraps_around_memory

TAGS UNPREDICTABLE DISPLAY Thumb instruction wraps around memory.

ArchMsg.Warning.warning_ttbr_sbz_bits_are_not_zero

TAGS UNPREDICTABLE DISPLAY {%TTBR}[{%MSB:d}:0] should be zero, but are not. Fields:

MSB unsigned int

x-1.

TTBR enum

which TTBR register.

ArchMsg.Warning.warning_unaligned_address_dbgdtrrx_write

TAGS **UNPREDICTABLE** DISPLAY Unaligned address %{ADDR} in X0/R0 when DBGDTRRX written in memory access mode. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unaligned_address_dbgdtrtx_read

TAGS **UNPREDICTABLE** DISPLAY Unaligned address %{ADDR} in X0/R0 when DBGDTRTX read in memory access mode. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unaligned_to_device

TAGS **UNPREDICTABLE** DISPLAY Unaligned Access to Device Memory %{ADDR}. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unaligned_to_strongly_ordered

TAGS **UNPREDICTABLE** DISPLAY Unaligned Access to Strongly Ordered Memory %{ADDR}. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unknown_sau_rnr

TAGS **UNPREDICTABLE** DISPLAY SAU_RNR was set to an unsupported value.

ArchMsg.Warning.warning_unpredictable_AIRCR_PRIS_and_BFHFNMINs

TAGS **UNPREDICTABLE** DISPLAY The effect of setting both AIRCR.BFHFNMINs and AIRCR.PRIS to 1 is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_AIRCR_VECTCLRACTIVE_when_not_in_debug

TAGS **UNPREDICTABLE** DISPLAY The effect of writing a 1 to AIRCR.VECTCLRACTIVE if the processor is not halted in Debug state is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_AIRCR_VECTRESET_and_SYSRESETREQ

TAGS **UNPREDICTABLE** DISPLAY When the processor is halted in Debug state, if a write to the register writes a 1 to both VECTRESET and SYSRESETREQ, the behavior is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_AIRCR_VECTRESET_when_not_in_debug

TAGS **UNPREDICTABLE** DISPLAY The effect of writing a 1 to AIRCR.VECTRESET if the processor is not halted in Debug state is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_AIRCR_incorrect_VKEY

TAGS **UNPREDICTABLE** DISPLAY The value 0x05FA must be written to AIRCR.VKEY, otherwise the register write is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_EXC_RETURN_Reserved_Bit

EXC_RETURN[23:7] are reserved with the special condition that all bits should be written as one. Values other than all 1s are **UNPREDICTABLE**. TAGS **UNPREDICTABLE**. Fields:

EXC_RETURN unsigned int

Special PC value.

SBO_MASK unsigned int

Which bits Should Be One to not UNPRED.

ArchMsg.Warning.warning_unpredictable_PMEVTYPEPn_ELO_VS_reserved_value

TAGS **UNPREDICTABLE** DISPLAY Unpredictable reserved value of PMEVTYPEN<n>_ELO.VS.

ArchMsg.Warning.warning_unpredictable_change_to_DEMCR_MON_STEP_at_insufficient_priority

TAGS **UNPREDICTABLE** DISPLAY The effect of changing DEMCR.MON_STEP at an execution priority that is lower than the priority of the DebugMonitor exception is **CONSTRAINED UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_change_to_priority_of_active_exception

TAGS **UNPREDICTABLE** DISPLAY Changing the priority of an active exception is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_clear_lspact

TAGS **UNPREDICTABLE** DISPLAY **UNPREDICTABLE** state on context switch. Fields:

DRVEXC_TAKEN unsigned int

Derived exception taken, not pended.

REACHED_LAST_STORE unsigned int

Only the last faulted.

ArchMsg.Warning.warning_unpredictable_dbg_exit_unaligned_dlr

TAGS **UNPREDICTABLE** DISPLAY Debug exit to AArch32 with DLR[0] set to 1. PC[0] can be set to 0 or to DLR[0].

ArchMsg.Warning.warning_unpredictable_exception_catch

TAGS **UNPREDICTABLE** DISPLAY Generation of an exception catch event is unpredictable.

ArchMsg.Warning.warning_unpredictable_exception_return_inconsistent_state

TAGS **UNPREDICTABLE** DISPLAY state on exception return is unpredictable.

ArchMsg.Warning.warning_unpredictable_exception_return_instruction

TAGS **UNPREDICTABLE** DISPLAY exception return instruction is unpredictable.

ArchMsg.Warning.warning_unpredictable_in_debug_state

TAGS **UNPREDICTABLE** DISPLAY Instruction is **UNPREDICTABLE** when executed in debug state.

ArchMsg.Warning.warning_unpredictable_mair_encoding

TAGS **UNPREDICTABLE** DISPLAY Memory attribute %{ATTR} has invalid MAIR encoding; this may lead to unpredictable results. Fields:

ATTR unsigned int

Memory attribute.

ArchMsg.Warning.warning_unpredictable_pmu_counter_access

TAGS **UNPREDICTABLE** DISPLAY Access to PMU counters from non-secure EL0 or EL1 is **UNPREDICTABLE** with MDCR_EL2.HPMN set to 0.

ArchMsg.Warning.warning_unpredictable_prioritization_breakpoint_match_vector_catch

TAGS **UNPREDICTABLE** DISPLAY Two events with the same priority occurred at the same instruction: Address Matching Vector Catch debug event, and Breakpoint debug event. It is constrained unpredictable which is taken.

ArchMsg.Warning.warning_unpredictable_stack_selection

TAGS **UNPREDICTABLE** DISPLAY **UNPREDICTABLE** state on context switch.

ArchMsg.Warning.warning_unpredictable_tsize_out_of_range

TAGS **UNPREDICTABLE** DISPLAY %{TCR}.T%{TTBR:d}SZ (%{TSIZE:d} bits) is out of range. Must be between 25 and %{TSIZE_MAX:d}. Fields:

TCR enum

Which TCR.

TSIZE signed int

Value of TSize.

TSIZE_MAX signed int

The maximum allowed value.

TTBR unsigned int

Which TTBR region.

ArchMsg.Warning.warning_unpredictable_unaligned_pc_as_base_register

TAGS **UNPREDICTABLE** DISPLAY The PC value must be word-aligned, otherwise the behavior of the instruction is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_unaligned_pop_stack

TAGS **UNPREDICTABLE** DISPLAY **CONSTRAINED UNPREDICTABLE** unaligned stack frame %{FRAME_ADDR:x} in exception-return. Fields:

FRAME_ADDR unsigned int

unaligned stack frame address.

ArchMsg.Warning.warning_unpredictable_unaligned_to_non_writeback

TAGS **UNPREDICTABLE** DISPLAY Unaligned Access to Non-Writeback-Cacheable Memory.

ArchMsg.Warning.warning_unpredictable_vmsa_memattrib

TAGS **UNPREDICTABLE** DISPLAY Unpredictable vmsa memory attribute with texcb value programmed to %{TEXCB} when tex remap is %{TEX_REMAP}. Fields:

TEXCB unsigned int

texcb value.

TEX_REMAP bool

use tex remap.

ArchMsg.Warning.warning_unpredictable_vtcr_t0sz_sl0

TAGS **UNPREDICTABLE** DISPLAY The combination of VTCR.TOSZ=%{TOSZ:d} and VTCR.SLO=%{SLO:d} is unpredictable. Fields:

SLO signed int

VTCR.SLO, starting level for stage 2 translation table walks.

TOSZ signed int

VTCR.TOSZ, required input address range.

ArchMsg.Warning.warning_unpredictable_write_DHCSR

TAGS **UNPREDICTABLE** DISPLAY **UNPREDICTABLE** set of bit-changes in DHCSR. Fields:

NEW unsigned int

data attempted to be written.

OLD unsigned int

previous value.

ArchMsg.Warning.warning_unpredictable_write_SHCSR

TAGS **UNPREDICTABLE** DISPLAY **UNPREDICTABLE** setting a pending bit to 1 for an exception with priority >= execution priority.

ArchMsg.Warning.warning_unsupported_access_to_memory_mapped_register

TAGS **UNPREDICTABLE** DISPLAY Access to memory mapped register is unsupported.

ArchMsg.Warning.warning_user_jmcr_access

TAGS **UNPREDICTABLE** DISPLAY User %{WRITE:(write|read)} access to JMCR is unpredictable. Fields:

WRITE unsigned int

access is write.

ArchMsg.Warning.warning_wcr_mask_and_bas

TAGS **UNPREDICTABLE** DISPLAY DBGWCR_EL1.MASK is non-zero and BAS is not 0xff.

ArchMsg.Warning.warning_wcr_mask_reserved

TAGS **UNPREDICTABLE** DISPLAY DBGWCR_ELn.MASK is set to a reserved value.

ArchMsg.Warning.warning_wcr_non_configuous_bas

TAGS **UNPREDICTABLE** DISPLAY DBGWCR_EL1.BAS has non-contiguous set of ones.

ArchMsg.Warning.warning_wvr_masked_bit_not_zero

TAGS **UNPREDICTABLE** DISPLAY DBGWVRn_EL1.VA contains a bit masked by DBGWCRn_EL1.MASK that is not zero.

ArchMsg.Why.why_branch_target_exception

A branch target exception has occurred. DISPLAY branch target exception: target pc=%{PC}, target opcode=%{OPCODE}. PRIMARY KEY PC. Fields:

BTYPE unsigned int

pstate.BTYPE at time branch taken.

OPCODE unsigned int

opcode at branch target.

PC unsigned int

Virtual Address of branch target.

ArchMsg.Why.why_illegal_state

An illegal mode change has occurred. DISPLAY illegal mode change: %{MESSAGE}. Fields:

MESSAGE string

The message.

BRANCH_MISPREDICT

Simulating branch mispredict. Fields:

PC unsigned int

Origin address (or 0 if unavailable).

BRA_DIR

Direct branches, to immediate address. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISSET enum

The instruction set of the branch instruction.

IS_COND bool

Indicates if this is a conditional branch.

IS_HINTED bool

Whether the conditional instruction is hinted.

IS_ISB bool

Whether the instruction is ISB.

IS_LINK bool

Indicates if this is a branch with link.

PC unsigned int

The address of the branch instruction.

TARGET_ISET enum

The instruction set after the branch.

TARGET_PC unsigned int

The address the instruction branches to.

BRA_INDIR

Indirect branches, perhaps to a register. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISET enum

The instruction set of the branch instruction.

IS_COND bool

Indicates if this is a conditional branch.

IS_HINTED bool

Whether the conditional instruction is hinted.

IS_ISB bool

Whether the instruction is ISB.

IS_LINK bool

Indicates if this is a branch with link.

PC unsigned int

The address of the branch instruction.

TARGET_ISET enum

The instruction set after the branch.

TARGET_PC unsigned int

The address the instruction branches to.

BRA_INDIR_CCFAIL

Indirect branches fail, perhaps to a register.

BRA_RET_CCFAIL

Branch return fail, perhaps to a register.

CACHE_MAINTENANCE_OP

Cache Maintenance Operation. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

DATA unsigned int

Specified MVA or Set/way.

FUNCTION enum

Clean / Invalidate.

SCOPE enum

Affected region.

SIDE enum

Inst / Data.

CCFAIL

Conditional instruction condition check fail. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CCFAIL_UNC

Unconditional instruction condition check fail. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CCPASS

Conditional instruction condition check pass. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CCPASS_UNC

Unconditional instruction condition check pass. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CODE_CACHE_MAINT

Code cache maintenance. Fields:

END_INCL unsigned int

Inclusive end address for VA or PA requests.

NS bool

Non-secure world for PA.

START unsigned int

Start address for VA or PA requests.

TYPE enum

Request type.

COMPILE_BLOCK_END

Last instruction of basic block translated. Fields:

VADDR unsigned int

Address of next instruction after this basic block.

COMPILE_BLOCK_START

First instruction of basic block translated. Fields:

VADDR unsigned int

Address of first instruction in basic block.

COMPILE_INST

ARM instruction compiled. Fields:

DISASS string

Disassembly of the instruction.

ISSET enum

The instruction set of this instruction.

ITSTATE unsigned int

The ITSTATE current for the instruction.

OPCODE unsigned int

The opcode of the instruction.

PC unsigned int

The address of the instruction.

SIZE unsigned int

The size of the instruction in bytes.

CONTEXTIDR

Context ID Register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

NS bool

Secure or nonsecure banked register is accessed.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The new value written.

CONTEXT_SYNC

Called for every context synchronization event. Fields:

ARCH_CSE bool

Architectural Context Synchronization Event.

PC unsigned int

The address of the synchronization instruction.

SYSREG_SYNC bool

System register synchronization event.

CORE_ENDIAN

Core BE8 Big-Endian state changed. Fields:

BE8 bool

Core BE8 Big-Endian state.

CORE_INFO

Static processor attributes. Only triggered by a call to DumpState(). Fields:

ARCH_PROFILE enum

The architecture profile of the core.

CLUSTER_ID unsigned int

The cluster ID of this processor.

CORE_NUM unsigned int

The number of this core in an MP processor.

DETAILED_CORE_FEATURES enum

Detailed core features as a bitfield. Enum values are bitmasks to extract the info from the field.

FPU_VERSION enum

The VFP version implemented by the core.

MEM_ARCH enum

The memory architecture of the core.

MIDR_EL1 unsigned int

MIDR_EL1 value.

NUM_CORES unsigned int

The number of cores in this MP processor.

QUANTUM_SIZE unsigned int

The default quantum size of the core (in ticks).

SECURITY_FEATURES bool

Does the core have security features?.

CORE_LOADS

Processor load accesses. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ACQREL enum

Is this an acquire/release.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DATA unsigned int

The data read or written.

ELEMENT_SIZE unsigned int

Width of each element.

ENDIANNESS enum

Byte ordering of this access.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

LOCK enum

Normal, exclusive or locked access.

MECID unsigned int

Memory Encryption Context ID, if applicable.

MECID2 unsigned int

Memory Encryption Context ID of the second page, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PAGE_SIZE unsigned int

Page size as $\log_2(\text{size in bytes})$.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK, 2=Exclusive Failed, 3=Unsupported.

SIZE unsigned int

Width of the access in bytes. Only required if DATA is not traced.

TRANS bool

Is this a translated access.

VADDR unsigned int

The virtual address of the access.

CORE_REGS

Changes of the core registers R0 to R14. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number, 0 to 14.

MODE enum

Bank of the register accessed.

OLD_VALUE unsigned int

The old value overwritten.

PHYS_ID enum

The physical register accessed.

VALUE unsigned int

The new value written to the register.

CORE_REGS64

Changes of the core registers X0..X30, SP_ELn. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ID enum

The register number, 0-30 for X0-X30, >=32 for SP_ELn.

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

CORE_REGS64_READ

General purpose AArch64 register has been read X0..X30, SP_ELn. Fields:

ID enum

The register number, 0-30 for X0-X30, >=32 for SP_ELn.

VALUE unsigned int

The value read from the register.

CORE_STALLED_TRANSITION

Detect when a core is prevented from progressing PChannel states. Fields:

CURRENT_STATE enum

Current power mode of core.

REASON string

The reason a transition was stalled, and what must be done to unblock it.

TARGET_STATE enum

Desired power mode of core after transition.

CORE_STORES

Processor store accesses. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ACQREL enum

Is this an acquire/release.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DATA unsigned int

The data read or written.

ELEMENT_SIZE unsigned int

Width of each element.

ENDIANNESS enum

Byte ordering of this access.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

LOCK enum

Normal, exclusive or locked access.

MECID unsigned int

Memory Encryption Context ID, if applicable.

MECID2 unsigned int

Memory Encryption Context ID of the second page, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PAGE_SIZE unsigned int

Page size as $\log_2(\text{size in bytes})$.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK, 2=Exclusive Failed, 3=Unsupported.

SIZE unsigned int

Width of the access in bytes. Only required if DATA is not traced.

TRANS bool

Is this a translated access.

VADDR unsigned int

The virtual address of the access.

CP14_READ

System Coprocessor register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP14 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc1 unsigned int

opcode 1.

opc2 unsigned int

opcode 2.

CP14_WRITE

System Coprocessor register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP14 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc1 unsigned int

opcode 1.

opc2 unsigned int

opcode 2.

CP15_READ

System Coprocessor register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP15 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc1 unsigned int

opcode 1.

opc2 unsigned int

opcode 2.

CP15_READ64

System Coprocessor register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP15 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc unsigned int

opcode 1.

CP15_WRITE

System Control Coprocessor register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP15 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc1 unsigned int

opcode 1.

opc2 unsigned int

opcode 2.

CP15_WRITE64

System Coprocessor register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP15 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc unsigned int

opcode 1.

CPSR

CPSR change. (In AArch64 PSTATE represented in SPSR format). Fields:

CORE_NUM unsigned int

Core number in a multi processor.

OLD_VALUE unsigned int

The old CPSR value.

UNKNOWN unsigned int

Bits within the register that have unknown value.

VALUE unsigned int

The new CPSR value.

CRYPTO_SPEC

Every crypto instruction speculatively executed.

CorePowerStateChange

Core power-up state has changed. DISPLAY Core powered %{PU:(up|down)}. Fields:

PU bool

Whether the core powered up.

DATA_CACHE_SET_ALLOC_TAGS

Set allocation tags from DC GVA instruction. Fields:

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

NSDESC unsigned int

The security state of the access.

PADDR unsigned int

The physical address of the access.

PAS enum

The physical address space of the page.

RESPONSE enum

0=Aborted, 1=OK.

SIZE unsigned int

The size of memory in bytes.

TAG unsigned int

The value of the tag being set.

TAGGING_TYPE enum

Type of Memory tagging (Physical or Virtual).

TAG_VADDR unsigned int

The virtual address of the Virtual Tag if Virtual tagging is used; otherwise 0.

VADDR unsigned int

Virtual address used for setting allocation tags.

DATA_CACHE_ZERO

Zero and invalidate cache line from DC ZVA instruction. Fields:

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

NSDESC unsigned int

The security state of the access.

PADDR unsigned int

The physical address of the access.

PADDR_TAG unsigned int

If tag checked, the physical address tag for the access.

PAS enum

The physical address space of the page.

RESPONSE enum

0=Aborted, 1=OK.

SIZE unsigned int

The size of zero'd memory in bytes.

TAG_CHECKED bool

Is this access tag checked.

VADDR unsigned int

Virtual address of the zero'd cache line.

DEBUG_EVENT

Hardware debug support event. Fields:

EVENT enum

Description of event.

VALUE unsigned int

data value.

DEBUG_TARGET_ISET

Hardware debug target ISA. Fields:

TARGET_ISET enum

The target instruction set of the debug state.

DMI_ALLOCATE

DMI allocated. Fields:

ACCESS_DIRECTION enum

READ / WRITE.

PADDR unsigned int

The physical address of the DMI.

PTR unsigned int

DMI range pointer.

SIDE enum

DSIDE / ISIDE.

SIZE unsigned int

The size of DMI.

VADDR unsigned int

The virtual address of the DMI..

DMI_HIT

DMI hit, access used DMI (only fire for small DMI regions i.e. less than a full page). Fields:

ACCESS_DIRECTION enum

READ / WRITE.

PADDR unsigned int

The physical address of the DMI.

SIDE enum

DSIDE / ISIDE.

SIZE unsigned int

The size of the DMI.

VADDR unsigned int

The virtual address of the DMI..

DMI_REVOKE

DMI revoked. Fields:

ACCESS_DIRECTION enum

READ / WRITE.

PADDR unsigned int

The physical address of the DMI.

SIDE enum

DSIDE / ISIDE.

SIZE unsigned int

The size of DMI.

VADDR unsigned int

The virtual address of the DMI..

END_COMPILE

Compilation end. Fields:

END_OF_PAGE_COUNT unsigned int

Number of basic blocks exited due to page boundary since START_COMPILE.

FETCHFAIL_COUNT unsigned int

Number of basic blocks exited due to ifetch failure START_COMPILE.

INST_COUNT unsigned int

Number of instructions compiled since START_COMPILE.

NONSEQ_COUNT unsigned int

Number of basic blocks exited due to non-sequential instructions since START_COMPILE.

PAGE_STRADDLE_COUNT unsigned int

Number of basic blocks exited due to unaligned instructions crossing page since START_COMPILE.

ESB

Error Synchronization Barrier hit. Fields:

ASYNC bool

ESB is async.

IMPLICIT bool

ESB is implicit.

EXCEPTION

Exception taken. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ESR unsigned int

The value assigned to ESR when exception is taken.

ESR_EC unsigned int

The Value assigned to ESR.EC when exception is taken.

IS_PHYSICAL bool

Physical or Virtual exception.

IS_TRAP bool

Exception coming from a trap.

LR unsigned int

The value assigned to the link register.

PC unsigned int

The location where the exception occurred.

PREFERRED_RETURN unsigned int

The preferred return address for the exception.

TAKEN_LOCALLY bool

Whether exception is taken locally(same exception level).

TARGET_ISET enum

The instruction set of the exception handler code.

TARGET_PC unsigned int

The address the exception branches to.

VECTOR enum

The exception vector.

VECTOR_OFFSET unsigned int

The offset of an exception vector from the base address.

EXCEPTION_END

Every exception completed.

EXCEPTION_INFO

Trace human-readable reason of the current exception for data/instruction aborts, floating point exceptions and unknown exceptions. Fields:

DESCRIPTION string

Reason for current exception.

ESR_EC unsigned int

The value of ESR.EC when exception is taken.

EXCEPTION_RAISE

Every exception raised.

EXCEPTION_RETURN

Branches on leaving exception. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISET enum

The instruction set of the branch instruction.

IS_COND bool

Indicates if this is a conditional branch.

IS_HINTED bool

Whether the conditional instruction is hinted.

IS_ISB bool

Whether the instruction is ISB.

IS_LINK bool

Indicates if this is a branch with link.

PC unsigned int

The address of the branch instruction.

TARGET_ISET enum

The instruction set after the branch.

TARGET_PC unsigned int

The address the instruction branches to.

EXCEPTION_RETURN_PREBRANCH

Exception return event that must occur before PC and processor state have updated.

EXCEPTION_START

Every exception started.

FIQ_TAKEN

FIQ taken exception.

FP_EXCEPTIONS_SIGNALLED

Trace which FP exceptions were signalled as a part of the FP operations performed by an instruction. Fields:

FP_EXCEPTIONS_SIGNALLED_MASK unsigned int

Indicates which FP exceptions were signalled by an instruction. This uses the same encoding as the cumulative floating-point exception bitfields in FPSR.

FP_STATE

Floating point state. Fields:

FPSR unsigned int

FPSR register value.

MASK unsigned int

The mask indicating updated cumulative bits.

FREQ_CHANGED

Core frequency changed. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The instruction count of this core.

NEW_FREQ signed int

The new frequency of this core (expressed in Hz).

OLD_FREQ signed int

The old frequency of this core (expressed in Hz).

HLT

HLT instruction occurred. Fields:

IMM16 unsigned int

The 16-bit immediate value encoded in HLT instruction.

INST

Every instruction executed. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DEBUG_STATE bool

The instruction is executed in debug state.

DISASS string

Disassembly of instruction.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISSET enum

The current instruction set.

ITSTATE unsigned int

The current ITSTATE.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

MODE enum

The mode the core is in.

NS unsigned int

The core's non-secure bit.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

OPCODE unsigned int

The opcode of the instruction.

PADDR unsigned int

The physical address of the instruction.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the instruction.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the page.

PC unsigned int

The address of the instruction.

SECURITY_STATE enum

The core's security state.

SIZE unsigned int

The size of the instruction in bytes.

INST_END

Every instruction completed.

INST_START

Every instruction started. Fields:

DEBUG_STATE bool

The instruction is executed in debug state.

ISET enum

The current instruction set.

MODE enum

The mode the core is in.

NS enum

The current Secure State.

PC unsigned int

The address of the conditional instruction.

SECURITY_STATE enum

The current Security State.

INST_STRADDLE

Instruction straddles boundary. Fields:

DEBUG_STATE bool

The instruction is executed in debug state.

ISET enum

The current instruction set.

MODE enum

The mode the core is in.

NS enum

The current Secure State.

PC unsigned int

The address of the conditional instruction.

SECURITY_STATE enum

The current Security State.

IRQ_TAKEN

IRQ taken exception.

ISB

ISB instruction occurred. Fields:

PC unsigned int

The address of the ISB instruction.

LOAD_MULTIPLE_REGS_LIST

List of destination registers (not source) of a multiple registers load instruction. Fields:

REG string

Register to add into the list.

LOCAL_MONITOR

Local monitor activity. Fields:

PADDR unsigned int

Local Monitor Address.

State enum

State of the monitor (Open/Exclusive).

MAPPING_CREATE

A memory mapping has been created. This is an implementation detail of the model and so may change without notice. Fields:

ASID unsigned int

The ASID used during translation, or 0 if no ASID was used.

IS_VALID bool

Whether the mapping is valid or not.

PADDR unsigned int

Base physical address of region, or 0 if not known.

SIDE enum

I-side or D-side.

SIZE unsigned int

Size of the page, or 0 if not known.

VADDR unsigned int

Base virtual address of region.

VMID unsigned int

The VMID used during translation, or 0 if no VMID was used.

MAPPING_DESTROY

A memory mapping has been destroyed. This is an implementation detail of the model and so may change without notice. Fields:

ASID unsigned int

The ASID used during translation.

CAUSE enum

The cause of invalidation.

PADDR unsigned int

Base physical address of page.

SIDE enum

I-side or D-side.

SIZE unsigned int

Size of the page.

VADDR unsigned int

Base virtual address of page.

VMID unsigned int

The VMID used during translation.

MEMMAP_DEBUG_READ

Memory mapped reads to the debug registers. Fields:

ADDR unsigned int

address.

CORE_NUM unsigned int

Core number in a multi processor.

EXT bool

Whether access is from an external device (such as the DAP).

NS enum

Secure state of the access.

REG_NAME string

Name of the debug register accessed.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

MEMMAP_DEBUG_WRITE

Memory mapped writes to the debug registers. Fields:

ADDR **unsigned int**

address.

CORE_NUM **unsigned int**

Core number in a multi processor.

EXT **bool**

Whether access is from an external device (such as the DAP).

NS **enum**

Secure state of the access.

REG_NAME **string**

Name of the debug register accessed.

UNDEF **bool**

The register accessed is undefined.

VALUE **unsigned int**

The new value written.

MMU_D128_TTB_READ

This event is triggered by reads caused by a D128 translation table walk. Fields:

ABORTED **bool**

Was the walk successful.

DATA_HIGH **unsigned int**

The data upper bits read.

DATA_LOW **unsigned int**

The data lower bits read.

IPA **unsigned int**

For stage 1, the IPA of the read.

LEVEL **unsigned int**

Translation table level.

MECID **unsigned int**

Memory Encryption Context(MEC) ID, if applicable.

MPAM_PARTID **unsigned int**

MPAM Partition ID.

MPAM_PMG **unsigned int**

MPAM Performance Monitoring Group.

MPAM_SP **unsigned int**

MPAM Partition ID Space.

OUTADDR unsigned int

The output physical address.

PADDR unsigned int

The physical address of the read.

PASmem enum

Physical Address Space of the accessed memory.

PASreq enum

Physical Address Space of the requested lookup.

REGIME_EL enum

Entry matches in this translation regime.

REQUESTER enum

The requester of this table walk.

SIDE enum

Inst / Data.

STAGE unsigned int

Translation stage.

MMU_D128_TTB_WRITE

This event is triggered by writes caused by a D128 translation table walk. Fields:

ABORTED bool

Was the walk successful.

DATA_HIGH unsigned int

The data upper bits written.

DATA_LOW unsigned int

The data lower bits written.

IPA unsigned int

For stage 1, the IPA of the write.

IS_HACDBS_ACCESS bool

If the access was due to HACDBS_CLEAN.

LEVEL unsigned int

Translation table level.

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

OUTADDR unsigned int

The output physical address.

PADDR unsigned int

The physical address of the write.

PASmem enum

Physical Address Space of the accessed memory.

PASreq enum

Physical Address Space of the requested lookup.

REGIME_EL enum

Entry matches in this translation regime.

SIDE enum

Inst / Data.

STAGE unsigned int

Translation stage.

MMU_TRANS

Address translation information. Fields:

ASID unsigned int

Address space identifier.

CORE_NUM unsigned int

Core number in a multi processor.

Hyp bool

Entry matches in Hyp state only.

INNERCACHE_RA bool

Is the inner cache allocate on read.

INNERCACHE_TRANSIENT bool

Is the inner write-through transient.

INNERCACHE_TYPE enum

Inner Caching scheme (NC/MB/WA).

INNERCACHE_WA bool

Is the inner cache allocate on write.

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MEMTYPE enum

Memory type.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC enum

Is secure side supposed to access secure or nonsecure memory.

OUTERCACHE_RA bool

Is the outer cache allocate on read.

OUTERCACHE_TRANSIENT bool

Is the outer write-through transient.

OUTERCACHE_TYPE enum

Outer Caching scheme (NC/MB/WA).

OUTERCACHE_WA bool

Is the outer cache allocate on write.

PADDR unsigned int

Physical address of the access.

PAGESIZE unsigned int

Page size as log2(size).

PAS enum

PAS supposed to be accessed is secure or nonsecure memory or Root/Realm if RME is implemented.

SH enum

Shareability.

SIDE enum

Inst / Data.

STAGE1_PERM unsigned int

Stage 1 permissions, [5:3] Privileged access (XWR) [2:0] User access (XWR).

STAGE2_PERM unsigned int

Stage 2 permission mask [5:3] Privileged access (XWR) [2:0] User access (XWR).

VADDR unsigned int

Virtual address of the access.

VMID unsigned int

Virtual machine identifier.

XS bool

XS Attribute.

nG enum

Flag indicating whether ASID will be matched.

MMU_TTB_READ

This event is triggered by reads caused by a translation table walk. Fields:

ABORTED bool

Was the walk successful.

DATA unsigned int

The data read.

DESC_KIND enum

The encoding scheme of the descriptors used for this walk.

IPA unsigned int

For stage 1, the IPA of the read.

LEVEL unsigned int

Translation table level.

LPAE bool

Is this for an LPAE translation (*DEPRECATED*, use DESC_KIND).

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSmem bool

Non secure memory access.

NSreq bool

Non secure request.

OUTADDR unsigned int

The output physical address.

PADDR unsigned int

The physical address of the read.

PASmem enum

Physical Address Space of the accessed memory.

PASreq enum

Physical Address Space of the requested lookup.

REGIME_EL enum

Entry matches in this translation regime.

REQUESTER enum

The requester of this table walk.

SIDE enum

Inst / Data.

STAGE unsigned int

Translation stage.

MMU_TTB_WRITE

This event is triggered by writes caused by a translation table walk. Fields:

ABORTED bool

Was the walk successful.

DATA unsigned int

The data written.

DESC_KIND enum

The encoding scheme of the descriptors used for this walk.

IPA unsigned int

For stage 1, the IPA of the write.

IS_HACDBS_ACCESS bool

If the access was due to HACDBS_CLEAN.

LEVEL unsigned int

Translation table level.

LPAE bool

Is this for an LPAE translation (*DEPRECATED*, use DESC_KIND).

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSmem bool

Non secure memory access.

NSreq bool

Non secure request.

OUTADDR **unsigned int**

The output physical address.

PADDR **unsigned int**

The physical address of the write.

PASmem **enum**

Physical Address Space of the accessed memory.

PASreq **enum**

Physical Address Space of the requested lookup.

REGIME_EL **enum**

Entry matches in this translation regime.

SIDE **enum**

Inst / Data.

STAGE **unsigned int**

Translation stage.

MODE_CHANGE

Mode change. Fields:

CORE_NUM **unsigned int**

Core number in a multi processor.

MODE **enum**

The new mode.

NON_SECURE **enum**

The core's new non-secure bit.

OLD_MODE **enum**

The old mode.

SECURITY_STATE **enum**

The new security state.

NOT_TAG_CHECKED_INST

MTE Tag not checked for current instruction.

PERIODIC

Called for every quantum. Fields:

CORE_NUM **unsigned int**

Core number in a multi processor.

INST_COUNT **unsigned int**

The instruction count of this CPU.

PC **unsigned int**

The address of the next instruction to be executed on this CPU.

PMU_COUNTER_OVERFLOW

PMU counter overflow. Fields:

COUNTER_GROUP enum

Counter group.

COUNTER_RANGE enum

Counter range.

EVENT_ID unsigned int

Event Identifier, field is valid only for Event counter group.

INDEX unsigned int

Counter index (as selected by PMSELR), field is valid only for Event/Cycle counter groups.

INTERRUPT bool

Is interrupt enabled on overflow for this counter.

POTINDEX_UPDATE

Value of POT index for current pc changed. Fields:

NEW_POTINDEX unsigned int

New POTIndex.

OLD_POTINDEX unsigned int

Old POTIndex.

PREFETCH_MEMORY64

Prefetch from PRFM or PRFUM instructions. Fields:

PRFOP enum

Prefetch hint.

VADDR unsigned int

Virtual address of the location that should be prefetched.

PRELOAD64_REQUEST

Preload requests from PRFM or PRFUM instructions. Fields:

PRFOP enum

Prefetch hint.

VADDR unsigned int

Virtual address of the location that should be prefetched.

PRELOAD_DATA

Data preload from PLD instruction. Fields:

VADDR unsigned int

Virtual address of the data that should be preloaded.

PRELOAD_INST

Instruction preload from PLI instruction. Fields:

VADDR unsigned int

Virtual address of the instruction that should be preloaded.

PRE_CORE_LOAD

Trace just before a core load. Fields:

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

PRE_CORE_STORE

Trace just before a core store. Fields:

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

PRE_TTB_READ

This event is triggered before reads caused by a translation table walk. Fields:

ADDR unsigned int

The physical address of the read.

PSTATE

PSTATE fields update. Fields:

ALLINT unsigned int

All interrupt mask bit.

BTYPE unsigned int

Branch target identification bit.

CORE_NUM unsigned int

Core number in a multi processor.

DAIF unsigned int

Exception mask bits.

DIT unsigned int

Data Independent Timing bit.

EL unsigned int

Current Exception level field.

EXLOCK unsigned int

Exception return state lock bit.

IL unsigned int

Illegal Execution state bit.

NZCV unsigned int

Condition flags.

PACM unsigned int

PACM bit.

PAN unsigned int

Privileged Access Never state bit.

PM unsigned int

PMU exception mask bit.

PPEND unsigned int

PMU exception pending bit.

SM unsigned int

StreamingMode status bit.

SP unsigned int

Stack pointer register selection bit.

SS unsigned int

Software Step bit.

SSBS unsigned int

Speculative Store Bypass Safe bit.

TCO unsigned int

Tag Check Override bit.

TOUCHED unsigned int

Mask of the fields that have been touched since this trace has last been emitted. Note that touched does not necessarily mean the value changed. The index of the field is the bit index in the mask plus the index of this field plus one.

UAO unsigned int

User Access Override bit.

UINJ unsigned int

Undefined Instruction exceptions bit.

ZA unsigned int

ZA storage and ZT0 register enable bit.

nRW unsigned int

Current Execution state bit.

PSTATE_PACTIVE_OPMODE_TRANSITION

Trace pactive and Opmode mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Operating mode after the transition.

OLD_STATE enum

Operating mode before the transition.

PSTATE_PACTIVE_OPMODE_TRANSITION_DENIED

Trace the denied pactive and Opmode mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Operating mode after the transition.

OLD_STATE enum

Operating mode before the transition.

REASON string

Reason for the transition being denied.

PSTATE_PACTIVE_POWER_TRANSITION

Trace pactive and power mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Power mode after the transition.

OLD_STATE enum

Power mode before the transition.

PSTATE_PACTIVE_POWER_TRANSITION_DENIED

Trace the denied pactive and power mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Power mode after the transition.

OLD_STATE enum

Power mode before the transition.

REASON string

Reason for the transition being denied.

PSTATE_PACTIVE_THREAD_OPMODE_TRANSITION

Trace pactive and Opmode mode transitions for threads on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Operating mode after the transition.

OLD_STATE enum

Operating mode before the transition.

QUANTUM_START

Called at the start of every quantum.

RANGE_PRELOAD_REQUEST

Preload requests from RPRFM. Fields:

COUNT unsigned int

Number of blocks of data to access.

LENGTH unsigned int

Number of contiguous bytes to be accessed in each block.

OP unsigned int

Range prefetch operation .

REUSE unsigned int

Maximum number of bytes to be accessed before next RPRFM .

STRIDE signed int

Stride to advance after accessing LENGTH bytes.

VADDR unsigned int

Base virtual address of range prefetch request.

RUN_STATE

Run state transition. Fields:

INST_COUNT unsigned int

Ticks count at point of transition.

NEW enum

New run state.

OLD enum

Old run state.

SEMIHOSTING_CALL

Call of a semihost function occurred. Fields:

PC unsigned int

The program counter after the semihosting call.

SEMIHOSTING_PRECALL

About to call semihost. Fields:

EL2 bool

Use EL2 translation regime.

NS bool

Is Non-Secure.

REG_WIDTH unsigned int

The current register width in bytes.

SIGNAL

External signal state change. Fields:

SIGNAL enum

Signal that changed.

STATE bool

Signal asserted state.

SOFTWARE_STEP

Return the Debug Software Step state. Fields:

STATE enum

Software Step state.

SPSR

SPSR change. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

MODE enum

Which of the banked SPSR registers is written.

OLD_VALUE unsigned int

The old SPSR value.

VALUE unsigned int

The new SPSR value.

START_COMPILE

Compilation started. Fields:

VADDR unsigned int

Instruction where compilation begins.

STG_MULTIPLE_MICRO_OPS_ADDRESS_RANGE

Address range for MTE stores with multiple uops. Fields:

ADDR_END unsigned int

Final address.

ADDR_START unsigned int

Starting address.

STORE_MULTIPLE_REGS_ADDRESS_RANGE

Address range for stores with multiple registers. Fields:

ADDR_END unsigned int

Final address.

ADDR_START unsigned int

Starting address.

SUBOPTIMAL_LDST_RETIRED

suboptimal load/store. Fields:

ACCESS_SIZE unsigned int

Log2 of the access width in bytes used for alignment checking (1: half-word, 2: word ...).

OPTIMAL_ALIGN_SELECTOR unsigned int

Selector to decide the optimal alignment size (1: access width, 2: 4-bytes, 3: 8-bytes, ..., 12: 4-kbytes).

VADDR unsigned int

The virtual address of the access.

SYNC

Called for every synchronization, that is every quantum break. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The instruction count of this CPU.

LOCAL_QUANTUM unsigned int

The local quantum of this CPU.

LOCAL_TIME unsigned int

The local time of this CPU.

SYNC_REASON enum

Reason for sync.

SYSCALL

System call instruction executed. Fields:

IMM unsigned int

Immediate value of the system call instruction.

TYPE enum

System call type.

VADDR unsigned int

Instruction that caused the system call.

SYSREG_READ128

System Coprocessor 128 bit register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

REGNUM enum

Internal register number.

REGNUM_PHYS enum

Internal register number physically accessed (e.g. in presense of VHE redirection).

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_READ64

System Coprocessor register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REGNUM enum

Internal register number.

REGNUM_OPERAND enum

Register number corresponding to the instruction operand.

REGNUM_PHYS enum

Internal register number physically accessed (e.g. in presense of VHE redirection).

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_UPDATE128

Triggers when the system updates a register. Fields:

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG enum

Register number.

UNKNOWN unsigned int

Bits of the register which became unknown.

VALUE unsigned int

Value written to the register.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_UPDATE32

Triggers when the system updates a register. Fields:

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG enum

Register number.

UNKNOWN unsigned int

Bits of the register which became unknown.

VALUE unsigned int

Value written to the register.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_UPDATE64

Triggers when the system updates a register. Fields:

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG enum

Register number.

UNKNOWN unsigned int

Bits of the register which became unknown.

VALUE unsigned int

Value written to the register.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_WRITE128

System Coprocessor 128 bit register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

REGNUM enum

Internal register number.

REGNUM_PHYS enum

Internal register number physically accessed (e.g. in presense of VHE redirection).

UNDEF bool

The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_WRITE64

System Coprocessor register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REGNUM enum

Internal register number.

REGNUM_OPERAND enum

Register number corresponding to the instruction operand.

REGNUM_PHYS enum

Internal register number physically accessed (e.g. in presense of VHE redirection).

UNDEF bool

The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

UNALIGNED_LDST_RETIRED

Processor unaligned load/store. Fields:

ACCESS_SIZE unsigned int

Log2 of the access width in bytes used for alignment checking (1=>half-word, 2->word ...).

NUMBER_OF_BEATS unsigned int

Number of accesses (beats) in this burst (Total size in bytes is ACCESS_SIZE * NUMBER_OF_BEATS).

VADDR unsigned int

The virtual address of the access.

VFP_D_REGS

VFP/NEON D 64 bit register write. Fields:

ALIASING enum

Type of register aliasing used.

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a write to the corresponding bit in the register. For example, 0x1 would indicate a write to VALUE[0].

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

VFP_D_REGS_READ

VFP/NEON D 64 bit register read. Fields:

ID unsigned int

The register number.

VALUE unsigned int

The current value of the register.

VFP_Q_REGS

VFP/NEON Q 128 bit register write. Fields:

ALIASING enum

Type of register aliasing used.

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a write to the corresponding byte in the register. For example, 0x1 would indicate a write to VALUE[7:0].

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

VFP_Q_REGS_READ

VFP/NEON Q 128 bit register read. Fields:

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a read to the corresponding byte in the register. For example, 0x1 would indicate a read to VALUE[7:0].

VALUE unsigned int

The current value of the register.

VFP_SYS_REGS

Writes to the VFP/NEON units system registers. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ID enum

Which VFP system register is written.

MASK unsigned int

The mask indicating updated cumulative bits.

OLD_VALUE unsigned int

The register's old value overwritten.

VALUE unsigned int

The new value written to the VFP system register.

VFP_S_REGS

VFP/NEON S 32 bit register write. Fields:

ALIASING enum

Type of register aliasing used.

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a write to the corresponding bit in the register. For example, 0x1 would indicate a write to VALUE[0].

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

VFP_S_REGS_READ

VFP/NEON S 32 bit register read. Fields:

ID unsigned int

The register number.

VALUE unsigned int

The current value of the register.

VIRTUAL_TAG_ALLOCATED

Virtual tag information when allocated. Fields:

TAG_VADDR unsigned int

Virtual tag virtual address.

VADDR unsigned int

Data Virtual address.

VIRTUAL_TAG unsigned int

Virtual tag value.

WAYPOINT

Signals end of basic block execution. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ISET enum

Origin instruction set.

IS_COND bool

Indicates if this is a conditional waypoint.

PC unsigned int

Origin address (or 0 if unavailable).

TAKEN bool

Indicates if this waypoint was taken.

TARGET unsigned int

Destination address.

TARGET_ISET enum

Destination instruction set.

WFE_END

WFE ended. Fields:

INST_COUNT unsigned int

Ticks count when leaving WFE.

WFE_EVENT_REGISTER

WFE event register status: set/clear, reason. Fields:

INST_COUNT unsigned int

Ticks count.

REASON enum

Reason for set/clear. Only REASON==0 (Cleared by WFE) clears the bit, all other reasons set it.

WFE_IGNORED

WFE ignored. Fields:

EVENT bool

This WFE was ignored because the event register was set.

INST_COUNT unsigned int

Ticks count when ignoring WFE.

TRAPPED bool

This WFE was trapped.

WFE_START

WFE entered. Fields:

INST_COUNT unsigned int

Ticks count when entering WFE.

WFI_END

WFI ended. Fields:

INST_COUNT unsigned int

Ticks count when leaving WFI.

WFI_IGNORED

WFI ignored. Fields:

DISABLED bool

This WFI was ignored because WFI is disabled.

INST_COUNT unsigned int

Ticks count when ignoring WFI.

REASON enum

specifies reason why WFI trace was ignored.

TRAPPED bool

This WFI was trapped.

WFI_START

WFI entered. Fields:

INST_COUNT unsigned int

Ticks count when entering WFI.

WFI_WAKEUP

WFI wakeup. Fields:

INST_COUNT unsigned int

Ticks count when WFI wakeup occurred.

REASON enum

Reason for wakeup.

2.20 ARM_Cortex-A55

This section describes the trace sources.

AA64_ASE_SVE_REGS

Modification of AArch64 Advanced SIMD or SVE Vector Registers (via SIMD or Scalar operations), SVE Predicate Registers and ZT0. Fields:

ID **enum**

The register identifier.

MASK **unsigned int**

Bitmask of the register to signal the modified bits in the VALUE field.

SM **bool**

Whether the PE is in Streaming Mode.

VALUE **unsigned int**

Value of AArch64 vector register.

ASYNC_MEMORY_FAULT

Context ID Register write. Fields:

FAULT **unsigned int**

Fault status present in ESR.ISS encoding for Serror in ARCH64 and present in DFSR encoding for external abort format in ARCH32.

PADDR **unsigned int**

Physical Address (or 0 if unavailable).

VADDR **signed int**

Virtual Address (or 0 if unavailable).

ATOMIC_END_ACCESS

Bus trace access for atomics. Fields:

ACCESS_FAIL **bool**

Memory access failed.

ADDR **unsigned int**

The virtual address of the access.

ATTR **unsigned int**

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

COMPARE_VALUE **unsigned int**

Compare value for CAS.

LOAD_VALUE unsigned int

Loaded values.

NSDESC unsigned int

The security state of the access.

OPERAND_VALUE unsigned int

Operation's operand.

OPERATION enum

Operation type.

PADDR unsigned int

The physical address of the access.

PRIV bool

Is this a privileged access?.

ATOMIC_START_ACCESS

Bus trace access for atomics. Fields:

ADDR unsigned int

The virtual address of the access.

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

COMPARE_VALUE unsigned int

Compare value for CAS.

NSDESC unsigned int

The security state of the access.

OPERAND_VALUE unsigned int

Operation's operand.

OPERATION enum

Operation type.

PADDR unsigned int

The physical address of the access.

PRIV bool

Is this a privileged access?.

ArchMsg.Error.exit_code

DISPLAY exit code: %{EXITCODE} component kind: %{KIND} reason: %{REASON}. Fields:

EXITCODE signed int

Exit code to be returned.

KIND string

Component kind that invoked the exit code trace.

REASON string

Reason why the exit code trace was invoked (optional).

ArchMsg.Warning.IgnoredWarmResetRequest

Event fires when software attempts to request warm reset through write to deprecated DBGPRCR.CWRR.

ArchMsg.Warning.RAS_ErrselrUnpredictableState

DISPLAY ERRSELR is in unpredictable state due to value greater than number_of_error_records written to it.

ArchMsg.Warning.RAS_UnpredictableStateWrite

DISPLAY %{REG_NAME} write ignored due to unpredictable ERRSELR state. Fields:

REG_NAME string

Name of register being written.

ArchMsg.Warning.ReportedRevisionRegisterRead

DISPLAY Reported revision and patch do not match that of the cpu. Fields:

ORIGINAL_VALUE unsigned int

The value the register would have based on the modelled revision.

REGISTER string

The name of the register that has ben read.

REPORTED_VALUE unsigned int

The value the register has been overridden to.

ArchMsg.Warning.Unpredictably Indexed PM Event Register

TAGS UNPREDICTABLE DISPLAY Event %{IsDirect:(Selector value|Register)} %{SEL} is >= %{N}. Fields:

IsDirect unsigned int

Direct Access rather than indirect.

N unsigned int

Number of Accessible Registers.

SEL unsigned int

Selector.

ArchMsg.Warning.cache_contents_unknown

A cache has been activated without being invalidated and may contain **UNKNOWN** content. DISPLAY %{SIDE:(D|I)}-Cache was enabled in %{REGIME:(SECURE|NON_SECURE|HYP)} regime but was not invalidated since power-on: cache lines could contain **UNKNOWN** content %{SIDE:(Note: D-side caches must invalidate SECURE lines even if the cache is only used by NON_SECURE code in order to ensure that unexpected dirty lines tagged SECURE are not naturally evicted)}. Fields:

REGIME unsigned int

Current translation regime.

SIDE unsigned int

0 d-side cache or unified cache, 1 i-side.

ArchMsg.Warning.dcimvac_matches_watchpoint

TAGS **UNPREDICTABLE** DISPLAY Watchpoints matching an AArch32 DCIMVA are implementation defined.

ArchMsg.Warning.decode_fieldmismatch

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with different fields %{FIELD1} and %{FIELD2}. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_fracbitsnegative

TAGS **UNPREDICTABLE** DISPLAY VCVT Instruction is unpredictable with a negative number of fraction bits.

ArchMsg.Warning.decode_initblock

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable in an IT block.

ArchMsg.Warning.decode_initblocknotlast

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable in an IT block when not the last.

ArchMsg.Warning.decode_invalid_condition

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable when the condition is not AL.

ArchMsg.Warning.decode_invalidfieldcombination

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with the values in %{FIELD1} and %{FIELD2}. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_invalidvalue

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with the value in %{FIELD}. Fields:

FIELD enum

Field name.

ArchMsg.Warning.decode_mem_hint_unallocated

TAGS **UNPREDICTABLE** DISPLAY There is no memory hint allocated to this bit pattern.

ArchMsg.Warning.decode_registermatch

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with %{FIELD1} and %{FIELD2} as the same register. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_registermismatch

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with %{FIELD1} and %{FIELD2} as different registers. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_registeroutofrange

TAGS **UNPREDICTABLE** DISPLAY Floating-point instruction is unpredictable indexing beyond the end of the register bank.

ArchMsg.Warning.decode_sbzsbo

TAGS **UNPREDICTABLE** DISPLAY Reserved bits in the instruction are not canonical.

ArchMsg.Warning.decode_transactiontoobig

TAGS **UNPREDICTABLE** DISPLAY Attempt to perform a memory transaction of more than 128 bytes.

ArchMsg.Warning.decode_unpred_other

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable.

ArchMsg.Warning.decode_unpreduseofpc

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable when %{FIELD} is PC. Fields:

FIELD enum

Field name.

ArchMsg.Warning.decode_unpreduseofr13

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable when %{FIELD} is R13. Fields:

FIELD enum

Field name.

ArchMsg.Warning.decode_unpreduseofr13inreglist

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with R13 in register-list.

ArchMsg.Warning.decode_useofd16orhigher

TAGS **UNPREDICTABLE** DISPLAY Attempt to use register D16 or higher where it is not allowed.

ArchMsg.Warning.decode_writebackandbaseinlist

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with write back when the base register is in the transfer list.

ArchMsg.Warning.decode_zeroregistersinlist

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable when no registers are to be transferred.

ArchMsg.Warning.recursive_branch

An instruction is performing a branch that targets the same instruction. If this was intended then a WFI might be more effective DISPLAY recursive branch detected at PC=%{PC}. PRIMARY KEY PC. Fields:

PC unsigned int

Address of instruction causing the branch.

ArchMsg.Warning.recursive_exception

An instruction has generated an exception that targets the same instruction. DISPLAY recursive %{TYPE} exception detected at PC=%{PC} (CPSR=%{CPSR}). PRIMARY KEY PC. Fields:

CPSR unsigned int

Processor state.

PC unsigned int

Address of instruction causing the exception.

TYPE enum

Type of exception.

ArchMsg.Warning.reentrant_vector_catch

TAGS **UNPREDICTABLE** DISPLAY Vector catch debug event on Prefetch or Data abort vector %{ADDR}. Fields:

ADDR unsigned int

vector address.

ArchMsg.Warning.reserved_it_state

TAGS **UNPREDICTABLE** DISPLAY Execution with reserved IT state %{ITSTATE}. Fields:

ITSTATE unsigned int

reserved IT state.

ArchMsg.Warning.tlb_contents_unknown

Translation has been enabled without invalidating the TLB. DISPLAY Translation was enabled in the %{PAS:(Secure|Non-secure|Root|Realm)} EL%{EL:d} translation regime (vmid=%{VMID:x}) without invalidating %{INVALIDITY:(|d-side |i-side |)}entries since power-on: %{INVALIDITY:(|D|I|)}TLB could contain **UNKNOWN** content. Fields:

EL unsigned int

Exception level of translation regime.

INVALIDITY unsigned int

Which TLB may be invalid (if TLBs are separate).

NON_SECURE bool

Whether the translation regime is non-secure.

PAS unsigned int

Physical address space of translation regime.

VMID unsigned int

Current VMID, if applicable.

ArchMsg.Warning.unknown_DLR_DSPSR

Exiting debug state with unknown DLR or DSPSR.

ArchMsg.Warning.unknown_ELR_SPSR

Returning from debug exception with unknown ELR or SPSR.

ArchMsg.Warning.unpred_FAR_on_LRCPC3_non_atomic_mem_access

TAGS **UNPREDICTABLE** Unpredictable FAR/FSTATUS on LRCPC3 pair instructions resulting in two separate memory accesses
DISPLAY Unpredictable FAR on LRCPC3 non atomic memory access at %{ADDR1} with FSTATUS %{FSTATUS1} or %{ADDR2} with FSTATUS %{FSTATUS2}. Fields:

ADDR1 unsigned int

Possible FAR address 1.

ADDR2 unsigned int

Possible FAR address 2.

FSTATUS1 unsigned int

Possible fault status value 1.

FSTATUS2 unsigned int

Possible fault status value 2.

ArchMsg.Warning.unpred_async_tag_check_fail_on_sve_speculative_load_do_update_ffr

Unpredictable set to FALSE of the FFR element associated to SVE speculative load with an async tag check fail TAGS **UNPREDICTABLE**.

ArchMsg.Warning.unpredictable_a32_breakpoint

TAGS **UNPREDICTABLE** DISPLAY Unpredictable breakpoint on a A32 instruction at %{ADDR1} Offset %{OFFSET} BAS %{BAS} for a %{IS_ADDRESS_MISMATCH_BREAKPOINT:(Mismatch|Match)} Breakpoint number %{N:d} Breakpoint address %{ADDR2}. Fields:

ADDR1 unsigned int

Address.

ADDR2 unsigned int

Breakpoint address.

BAS unsigned int

BAS field.

IS_ADDRESS_MISMATCH_BREAKPOINT bool

Is a Mismatch breakpoint, not a Match breakpoint.

N unsigned int

Breakpoint number.

OFFSET unsigned int

Always has a value of zero.

ArchMsg.Warning.unpredictable_banked_register_access

TAGS **UNPREDICTABLE** DISPLAY Access to banked register (R=%{R:d} SYSm=%{SYSM:2x}) is unpredictable. Fields:

R unsigned int

R bit.

SYSM unsigned int

SYSm.

ArchMsg.Warning.unpredictable_ccfail_undef

TAGS **UNPREDICTABLE** DISPLAY An instruction %{OPCODE} at VA=%{VADDR} failed its condition codes check but would otherwise have caused a non-data-dependent trap or undefined exception. Behaviour is implementation-defined choice of exception or nop. Fields:

OPCODE unsigned int

opcode of the instruction.

VADDR unsigned int

Virtual address of the instruction being executed.

ArchMsg.Warning.unpredictable_ici_bits

TAGS **UNPREDICTABLE** DISPLAY ICI bits do not apply or are architecturally unpredictable for instruction.

ArchMsg.Warning.unpredictable_t32_breakpoint

TAGS **UNPREDICTABLE** DISPLAY Unpredictable breakpoint on a T32 instruction at %{ADDR1} Offset %{OFFSET} BAS %{BAS} for a %{IS_ADDRESS_MISMATCH_BREAKPOINT:(Mismatch|Match)} Breakpoint number %{N:d} Breakpoint address %{ADDR2}. Fields:

ADDR1 unsigned int

Address.

ADDR2 unsigned int

Breakpoint address.

BAS unsigned int

BAS field.

IS_ADDRESS_MISMATCH_BREAKPOINT bool

Is a Mismatch breakpoint, not a Match breakpoint.

N unsigned int

Breakpoint number.

OFFSET unsigned int

Indicates which halfword of a 32-bit Thumb instruction the breakpoint is set on. 0 for the first halfword or if the instruction is 16 bits wide, or 1 for the second halfword.

ArchMsg.Warning.unpredictable_t32_vector_catch

TAGS **UNPREDICTABLE** DISPLAY Unpredictable Vector catch exception on a T32 instruction at **%{ADDR}** Offset **%{OFFSET}**. Fields:

ADDR unsigned int

Address.

OFFSET unsigned int

Indicates which halfword of a 32-bit Thumb instruction the vector catch exception is set on. 0 for the first halfword or if the instruction is 16 bits wide, or 1 for the second halfword.

ArchMsg.Warning.unpredictable_watchpoint_far

TAGS **UNPREDICTABLE** DISPLAY The value for FAR/EDWAR for hitting this watchpoint may be anywhere in the range **%{LOWER_BOUND}** to **%{UPPER_BOUND}**. Fields:

LOWER_BOUND unsigned int

the lowest address accessed by the instruction that triggered the watchpoint.

UPPER_BOUND unsigned int

the highest watchpointed address accessed by that instruction.

ArchMsg.Warning.unpredictable_watchpoint_far_isscmp

DISPLAY The value for FAR/EDWAR for hitting this watchpoint may be anywhere in the range **%{LOWER_BOUND}** to **%{UPPER_BOUND}**. Fields:

LOWER_BOUND unsigned int

the lowest address accessed by the instruction that triggered the watchpoint.

UPPER_BOUND unsigned int

the highest watchpointed address accessed by that instruction.

ArchMsg.Warning.warning_AdvSIMDEExpandImmUnexpectedZero

TAGS **UNPREDICTABLE** DISPLAY AdvSIMDEExpandImm may treat this immediate value as **UNPREDICTABLE**.

ArchMsg.Warning.warning_ConditionalSMC

TAGS **UNPREDICTABLE** DISPLAY SMC instruction has **UNPREDICTABLE** effects when conditional and when combined with traps.

ArchMsg.Warning.warning_TLBID_out_of_range

TLBID out of range in TLBI according to the configuration of NIS, NVIS, NOS and NVOS. Fields:

NIS unsigned int

NIS.

NOS unsigned int

NOS.

NVIS unsigned int

NVIS.

NVOS unsigned int

NVOS.

TLBID unsigned int

The TLBID value used in the TLBI instruction.

ArchMsg.Warning.warning_TTL_or_TG_mismatches_current_translation_granule

TTL or TG mismatches current translation granule in TLBI VA/RVA/IPA/RIPA. Fields:

CURRENT_TG unsigned int

Current TG as indicated by TCR_ELx or VTCR_EL2.

TLBI_TG unsigned int

TG (or TTL[3:2]) used in TLBI instruction.

ArchMsg.Warning.warning_access_crosses_page_boundary

TAGS **UNPREDICTABLE** DISPLAY Access crosses page boundary.

ArchMsg.Warning.warning_access_crosses_page_boundary_has_fault

TAGS **UNPREDICTABLE** DISPLAY Access that crosses page boundary has fault(s).

ArchMsg.Warning.warning_access_crosses_page_boundary_has_fault_on_both_pages

TAGS **UNPREDICTABLE** DISPLAY Access that crosses page boundary has a fault on each page.
Fields:

ADDR1 unsigned int

FAR address on first page.

ADDR2 unsigned int

FAR address on second page.

FSTATUS1 unsigned int

Fault status on first page.

FSTATUS2 unsigned int

Fault status on second page.

ArchMsg.Warning.warning_access_crosses_page_boundary_spanning_different_memory

TAGS **UNPREDICTABLE** DISPLAY Access at address %{ADDR} crosses page from MEM_TYPE %{MEMTYPE_PAGE1} to %{MEMTYPE_PAGE2}. Fields:

ADDR unsigned int

address of access crossing page.

MEMTYPE_PAGE1 signed int

First Page's memory type (0-Normal, 1-Non-Normal).

MEMTYPE_PAGE2 signed int

Second Page's memory type (0-Normal, 1-Non-Normal).

ArchMsg.Warning.warning_access_wraps_around_memory

TAGS **UNPREDICTABLE** DISPLAY Access wraps around memory in %{WIDTH:d}bit mode.

Fields:

WIDTH unsigned int

Address width.

ArchMsg.Warning.warning_bcr_linking_status

TAGS **UNPREDICTABLE** DISPLAY DBGBCR.{BT2,BT} combination gives unpredictable behaviour.

ArchMsg.Warning.warning_bcr_mask_reserved

TAGS **UNPREDICTABLE** DISPLAY DBGBCR_ELn.MASK is set to a reserved value.

ArchMsg.Warning.warning_bvr_masked_bit_not_zero

TAGS **UNPREDICTABLE** DISPLAY DBGBVRn_EL1.VA contains a bit masked by DBGBCRn_EL1.MASK that is not zero.

ArchMsg.Warning.warning_ccsidr_unimplemented_level

TAGS **UNPREDICTABLE** DISPLAY CCSIDR read while CSSELR %{CSSELR:x} points to an unimplemented cache level. Fields:

CSSELR unsigned int

current effective value of CSSELR.

ArchMsg.Warning.warning_change_to_ns_when_tge_set

TAGS **UNPREDICTABLE** DISPLAY Attempting to change NS to 1 when HCR.TGE = 1.

ArchMsg.Warning.warning_contiguous_bit_check_abort

TAGS **UNPREDICTABLE** DISPLAY An abort occurred whilst attempting to check TLB entry is contiguous at %{ENTRY_ADDR}. Fields:

ENTRY_ADDR unsigned int

address of conflicting TLB entry.

ArchMsg.Warning.warning_contiguous_bit_error

TAGS **UNPREDICTABLE** DISPLAY TLB entry at %{ENTRY_ADDR} was not contiguous with entry at %{CONTIG_ADDR} Contents are %{ENTRY_DATA} but expected %{CONTIG_DATA}. Fields:

CONTIG_ADDR unsigned int

address of the TLB first read, with which this entry is expected to be contiguous.

CONTIG_DATA unsigned int

expected contents based on the entry at CONTIG_ADDR.

ENTRY_ADDR unsigned int

address of conflicting TLB entry.

ENTRY_DATA unsigned int

contents of conflicting TLB entry.

ArchMsg.Warning.warning_contiguous_bit_error_D128

TAGS **UNPREDICTABLE** DISPLAY TLB entry at %{ENTRY_ADDR} was not contiguous with entry at %{CONTIG_ADDR} Contents are %{ENTRY_DATA_HI} %{ENTRY_DATA_LO} but expected %{CONTIG_DATA}. Fields:

CONTIG_ADDR unsigned int

address of the TLB first read, with which this entry is expected to be contiguous.

CONTIG_DATA unsigned int

expected contents based on the entry at CONTIG_ADDR.

ENTRY_ADDR unsigned int

address of conflicting TLB entry.

ENTRY_DATA_HI unsigned int

Higher bit contents of conflicting TLB entry.

ENTRY_DATA_LO unsigned int

Lower bit contents of conflicting TLB entry.

ArchMsg.Warning.warning_cp10_cp11_mismatch

TAGS **UNPREDICTABLE** DISPLAY Unpredictable configuration of CP10 and CP11 controls at EL %{EL:d}. Fields:

EL unsigned int

EL owning the control.

ArchMsg.Warning.warning_cp10_cp11_reserved_value

TAGS **UNPREDICTABLE** DISPLAY Unpredictable configuration of CPACR CP10 (%{CP10}) or CP11 (%{CP11}). Fields:

CP10 unsigned int

CP10 access.

CP11 unsigned int

CP11 access.

ArchMsg.Warning.warning_csselr_level_out_of_range

TAGS **UNPREDICTABLE** DISPLAY CSSELR.Level out of range, written value %{LEVEL} with only %{IMPLEMENTED:d} levels implemented. Fields:

IMPLEMENTED unsigned int

number of cache levels implemented.

LEVEL unsigned int

written value of CSSELR.Level.

TYPE enum

The type of cache selected by CSSELR.InD and TnD.

ArchMsg.Warning.warning_debug_flow_control_bits_not_obeyed

TAGS **UNPREDICTABLE** DISPLAY Flow control bits for %{REGISTER} were not obeyed, causing %{ERROR}. Fields:

ERROR enum

type of error resulting.

REGISTER enum

register under consideration.

ArchMsg.Warning.warning_debug_register_access_during_reset

TAGS **UNPREDICTABLE** DISPLAY Unpredictable %{IS_WRITE:(read|write)} access to debug register offset %{OFFSET} during reset. Fields:

IS_WRITE unsigned int

Write Not Read.

OFFSET unsigned int

Register Offset.

ArchMsg.Warning.warning_decode_cps_inconsistent_fields

TAGS **UNPREDICTABLE** DISPLAY The mode setting fields in the CPS instruction are inconsistent.

ArchMsg.Warning.warning_decode_invalid_state

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable in current exception-level/security state.

ArchMsg.Warning.warning_default_cacheable_mmu_on

TAGS **UNPREDICTABLE** DISPLAY Default cacheable is enabled (HCR.DC = 1) while MMU is enabled.

ArchMsg.Warning.warning_default_cacheable_vmmu_off

TAGS **UNPREDICTABLE** DISPLAY Default cacheable is enabled (HCR.DC = 1) while HCR.VM = 0.

ArchMsg.Warning.warning_deprecated_wvr_bit_2_set

TAGS **UNPREDICTABLE** DISPLAY DBGWVR%{N:d}_EL1=%{WVR:h} has bit 2 set, deprecated in ARMv8. Fields:

N unsigned int

index.

WVR unsigned int

DBGWVR.

ArchMsg.Warning.warning_eret_while_software_step_active_pending

Missing ISB between setting MDSCR_EL1.SS and ERET TAGS **UNPREDICTABLE**.

ArchMsg.Warning.warning_exclusive_to_non_normal

TAGS **UNPREDICTABLE** DISPLAY Exclusive Access to Non-Normal Memory %{ADDR}. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_exclusive_to_non_writeback

TAGS **UNPREDICTABLE** DISPLAY Exclusive Access to Non-Writeback-Cacheable Memory
%{ADDR}. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_execute_from_device_memory

TAGS **UNPREDICTABLE** DISPLAY An attempt was made to execute from Device memory.

ArchMsg.Warning.warning_illegal_cpsr_mode

TAGS **UNPREDICTABLE** DISPLAY Writing an illegal or unimplement mode (%{NEW_MODE:h})
to CPSR was unpredictable. Fields:

NEW_MODE unsigned int

New value of CPSR.M.

ArchMsg.Warning.warning_illegal_srs_mode

TAGS **UNPREDICTABLE** DISPLAY Illegal or **UNPREDICTABLE** mode (%{MODE:h}) used for SRS
instruction. Fields:

MODE unsigned int

mode for Banked SP.

ArchMsg.Warning.warning_implementation_defined_read_debug_register_in_SCS

Read Access to Debug registers from the processor is IMPLEMENTATION DEFINED TAGS
IMP_DEF. Fields:

PPB_OFFSET unsigned int

debug register offset.

ArchMsg.Warning.warning_implementation_defined_sequential_security_transitions_supported

The behavior of sequential instruction fetches that cross from non-secure to secure memory
and contain an SG instruction is CONSTRAINED **UNPREDICTABLE** TAGS **UNPREDICTABLE**.

ArchMsg.Warning.warning_implementation_defined_stack_limit_check_supported

It is **IMPLEMENTATION DEFINED** whether stack pointer limit checking is performed for this
instruction TAGS IMP_DEF. Fields:

ADDRESS unsigned int

Address of instruction.

OPCODE unsigned int

instruction opcode.

ArchMsg.Warning.warning_implementation_defined_taggednonsharedmemorytype

TAGS IMP_DEF DISPLAY Memory attribute has Tagged, non shared memory type; it is
IMPLEMENTATION DEFINED whether the memory location is treated as Tagged or non-Tagged.

ArchMsg.Warning.warning_implementation_defined_write_debug_register_in_SCS

Write Access to Debug registers from the processor is IMPLEMENTATION DEFINED TAGS IMP_DEF. Fields:

DATA unsigned int

data attempted to be written.

PPB_OFFSET unsigned int

debug register offset.

ArchMsg.Warning.warning_invalid_tcr_granule

TAGS UNPREDICTABLE DISPLAY TCR.TG%{TG_ID:(0|1)} bits have been set to a granule size %{{REQUEST:d}}K not implemented - using %{{SUBSTITUTE:d}}K. Fields:

REQUEST signed int

page size requested (or 0 for reserved).

SUBSTITUTE signed int

best guess available page size.

TG_ID bool

bits TG0 or TG1.

ArchMsg.Warning.warning_load_multiple_user_registers_from_user_mode

TAGS UNPREDICTABLE DISPLAY An LDM(user registers) instruction executed from user mode.

ArchMsg.Warning.warning_mte_in_mem_attr_not_supported

TAGS UNPREDICTABLE DISPLAY Memory attribute %{{ATTR}} has Tagged memory type but Armv8.5-MTE is not implemented. Fields:

ATTR unsigned int

Memory attribute.

ArchMsg.Warning.warning_reserved_breakpoint_state_match

TAGS UNPREDICTABLE DISPLAY %{{IS_BREAKPOINT:(Watchpoint|Breakpoint)}} programmed with a reserved combination of HMC, SSC and %{{IS_BREAKPOINT:(PAC|PMC)}}. Fields:

IS_BREAKPOINT unsigned int

Is a breakpoint, not a watchpoint.

ArchMsg.Warning.warning_shareability

TAGS UNPREDICTABLE DISPLAY Unpredictable: combination of the 1st and 2nd stages of translation is Normal Inner Non-Cacheable, Outer Non-Cacheable.

ArchMsg.Warning.warning_software_step_set_while_enabled

TAGS UNPREDICTABLE DISPLAY MDSCR_EL1.SS set to 1 while software step debug exceptions are enabled.

ArchMsg.Warning.warning_thumb_instruction_wraps_around_memory

TAGS UNPREDICTABLE DISPLAY Thumb instruction wraps around memory.

ArchMsg.Warning.warning_ttbr_sbz_bits_are_not_zero

TAGS UNPREDICTABLE DISPLAY `%{TTBR}[%{MSB:d}:0]` should be zero, but are not. Fields:

MSB unsigned int

x-1.

TTBR enum

which TTBR register.

ArchMsg.Warning.warning_unaligned_address_dbgdttrrx_write

TAGS UNPREDICTABLE DISPLAY Unaligned address `%{ADDR}` in X0/R0 when DBGDTTRRX written in memory access mode. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unaligned_address_dbgdttrtx_read

TAGS UNPREDICTABLE DISPLAY Unaligned address `%{ADDR}` in X0/R0 when DBGDTTRTX read in memory access mode. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unaligned_to_device

TAGS UNPREDICTABLE DISPLAY Unaligned Access to Device Memory `%{ADDR}`. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unaligned_to_strongly_ordered

TAGS UNPREDICTABLE DISPLAY Unaligned Access to Strongly Ordered Memory `%{ADDR}`. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unknown_sau_rnr

TAGS UNPREDICTABLE DISPLAY SAU_RNR was set to an unsupported value.

ArchMsg.Warning.warning_unpredictable_AIRCR_PRIS_and_BFHFNMINs

TAGS UNPREDICTABLE DISPLAY The effect of setting both AIRCR.BFHFNMINs and AIRCR.PRIS to 1 is UNPREDICTABLE.

ArchMsg.Warning.warning_unpredictable_AIRCR_VECTCLRACTIVE_when_not_in_debug

TAGS UNPREDICTABLE DISPLAY The effect of writing a 1 to AIRCR.VECTCLRACTIVE if the processor is not halted in Debug state is UNPREDICTABLE.

ArchMsg.Warning.warning_unpredictable_AIRCR_VECTRESET_and_SYSRESETREQ

TAGS UNPREDICTABLE DISPLAY When the processor is halted in Debug state, if a write to the register writes a 1 to both VECTRESET and SYSRESETREQ, the behavior is UNPREDICTABLE.

ArchMsg.Warning.warning_unpredictable_AIRCR_VECTRESET_when_not_in_debug

TAGS **UNPREDICTABLE** DISPLAY The effect of writing a 1 to AIRCR.VECTRESET if the processor is not halted in Debug state is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_AIRCR_incorrect_VKEY

TAGS **UNPREDICTABLE** DISPLAY The value 0x05FA must be written to AIRCR.VKEY, otherwise the register write is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_EXC_RETURN_Reserved_Bit

EXC_RETURN[23:7] are reserved with the special condition that all bits should be written as one. Values other than all 1s are **UNPREDICTABLE**. TAGS **UNPREDICTABLE**. Fields:

EXC_RETURN unsigned int

Special PC value.

SBO_MASK unsigned int

Which bits Should Be One to not UNPRED.

ArchMsg.Warning.warning_unpredictable_PMEVTYPEPn_ELO_VS_reserved_value

TAGS **UNPREDICTABLE** DISPLAY Unpredictable reserved value of PMEVTYPEn<n>_ELO.VS.

ArchMsg.Warning.warning_unpredictable_change_to_DEMCR_MON_STEP_at_insufficient_priority

TAGS **UNPREDICTABLE** DISPLAY The effect of changing DEMCR.MON_STEP at an execution priority that is lower than the priority of the DebugMonitor exception is **CONSTRAINED UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_change_to_priority_of_active_exception

TAGS **UNPREDICTABLE** DISPLAY Changing the priority of an active exception is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_clear_lspact

TAGS **UNPREDICTABLE** DISPLAY **UNPREDICTABLE** state on context switch. Fields:

DRVEXC_TAKEN unsigned int

Derived exception taken, not pended.

REACHED_LAST_STORE unsigned int

Only the last faulted.

ArchMsg.Warning.warning_unpredictable_dbg_exit_unaligned_dlr

TAGS **UNPREDICTABLE** DISPLAY Debug exit to AArch32 with DLR[0] set to 1. PC[0] can be set to 0 or to DLR[0].

ArchMsg.Warning.warning_unpredictable_exception_catch

TAGS **UNPREDICTABLE** DISPLAY Generation of an exception catch event is unpredictable.

ArchMsg.Warning.warning_unpredictable_exception_return_inconsistent_state

TAGS **UNPREDICTABLE** DISPLAY state on exception return is unpredictable.

ArchMsg.Warning.warning_unpredictable_exception_return_instruction

TAGS **UNPREDICTABLE** DISPLAY exception return instruction is unpredictable.

ArchMsg.Warning.warning_unpredictable_in_debug_state

TAGS **UNPREDICTABLE** DISPLAY Instruction is **UNPREDICTABLE** when executed in debug state.

ArchMsg.Warning.warning_unpredictable_mair_encoding

TAGS **UNPREDICTABLE** DISPLAY Memory attribute %{ATTR} has invalid MAIR encoding; this may lead to unpredictable results. Fields:

ATTR unsigned int

Memory attribute.

ArchMsg.Warning.warning_unpredictable_pmu_counter_access

TAGS **UNPREDICTABLE** DISPLAY Access to PMU counters from non-secure EL0 or EL1 is **UNPREDICTABLE** with MDCR_EL2.HPMN set to 0.

ArchMsg.Warning.warning_unpredictable_prioritization_breakpoint_match_vector_catch

TAGS **UNPREDICTABLE** DISPLAY Two events with the same priority occurred at the same instruction: Address Matching Vector Catch debug event, and Breakpoint debug event. It is constrained unpredictable which is taken.

ArchMsg.Warning.warning_unpredictable_stack_selection

TAGS **UNPREDICTABLE** DISPLAY **UNPREDICTABLE** state on context switch.

ArchMsg.Warning.warning_unpredictable_tsize_out_of_range

TAGS **UNPREDICTABLE** DISPLAY %{TCR}.T%{TTBR:d}SZ (%{TSIZE:d} bits) is out of range. Must be between 25 and %{TSIZE_MAX:d}. Fields:

TCR enum

Which TCR.

TSIZE signed int

Value of TSize.

TSIZE_MAX signed int

The maximum allowed value.

TTBR unsigned int

Which TTBR region.

ArchMsg.Warning.warning_unpredictable_unaligned_pc_as_base_register

TAGS **UNPREDICTABLE** DISPLAY The PC value must be word-aligned, otherwise the behavior of the instruction is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_unaligned_pop_stack

TAGS **UNPREDICTABLE** DISPLAY **CONSTRAINED UNPREDICTABLE** unaligned stack frame %{FRAME_ADDR:x} in exception-return. Fields:

FRAME_ADDR unsigned int

unaligned stack frame address.

ArchMsg.Warning.warning_unpredictable_unaligned_to_non_writeback

TAGS **UNPREDICTABLE** DISPLAY Unaligned Access to Non-Writeback-Cacheable Memory.

ArchMsg.Warning.warning_unpredictable_vmsa_memattrib

TAGS **UNPREDICTABLE** DISPLAY Unpredictable vmsa memory attribute with texcb value programmed to %{TEXCB} when tex remap is %{TEX_REMAP}. Fields:

TEXCB unsigned int

texcb value.

TEX_REMAP bool

use tex remap.

ArchMsg.Warning.warning_unpredictable_vtcr_t0sz_sl0

TAGS UNPREDICTABLE DISPLAY The combination of VTCR.TOSZ=%{TOSZ:d} and VTCR.SLO=%{SLO:d} is unpredictable. Fields:

SLO signed int

VTCR.SLO, starting level for stage 2 translation table walks.

TOSZ signed int

VTCR.TOSZ, required input address range.

ArchMsg.Warning.warning_unpredictable_write_DHCSR

TAGS UNPREDICTABLE DISPLAY UNPREDICTABLE set of bit-changes in DHCSR. Fields:

NEW unsigned int

data attempted to be written.

OLD unsigned int

previous value.

ArchMsg.Warning.warning_unpredictable_write_SHCSR

TAGS UNPREDICTABLE DISPLAY UNPREDICTABLE setting a pending bit to 1 for an exception with priority >= execution priority.

ArchMsg.Warning.warning_unsupported_access_to_memory_mapped_register

TAGS UNPREDICTABLE DISPLAY Access to memory mapped register is unsupported.

ArchMsg.Warning.warning_user_jmcr_access

TAGS UNPREDICTABLE DISPLAY User %{WRITE:(write|read)} access to JMCR is unpredictable. Fields:

WRITE unsigned int

access is write.

ArchMsg.Warning.warning_wcr_mask_and_bas

TAGS UNPREDICTABLE DISPLAY DBGWCR_EL1.MASK is non-zero and BAS is not 0xff.

ArchMsg.Warning.warning_wcr_mask_reserved

TAGS UNPREDICTABLE DISPLAY DBGWCR_ELn.MASK is set to a reserved value.

ArchMsg.Warning.warning_wcr_non_configuous_bas

TAGS UNPREDICTABLE DISPLAY DBGWCR_EL1.BAS has non-contiguous set of ones.

ArchMsg.Warning.warning_wvr_masked_bit_not_zero

TAGS UNPREDICTABLE DISPLAY DBGWVRn_EL1.VA contains a bit masked by DBGWCRn_EL1.MASK that is not zero.

ArchMsg.Why.why_branch_target_exception

A branch target exception has occurred. DISPLAY branch target exception: target pc=%{PC}, target opcode=%{OPCODE}. PRIMARY KEY PC. Fields:

BTTYPE unsigned int

pstate.BTYPE at time branch taken.

OPCODE unsigned int

opcode at branch target.

PC unsigned int

Virtual Address of branch target.

ArchMsg.Why.why_illegal_state

An illegal mode change has occurred. DISPLAY illegal mode change: %{MESSAGE}. Fields:

MESSAGE string

The message.

BRANCH_MISPREDICT

Simulating branch mispredict. Fields:

PC unsigned int

Origin address (or 0 if unavailable).

BRA_DIR

Direct branches, to immediate address. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISET enum

The instruction set of the branch instruction.

IS_COND bool

Indicates if this is a conditional branch.

IS_HINTED bool

Whether the conditional instruction is hinted.

IS_ISB bool

Whether the instruction is ISB.

IS_LINK bool

Indicates if this is a branch with link.

PC unsigned int

The address of the branch instruction.

TARGET_ISET enum

The instruction set after the branch.

TARGET_PC unsigned int

The address the instruction branches to.

BRA_INDIR

Indirect branches, perhaps to a register. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISET enum

The instruction set of the branch instruction.

IS_COND bool

Indicates if this is a conditional branch.

IS_HINTED bool

Whether the conditional instruction is hinted.

IS_ISB bool

Whether the instruction is ISB.

IS_LINK bool

Indicates if this is a branch with link.

PC unsigned int

The address of the branch instruction.

TARGET_ISET enum

The instruction set after the branch.

TARGET_PC unsigned int

The address the instruction branches to.

BRA_INDIR_CCFAIL

Indirect branches fail, perhaps to a register.

BRA_RET_CCFAIL

Branch return fail, perhaps to a register.

CACHE_MAINTENANCE_OP

Cache Maintenance Operation. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

DATA unsigned int

Specified MVA or Set/way.

FUNCTION enum

Clean / Invalidate.

SCOPE enum

Affected region.

SIDE enum

Inst / Data.

CCFAIL

Conditional instruction condition check fail. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CCFAIL_UNC

Unconditional instruction condition check fail. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CCPASS

Conditional instruction condition check pass. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CCPASS_UNC

Unconditional instruction condition check pass. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CODE_CACHE_MAINT

Code cache maintenance. Fields:

END_INCL unsigned int

Inclusive end address for VA or PA requests.

NS bool

Non-secure world for PA.

START unsigned int

Start address for VA or PA requests.

TYPE enum

Request type.

COMPILE_BLOCK_END

Last instruction of basic block translated. Fields:

VADDR unsigned int

Address of next instruction after this basic block.

COMPILE_BLOCK_START

First instruction of basic block translated. Fields:

VADDR unsigned int

Address of first instruction in basic block.

COMPILE_INST

ARM instruction compiled. Fields:

DISASS string

Disassembly of the instruction.

ISSET enum

The instruction set of this instruction.

ITSTATE unsigned int

The ITSTATE current for the instruction.

OPCODE unsigned int

The opcode of the instruction.

PC unsigned int

The address of the instruction.

SIZE unsigned int

The size of the instruction in bytes.

CONTEXTIDR

Context ID Register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

NS bool

Secure or nonsecure banked register is accessed.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The new value written.

CONTEXT_SYNC

Called for every context synchronization event. Fields:

ARCH_CSE bool

Architectural Context Synchronization Event.

PC unsigned int

The address of the synchronization instruction.

SYSREG_SYNC bool

System register synchronization event.

CORE_ENDIAN

Core BE8 Big-Endian state changed. Fields:

BE8 bool

Core BE8 Big-Endian state.

CORE_INFO

Static processor attributes. Only triggered by a call to DumpState(). Fields:

ARCH_PROFILE enum

The architecture profile of the core.

CLUSTER_ID unsigned int

The cluster ID of this processor.

CORE_NUM unsigned int

The number of this core in an MP processor.

DETAILED_CORE_FEATURES enum

Detailed core features as a bitfield. Enum values are bitmasks to extract the info from the field.

FPU_VERSION enum

The VFP version implemented by the core.

MEM_ARCH enum

The memory architecture of the core.

MIDR_EL1 unsigned int

MIDR_EL1 value.

NUM_CORES unsigned int

The number of cores in this MP processor.

QUANTUM_SIZE unsigned int

The default quantum size of the core (in ticks).

SECURITY_FEATURES bool

Does the core have security features?.

CORE_LOADS

Processor load accesses. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ACQREL enum

Is this an acquire/release.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DATA unsigned int

The data read or written.

ELEMENT_SIZE unsigned int

Width of each element.

ENDIANNESS enum

Byte ordering of this access.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

LOCK enum

Normal, exclusive or locked access.

MECID unsigned int

Memory Encryption Context ID, if applicable.

MECID2 unsigned int

Memory Encryption Context ID of the second page, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PAGE_SIZE unsigned int

Page size as $\log_2(\text{size in bytes})$.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK, 2=Exclusive Failed, 3=Unsupported.

SIZE unsigned int

Width of the access in bytes. Only required if DATA is not traced.

TRANS bool

Is this a translated access.

VADDR unsigned int

The virtual address of the access.

CORE_REGS

Changes of the core registers R0 to R14. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number, 0 to 14.

MODE enum

Bank of the register accessed.

OLD_VALUE unsigned int

The old value overwritten.

PHYS_ID enum

The physical register accessed.

VALUE unsigned int

The new value written to the register.

CORE_REGS64

Changes of the core registers X0..X30, SP_ELn. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ID enum

The register number, 0-30 for X0-X30, >=32 for SP_ELn.

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

CORE_REGS64_READ

General purpose AArch64 register has been read X0..X30, SP_ELn. Fields:

ID enum

The register number, 0-30 for X0-X30, >=32 for SP_ELn.

VALUE unsigned int

The value read from the register.

CORE_STALLED_TRANSITION

Detect when a core is prevented from progressing PChannel states. Fields:

CURRENT_STATE enum

Current power mode of core.

REASON string

The reason a transition was stalled, and what must be done to unblock it.

TARGET_STATE enum

Desired power mode of core after transition.

CORE_STORES

Processor store accesses. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ACQREL enum

Is this an acquire/release.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DATA unsigned int

The data read or written.

ELEMENT_SIZE unsigned int

Width of each element.

ENDIANNESS enum

Byte ordering of this access.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

LOCK enum

Normal, exclusive or locked access.

MECID unsigned int

Memory Encryption Context ID, if applicable.

MECID2 unsigned int

Memory Encryption Context ID of the second page, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PAGE_SIZE unsigned int

Page size as $\log_2(\text{size in bytes})$.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK, 2=Exclusive Failed, 3=Unsupported.

SIZE unsigned int

Width of the access in bytes. Only required if DATA is not traced.

TRANS bool

Is this a translated access.

VADDR unsigned int

The virtual address of the access.

CP14_READ

System Coprocessor register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP14 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc1 unsigned int

opcode 1.

opc2 unsigned int
opcode 2.

CP14_WRITE

System Coprocessor register write. Fields:

CORE_NUM unsigned int
Core number in a multi processor.

CRm unsigned int
CRm.

CRn unsigned int
CRn.

NS bool
Secure or nonsecure banked register is accessed.

REG_NAME string
Name of the CP14 register accessed.

SECURITY_STATE enum
The current security state.

UNDEF bool
The register accessed is undefined.

UPDATED_VALUE unsigned int
Updated value of the register now it has been written.

VALUE unsigned int
The new value written.

opc1 unsigned int
opcode 1.

opc2 unsigned int
opcode 2.

CP15_READ

System Coprocessor register read. Fields:

CORE_NUM unsigned int
Core number in a multi processor.

CRm unsigned int
CRm.

CRn unsigned int
CRn.

NS bool
Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP15 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc1 unsigned int

opcode 1.

opc2 unsigned int

opcode 2.

CP15_READ64

System Coprocessor register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP15 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc unsigned int

opcode 1.

CP15_WRITE

System Control Coprocessor register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP15 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc1 unsigned int

opcode 1.

opc2 unsigned int

opcode 2.

CP15_WRITE64

System Coprocessor register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP15 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc unsigned int

opcode 1.

CPSR

CPSR change. (In AArch64 PSTATE represented in SPSR format). Fields:

CORE_NUM unsigned int

Core number in a multi processor.

OLD_VALUE unsigned int

The old CPSR value.

UNKNOWN unsigned int

Bits within the register that have unknown value.

VALUE unsigned int

The new CPSR value.

CRYPTO_SPEC

Every crypto instruction speculatively executed.

CorePowerStateChange

Core power-up state has changed. DISPLAY Core powered %{PU:(up|down)}. Fields:

PU bool

Whether the core powered up.

DATA_CACHE_SET_ALLOC_TAGS

Set allocation tags from DC GVA instruction. Fields:

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

NSDESC unsigned int

The security state of the access.

PADDR unsigned int

The physical address of the access.

PAS enum

The physical address space of the page.

RESPONSE enum

0=Aborted, 1=OK.

SIZE unsigned int

The size of memory in bytes.

TAG unsigned int

The value of the tag being set.

TAGGING_TYPE enum

Type of Memory tagging (Physical or Virtual).

TAG_VADDR unsigned int

The virtual address of the Virtual Tag if Virtual tagging is used; otherwise 0.

VADDR unsigned int

Virtual address used for setting allocation tags.

DATA_CACHE_ZERO

Zero and invalidate cache line from DC ZVA instruction. Fields:

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

NSDESC unsigned int

The security state of the access.

PADDR unsigned int

The physical address of the access.

PADDR_TAG unsigned int

If tag checked, the physical address tag for the access.

PAS enum

The physical address space of the page.

RESPONSE enum

0=Aborted, 1=OK.

SIZE unsigned int

The size of zero'd memory in bytes.

TAG_CHECKED bool

Is this access tag checked.

VADDR unsigned int

Virtual address of the zero'd cache line.

DEBUG_EVENT

Hardware debug support event. Fields:

EVENT enum

Description of event.

VALUE unsigned int

data value.

DEBUG_TARGET_ISET

Hardware debug target ISA. Fields:

TARGET_ISET enum

The target instruction set of the debug state.

DMI_ALLOCATE

DMI allocated. Fields:

ACCESS_DIRECTION enum

READ / WRITE.

PADDR unsigned int

The physical address of the DMI.

PTR unsigned int

DMI range pointer.

SIDE enum

DSIDE / ISIDE.

SIZE unsigned int

The size of DMI.

VADDR unsigned int

The virtual address of the DMI..

DMI_HIT

DMI hit, access used DMI (only fire for small DMI regions i.e. less than a full page). Fields:

ACCESS_DIRECTION enum

READ / WRITE.

PADDR unsigned int

The physical address of the DMI.

SIDE enum

DSIDE / ISIDE.

SIZE unsigned int

The size of the DMI.

VADDR unsigned int

The virtual address of the DMI..

DMI_REVOKE

DMI revoked. Fields:

ACCESS_DIRECTION enum

READ / WRITE.

PADDR unsigned int

The physical address of the DMI.

SIDE enum

DSIDE / ISIDE.

SIZE unsigned int

The size of DMI.

VADDR unsigned int

The virtual address of the DMI..

END_COMPILE

Compilation end. Fields:

END_OF_PAGE_COUNT unsigned int

Number of basic blocks exited due to page boundary since START_COMPILE.

FETCHFAIL_COUNT unsigned int

Number of basic blocks exited due to ifetch failure START_COMPILE.

INST_COUNT unsigned int

Number of instructions compiled since START_COMPILE.

NONSEQ_COUNT unsigned int

Number of basic blocks exited due to non-sequential instructions since START_COMPILE.

PAGE_STRADDLE_COUNT unsigned int

Number of basic blocks exited due to unaligned instructions crossing page since START_COMPILE.

ESB

Error Synchronization Barrier hit. Fields:

ASYNC bool

ESB is async.

IMPLICIT bool

ESB is implicit.

EXCEPTION

Exception taken. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ESR unsigned int

The value assigned to ESR when exception is taken.

ESR_EC unsigned int

The Value assigned to ESR.EC when exception is taken.

IS_PHYSICAL bool

Physical or Virtual exception.

IS_TRAP bool

Exception coming from a trap.

LR unsigned int

The value assigned to the link register.

PC unsigned int

The location where the exception occurred.

PREFERRED_RETURN unsigned int

The preferred return address for the exception.

TAKEN_LOCALLY bool

Whether exception is taken locally(same exception level).

TARGET_ISET enum

The instruction set of the exception handler code.

TARGET_PC unsigned int

The address the exception branches to.

VECTOR enum

The exception vector.

VECTOR_OFFSET unsigned int

The offset of an exception vector from the base address.

EXCEPTION_END

Every exception completed.

EXCEPTION_INFO

Trace human-readable reason of the current exception for data/instruction aborts, floating point exceptions and unknown exceptions. Fields:

DESCRIPTION string

Reason for current exception.

ESR_EC unsigned int

The value of ESR.EC when exception is taken.

EXCEPTION_RAISE

Every exception raised.

EXCEPTION_RETURN

Branches on leaving exception. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISSET enum

The instruction set of the branch instruction.

IS_COND bool

Indicates if this is a conditional branch.

IS_HINTED bool

Whether the conditional instruction is hinted.

IS_ISB bool

Whether the instruction is ISB.

IS_LINK bool

Indicates if this is a branch with link.

PC unsigned int

The address of the branch instruction.

TARGET_ISET enum

The instruction set after the branch.

TARGET_PC unsigned int

The address the instruction branches to.

EXCEPTION_RETURN_PREBRANCH

Exception return event that must occur before PC and processor state have updated.

EXCEPTION_START

Every exception started.

FIQ_TAKEN

FIQ taken exception.

FP_EXCEPTIONS_SIGNALLED

Trace which FP exceptions were signalled as a part of the FP operations performed by an instruction. Fields:

FP_EXCEPTIONS_SIGNALLED_MASK unsigned int

Indicates which FP exceptions were signalled by an instruction. This uses the same encoding as the cumulative floating-point exception bitfields in FPSR.

FP_STATE

Floating point state. Fields:

FPSR unsigned int

FPSR register value.

MASK unsigned int

The mask indicating updated cumulative bits.

FREQ_CHANGED

Core frequency changed. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The instruction count of this core.

NEW_FREQ signed int

The new frequency of this core (expressed in Hz).

OLD_FREQ signed int

The old frequency of this core (expressed in Hz).

HLT

HLT instruction occurred. Fields:

IMM16 unsigned int

The 16-bit immediate value encoded in HLT instruction.

INST

Every instruction executed. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DEBUG_STATE bool

The instruction is executed in debug state.

DISASS string

Disassembly of instruction.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISSET enum

The current instruction set.

ITSTATE unsigned int

The current ITSTATE.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

MODE enum

The mode the core is in.

NS unsigned int

The core's non-secure bit.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

OPCODE unsigned int

The opcode of the instruction.

PADDR unsigned int

The physical address of the instruction.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the instruction.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the page.

PC unsigned int

The address of the instruction.

SECURITY_STATE enum

The core's security state.

SIZE unsigned int

The size of the instruction in bytes.

INST_END

Every instruction completed.

INST_START

Every instruction started. Fields:

DEBUG_STATE bool

The instruction is executed in debug state.

ISSET enum

The current instruction set.

MODE enum

The mode the core is in.

NS enum

The current Secure State.

PC unsigned int

The address of the conditional instruction.

SECURITY_STATE enum

The current Security State.

INST_STRADDLE

Instruction straddles boundary. Fields:

DEBUG_STATE bool

The instruction is executed in debug state.

ISSET enum

The current instruction set.

MODE enum

The mode the core is in.

NS enum

The current Secure State.

PC unsigned int

The address of the conditional instruction.

SECURITY_STATE enum

The current Security State.

IRQ_TAKEN

IRQ taken exception.

ISB

ISB instruction occurred. Fields:

PC unsigned int

The address of the ISB instruction.

LOAD_MULTIPLE_REGS_LIST

List of destination registers (not source) of a multiple registers load instruction. Fields:

REG string

Register to add into the list.

LOCAL_MONITOR

Local monitor activity. Fields:

PADDR unsigned int

Local Monitor Address.

State enum

State of the monitor (Open/Exclusive).

MAPPING_CREATE

A memory mapping has been created. This is an implementation detail of the model and so may change without notice. Fields:

ASID unsigned int

The ASID used during translation, or 0 if no ASID was used.

IS_VALID bool

Whether the mapping is valid or not.

PADDR unsigned int

Base physical address of region, or 0 if not known.

SIDE enum

I-side or D-side.

SIZE unsigned int

Size of the page, or 0 if not known.

VADDR unsigned int

Base virtual address of region.

VMID unsigned int

The VMID used during translation, or 0 if no VMID was used.

MAPPING_DESTROY

A memory mapping has been destroyed. This is an implementation detail of the model and so may change without notice. Fields:

ASID unsigned int

The ASID used during translation.

CAUSE enum

The cause of invalidation.

PADDR unsigned int

Base physical address of page.

SIDE enum

I-side or D-side.

SIZE unsigned int

Size of the page.

VADDR unsigned int

Base virtual address of page.

VMID unsigned int

The VMID used during translation.

MEMMAP_DEBUG_READ

Memory mapped reads to the debug registers. Fields:

ADDR unsigned int

address.

CORE_NUM unsigned int

Core number in a multi processor.

EXT bool

Whether access is from an external device (such as the DAP).

NS enum

Secure state of the access.

REG_NAME string

Name of the debug register accessed.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

MEMMAP_DEBUG_WRITE

Memory mapped writes to the debug registers. Fields:

ADDR unsigned int

address.

CORE_NUM unsigned int

Core number in a multi processor.

EXT bool

Whether access is from an external device (such as the DAP).

NS enum

Secure state of the access.

REG_NAME string

Name of the debug register accessed.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The new value written.

MMU_D128_TTB_READ

This event is triggered by reads caused by a D128 translation table walk. Fields:

ABORTED bool

Was the walk successful.

DATA_HIGH unsigned int

The data upper bits read.

DATA_LOW unsigned int

The data lower bits read.

IPA unsigned int

For stage 1, the IPA of the read.

LEVEL unsigned int

Translation table level.

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

OUTADDR unsigned int

The output physical address.

PADDR unsigned int

The physical address of the read.

PASmem enum

Physical Address Space of the accessed memory.

PASreq enum

Physical Address Space of the requested lookup.

REGIME_EL enum

Entry matches in this translation regime.

REQUESTER enum

The requester of this table walk.

SIDE enum

Inst / Data.

STAGE unsigned int

Translation stage.

MMU_D128_TTB_WRITE

This event is triggered by writes caused by a D128 translation table walk. Fields:

ABORTED bool

Was the walk successful.

DATA_HIGH unsigned int

The data upper bits written.

DATA_LOW unsigned int

The data lower bits written.

IPA unsigned int

For stage 1, the IPA of the write.

IS_HACDBS_ACCESS bool

If the access was due to HACDBS_CLEAN.

LEVEL unsigned int

Translation table level.

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

OUTADDR unsigned int

The output physical address.

PADDR unsigned int

The physical address of the write.

PASmem enum

Physical Address Space of the accessed memory.

PASreq enum

Physical Address Space of the requested lookup.

REGIME_EL enum

Entry matches in this translation regime.

SIDE enum

Inst / Data.

STAGE unsigned int

Translation stage.

MMU_TRANS

Address translation information. Fields:

ASID unsigned int

Address space identifier.

CORE_NUM unsigned int

Core number in a multi processor.

Hyp bool

Entry matches in Hyp state only.

INNERCACHE_RA bool

Is the inner cache allocate on read.

INNERCACHE_TRANSIENT bool

Is the inner write-through transient.

INNERCACHE_TYPE enum

Inner Caching scheme (NC/MB/WA).

INNERCACHE_WA bool

Is the inner cache allocate on write.

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MEMTYPE enum

Memory type.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC enum

Is secure side supposed to access secure or nonsecure memory.

OUTERCACHE_RA bool

Is the outer cache allocate on read.

OUTERCACHE_TRANSIENT bool

Is the outer write-through transient.

OUTERCACHE_TYPE enum

Outer Caching scheme (NC/MB/WA).

OUTERCACHE_WA bool

Is the outer cache allocate on write.

PADDR unsigned int

Physical address of the access.

PAGESIZE unsigned int

Page size as log2(size).

PAS enum

PAS supposed to be accessed is secure or nonsecure memory or Root/Realm if RME is implemented.

SH enum

Shareability.

SIDE enum

Inst / Data.

STAGE1_PERM unsigned int

Stage 1 permissions, [5:3] Privileged access (XWR) [2:0] User access (XWR).

STAGE2_PERM unsigned int

Stage 2 permission mask [5:3] Privileged access (XWR) [2:0] User access (XWR).

VADDR unsigned int

Virtual address of the access.

VMID unsigned int

Virtual machine identifier.

XS bool

XS Attribute.

nG enum

Flag indicating whether ASID will be matched.

MMU_TTB_READ

This event is triggered by reads caused by a translation table walk. Fields:

ABORTED bool

Was the walk successful.

DATA unsigned int

The data read.

DESC_KIND enum

The encoding scheme of the descriptors used for this walk.

IPA unsigned int

For stage 1, the IPA of the read.

LEVEL unsigned int

Translation table level.

LPAE bool

Is this for an LPAE translation (*DEPRECATED*, use DESC_KIND).

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSmem bool

Non secure memory access.

NSreq bool

Non secure request.

OUTADDR unsigned int

The output physical address.

PADDR unsigned int

The physical address of the read.

PASmem enum

Physical Address Space of the accessed memory.

PASreq enum

Physical Address Space of the requested lookup.

REGIME_EL enum

Entry matches in this translation regime.

REQUESTER enum

The requester of this table walk.

SIDE enum

Inst / Data.

STAGE unsigned int

Translation stage.

MMU_TTB_WRITE

This event is triggered by writes caused by a translation table walk. Fields:

ABORTED bool

Was the walk successful.

DATA unsigned int

The data written.

DESC_KIND enum

The encoding scheme of the descriptors used for this walk.

IPA unsigned int

For stage 1, the IPA of the write.

IS_HACDBS_ACCESS bool

If the access was due to HACDBS_CLEAN.

LEVEL unsigned int

Translation table level.

LPAE bool

Is this for an LPAE translation (*DEPRECATED*, use DESC_KIND).

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSmem bool

Non secure memory access.

NSreq bool

Non secure request.

OUTADDR unsigned int

The output physical address.

PADDR unsigned int

The physical address of the write.

PASmem enum

Physical Address Space of the accessed memory.

PASreq enum

Physical Address Space of the requested lookup.

REGIME_EL enum

Entry matches in this translation regime.

SIDE enum

Inst / Data.

STAGE unsigned int

Translation stage.

MODE_CHANGE

Mode change. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

MODE enum

The new mode.

NON_SECURE enum

The core's new non-secure bit.

OLD_MODE enum

The old mode.

SECURITY_STATE enum

The new security state.

NOT_TAG_CHECKED_INST

MTE Tag not checked for current instruction.

PERIODIC

Called for every quantum. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The instruction count of this CPU.

PC unsigned int

The address of the next instruction to be executed on this CPU.

PMU_COUNTER_OVERFLOW

PMU counter overflow. Fields:

COUNTER_GROUP enum

Counter group.

COUNTER_RANGE enum

Counter range.

EVENT_ID unsigned int

Event Identifier, field is valid only for Event counter group.

INDEX unsigned int

Counter index (as selected by PMSELR), field is valid only for Event/Cycle counter groups.

INTERRUPT bool

Is interrupt enabled on overflow for this counter.

POTINDEX_UPDATE

Value of POT index for current pc changed. Fields:

NEW_POTINDEX unsigned int

New POTIndex.

OLD_POTINDEX unsigned int

Old POTIndex.

PREFETCH_MEMORY64

Prefetch from PRFM or PRFUM instructions. Fields:

PRFOP enum

Prefetch hint.

VADDR unsigned int

Virtual address of the location that should be prefetched.

PRELOAD64_REQUEST

Preload requests from PRFM or PRFUM instructions. Fields:

PRFOP enum

Prefetch hint.

VADDR unsigned int

Virtual address of the location that should be prefetched.

PRELOAD_DATA

Data preload from PLD instruction. Fields:

VADDR unsigned int

Virtual address of the data that should be preloaded.

PRELOAD_INST

Instruction preload from PLI instruction. Fields:

VADDR unsigned int

Virtual address of the instruction that should be preloaded.

PRE_CORE_LOAD

Trace just before a core load. Fields:

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

PRE_CORE_STORE

Trace just before a core store. Fields:

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

PRE_TTB_READ

This event is triggered before reads caused by a translation table walk. Fields:

ADDR unsigned int

The physical address of the read.

PSTATE

PSTATE fields update. Fields:

ALLINT unsigned int

All interrupt mask bit.

BTYPE unsigned int

Branch target identification bit.

CORE_NUM unsigned int

Core number in a multi processor.

DAIF unsigned int

Exception mask bits.

DIT unsigned int

Data Independent Timing bit.

EL unsigned int

Current Exception level field.

EXLOCK unsigned int

Exception return state lock bit.

IL unsigned int

Illegal Execution state bit.

NZCV unsigned int

Condition flags.

PACM unsigned int

PACM bit.

PAN unsigned int

Privileged Access Never state bit.

PM unsigned int

PMU exception mask bit.

PPEND unsigned int

PMU exception pending bit.

SM unsigned int

StreamingMode status bit.

SP unsigned int

Stack pointer register selection bit.

SS unsigned int

Software Step bit.

SSBS unsigned int

Speculative Store Bypass Safe bit.

TCO unsigned int

Tag Check Override bit.

TOUCHED unsigned int

Mask of the fields that have been touched since this trace has last been emitted. Note that touched does not necessarily mean the value changed. The index of the field is the bit index in the mask plus the index of this field plus one.

UAO unsigned int

User Access Override bit.

UINJ unsigned int

Undefined Instruction exceptions bit.

ZA unsigned int

ZA storage and ZT0 register enable bit.

nRW unsigned int

Current Execution state bit.

PSTATE_PACTIVE_OPMODE_TRANSITION

Trace active and Opmode mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Operating mode after the transition.

OLD_STATE enum

Operating mode before the transition.

PSTATE_PACTIVE_OPMODE_TRANSITION_DENIED

Trace the denied active and Opmode mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Operating mode after the transition.

OLD_STATE enum

Operating mode before the transition.

REASON string

Reason for the transition being denied.

PSTATE_PACTIVE_POWER_TRANSITION

Trace active and power mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Power mode after the transition.

OLD_STATE enum

Power mode before the transition.

PSTATE_PACTIVE_POWER_TRANSITION_DENIED

Trace the denied active and power mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Power mode after the transition.

OLD_STATE enum

Power mode before the transition.

REASON string

Reason for the transition being denied.

PSTATE_PACTIVE_THREAD_OPMODE_TRANSITION

Trace pactive and Opmode mode transitions for threads on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Operating mode after the transition.

OLD_STATE enum

Operating mode before the transition.

QUANTUM_START

Called at the start of every quantum.

RANGE_PRELOAD_REQUEST

Preload requests from RPRFM. Fields:

COUNT unsigned int

Number of blocks of data to access.

LENGTH unsigned int

Number of contiguous bytes to be accessed in each block.

OP unsigned int

Range prefetch operation .

REUSE unsigned int

Maximum number of bytes to be accessed before next RPRFM .

STRIDE signed int

Stride to advance after accessing LENGTH bytes.

VADDR unsigned int

Base virtual address of range prefetch request.

RUN_STATE

Run state transition. Fields:

INST_COUNT unsigned int

Ticks count at point of transition.

NEW enum

New run state.

OLD enum

Old run state.

SEMIHOSTING_CALL

Call of a semihost function occurred. Fields:

PC unsigned int

The program counter after the semihosting call.

SEMIHOSTING_PRECALL

About to call semihost. Fields:

EL2 **bool**

Use EL2 translation regime.

NS **bool**

Is Non-Secure.

REG_WIDTH **unsigned int**

The current register width in bytes.

SIGNAL

External signal state change. Fields:

SIGNAL **enum**

Signal that changed.

STATE **bool**

Signal asserted state.

SOFTWARE_STEP

Return the Debug Software Step state. Fields:

STATE **enum**

Software Step state.

SPSR

SPSR change. Fields:

CORE_NUM **unsigned int**

Core number in a multi processor.

MODE **enum**

Which of the banked SPSR registers is written.

OLD_VALUE **unsigned int**

The old SPSR value.

VALUE **unsigned int**

The new SPSR value.

START_COMPILE

Compilation started. Fields:

VADDR **unsigned int**

Instruction where compilation begins.

STG_MULTIPLE_MICRO_OPS_ADDRESS_RANGE

Address range for MTE stores with multiple uops. Fields:

ADDR_END **unsigned int**

Final address.

ADDR_START unsigned int

Starting address.

STORE_MULTIPLE_REGS_ADDRESS_RANGE

Address range for stores with multiple registers. Fields:

ADDR_END unsigned int

Final address.

ADDR_START unsigned int

Starting address.

SUBOPTIMAL_LDST_RETIRED

suboptimal load/store. Fields:

ACCESS_SIZE unsigned int

Log2 of the access width in bytes used for alignment checking (1: half-word, 2: word ...).

OPTIMAL_ALIGN_SELECTOR unsigned int

Selector to decide the optimal alignment size (1: access width, 2: 4-bytes, 3: 8-bytes, ..., 12: 4-kbytes).

VADDR unsigned int

The virtual address of the access.

SYNC

Called for every synchronization, that is every quantum break. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The instruction count of this CPU.

LOCAL_QUANTUM unsigned int

The local quantum of this CPU.

LOCAL_TIME unsigned int

The local time of this CPU.

SYNC_REASON enum

Reason for sync.

SYSCALL

System call instruction executed. Fields:

IMM unsigned int

Immediate value of the system call instruction.

TYPE enum

System call type.

VADDR unsigned int

Instruction that caused the system call.

SYSREG_READ128

System Coprocessor 128 bit register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

REGNUM enum

Internal register number.

REGNUM_PHYS enum

Internal register number physically accessed (e.g. in presense of VHE redirection).

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_READ64

System Coprocessor register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REGNUM enum

Internal register number.

REGNUM_OPERAND enum

Register number corresponding to the instruction operand.

REGNUM_PHYS enum

Internal register number physically accessed (e.g. in presense of VHE redirection).

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_UPDATE128

Triggers when the system updates a register. Fields:

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG enum

Register number.

UNKNOWN unsigned int

Bits of the register which became unknown.

VALUE unsigned int

Value written to the register.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_UPDATE32

Triggers when the system updates a register. Fields:

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG enum

Register number.

UNKNOWN unsigned int

Bits of the register which became unknown.

VALUE unsigned int

Value written to the register.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_UPDATE64

Triggers when the system updates a register. Fields:

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG enum

Register number.

UNKNOWN unsigned int

Bits of the register which became unknown.

VALUE unsigned int

Value written to the register.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int
opcode 2.

SYSREG_WRITE128

System Coprocessor 128 bit register write. Fields:

CORE_NUM unsigned int
Core number in a multi processor.

CRm unsigned int
CRm.

CRn unsigned int
CRn.

REGNUM enum
Internal register number.

REGNUM_PHYS enum
Internal register number physically accessed (e.g. in presense of VHE redirection).

UNDEF bool
The register accessed is undefined.

UPDATED_VALUE unsigned int
Updated value of the register now it has been written.

VALUE unsigned int
The new value written.

opc unsigned int
opcode 1.

opc0 unsigned int
opcode 0.

opc2 unsigned int
opcode 2.

SYSREG_WRITE64

System Coprocessor register write. Fields:

CORE_NUM unsigned int
Core number in a multi processor.

CRm unsigned int
CRm.

CRn unsigned int
CRn.

NS bool
Secure or nonsecure banked register is accessed.

REGNUM enum

Internal register number.

REGNUM_OPERAND enum

Register number corresponding to the instruction operand.

REGNUM_PHYS enum

Internal register number physically accessed (e.g. in presense of VHE redirection).

UNDEF bool

The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

UNALIGNED_LDST_RETIRED

Processor unaligned load/store. Fields:

ACCESS_SIZE unsigned int

Log2 of the access width in bytes used for alignment checking (1=>half-word, 2->word ...).

NUMBER_OF_BEATS unsigned int

Number of accesses (beats) in this burst (Total size in bytes is ACCESS_SIZE * NUMBER_OF_BEATS).

VADDR unsigned int

The virtual address of the access.

VFP_D_REGS

VFP/NEON D 64 bit register write. Fields:

ALIASING enum

Type of register aliasing used.

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a write to the corresponding bit in the register. For example, 0x1 would indicate a write to VALUE[0].

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

VFP_D_REGS_READ

VFP/NEON D 64 bit register read. Fields:

ID unsigned int

The register number.

VALUE unsigned int

The current value of the register.

VFP_Q_REGS

VFP/NEON Q 128 bit register write. Fields:

ALIASING enum

Type of register aliasing used.

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a write to the corresponding byte in the register. For example, 0x1 would indicate a write to VALUE[7:0].

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

VFP_Q_REGS_READ

VFP/NEON Q 128 bit register read. Fields:

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a read to the corresponding byte in the register. For example, 0x1 would indicate a read to VALUE[7:0].

VALUE unsigned int

The current value of the register.

VFP_SYS_REGS

Writes to the VFP/NEON units system registers. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ID enum

Which VFP system register is written.

MASK unsigned int

The mask indicating updated cumulative bits.

OLD_VALUE unsigned int

The register's old value overwritten.

VALUE unsigned int

The new value written to the VFP system register.

VFP_S_REGS

VFP/NEON S 32 bit register write. Fields:

ALIASING enum

Type of register aliasing used.

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a write to the corresponding bit in the register. For example, 0x1 would indicate a write to VALUE[0].

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

VFP_S_REGS_READ

VFP/NEON S 32 bit register read. Fields:

ID unsigned int

The register number.

VALUE unsigned int

The current value of the register.

VIRTUAL_TAG_ALLOCATED

Virtual tag information when allocated. Fields:

TAG_VADDR unsigned int

Virtual tag virtual address.

VADDR unsigned int

Data Virtual address.

VIRTUAL_TAG unsigned int

Virtual tag value.

WAYPOINT

Signals end of basic block execution. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ISET enum

Origin instruction set.

IS_COND bool

Indicates if this is a conditional waypoint.

PC unsigned int

Origin address (or 0 if unavailable).

TAKEN bool

Indicates if this waypoint was taken.

TARGET unsigned int

Destination address.

TARGET_ISET enum

Destination instruction set.

WFE_END

WFE ended. Fields:

INST_COUNT unsigned int

Ticks count when leaving WFE.

WFE_EVENT_REGISTER

WFE event register status: set/clear, reason. Fields:

INST_COUNT unsigned int

Ticks count.

REASON enum

Reason for set/clear. Only REASON==0 (Cleared by WFE) clears the bit, all other reasons set it.

WFE_IGNORED

WFE ignored. Fields:

EVENT bool

This WFE was ignored because the event register was set.

INST_COUNT unsigned int

Ticks count when ignoring WFE.

TRAPPED bool

This WFE was trapped.

WFE_START

WFE entered. Fields:

INST_COUNT unsigned int

Ticks count when entering WFE.

WFI_END

WFI ended. Fields:

INST_COUNT unsigned int

Ticks count when leaving WFI.

WFI_IGNORED

WFI ignored. Fields:

DISABLED bool

This WFI was ignored because WFI is disabled.

INST_COUNT unsigned int

Ticks count when ignoring WFI.

REASON enum

specifies reason why WFI trace was ignored.

TRAPPED bool

This WFI was trapped.

WFI_START

WFI entered. Fields:

INST_COUNT unsigned int

Ticks count when entering WFI.

WFI_WAKEUP

WFI wakeup. Fields:

INST_COUNT unsigned int

Ticks count when WFI wakeup occurred.

REASON enum

Reason for wakeup.

2.21 ARM_Cortex-A57

This section describes the trace sources.

AA64_ASE_SVE_REGS

Modification of AArch64 Advanced SIMD or SVE Vector Registers (via SIMD or Scalar operations), SVE Predicate Registers and ZT0. Fields:

ID `enum`

The register identifier.

MASK `unsigned int`

Bitmask of the register to signal the modified bits in the VALUE field.

SM `bool`

Whether the PE is in Streaming Mode.

VALUE `unsigned int`

Value of AArch64 vector register.

ASYNC_MEMORY_FAULT

Context ID Register write. Fields:

FAULT `unsigned int`

Fault status present in ESR.ISS encoding for Serror in ARCH64 and present in DFSR encoding for external abort format in ARCH32.

PADDR `unsigned int`

Physical Address (or 0 if unavailable).

VADDR `signed int`

Virtual Address (or 0 if unavailable).

ATOMIC_END_ACCESS

Bus trace access for atomics. Fields:

ACCESS_FAIL `bool`

Memory access failed.

ADDR `unsigned int`

The virtual address of the access.

ATTR `unsigned int`

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

COMPARE_VALUE `unsigned int`

Compare value for CAS.

LOAD_VALUE unsigned int

Loaded values.

NSDESC unsigned int

The security state of the access.

OPERAND_VALUE unsigned int

Operation's operand.

OPERATION enum

Operation type.

PADDR unsigned int

The physical address of the access.

PRIV bool

Is this a privileged access?.

ATOMIC_START_ACCESS

Bus trace access for atomics. Fields:

ADDR unsigned int

The virtual address of the access.

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

COMPARE_VALUE unsigned int

Compare value for CAS.

NSDESC unsigned int

The security state of the access.

OPERAND_VALUE unsigned int

Operation's operand.

OPERATION enum

Operation type.

PADDR unsigned int

The physical address of the access.

PRIV bool

Is this a privileged access?.

ArchMsg.Error.exit_code

DISPLAY exit code: %{EXITCODE} component kind: %{KIND} reason: %{REASON}. Fields:

EXITCODE signed int

Exit code to be returned.

KIND string

Component kind that invoked the exit code trace.

REASON string

Reason why the exit code trace was invoked (optional).

ArchMsg.Warning.IgnoredWarmResetRequest

Event fires when software attempts to request warm reset through write to deprecated DBGPRCR.CWRR.

ArchMsg.Warning.ReportedRevisionRegisterRead

DISPLAY Reported revision and patch do not match that of the cpu. Fields:

ORIGINAL_VALUE unsigned int

The value the register would have based on the modelled revision.

REGISTER string

The name of the register that has ben read.

REPORTED_VALUE unsigned int

The value the register has been overridden to.

ArchMsg.Warning.Unpredictably Indexed PM Event Register

TAGS **UNPREDICTABLE** DISPLAY Event %{IsDirect:(Selector value|Register)} %{SEL} is >= %{N}. Fields:

IsDirect unsigned int

Direct Access rather than indirect.

N unsigned int

Number of Accessible Registers.

SEL unsigned int

Selector.

ArchMsg.Warning.cache_contents_unknown

A cache has been activated without being invalidated and may contain **UNKNOWN** content. DISPLAY %{SIDE:(D|I)}-Cache was enabled in %{REGIME:(SECURE|NON_SECURE|HYP)} regime but was not invalidated since power-on: cache lines could contain **UNKNOWN** content %{SIDE: (Note: D-side caches must invalidate SECURE lines even if the cache is only used by NON_SECURE code in order to ensure that unexpected dirty lines tagged SECURE are not naturally evicted)}. Fields:

REGIME unsigned int

Current translation regime.

SIDE unsigned int

0 d-side cache or unified cache, 1 i-side.

ArchMsg.Warning.dcimvac_matches_watchpoint

TAGS **UNPREDICTABLE** DISPLAY Watchpoints matching an AArch32 DCIMVA are implementation defined.

ArchMsg.Warning.decode_fieldmismatch

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with different fields %{FIELD1} and %{FIELD2}. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_fracbitsnegative

TAGS **UNPREDICTABLE** DISPLAY VCVT Instruction is unpredictable with a negative number of fraction bits.

ArchMsg.Warning.decode_initblock

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable in an IT block.

ArchMsg.Warning.decode_initblocknotlast

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable in an IT block when not the last.

ArchMsg.Warning.decode_invalid_condition

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable when the condition is not AL.

ArchMsg.Warning.decode_invalidfieldcombination

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with the values in %{FIELD1} and %{FIELD2}. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_invalidvalue

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with the value in %{FIELD}. Fields:

FIELD enum

Field name.

ArchMsg.Warning.decode_mem_hint_unallocated

TAGS **UNPREDICTABLE** DISPLAY There is no memory hint allocated to this bit pattern.

ArchMsg.Warning.decode_registermatch

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with %{FIELD1} and %{FIELD2} as the same register. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_registermismatch

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with %{FIELD1} and %{FIELD2} as different registers. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_registeroutofrange

TAGS **UNPREDICTABLE** DISPLAY Floating-point instruction is unpredictable indexing beyond the end of the register bank.

ArchMsg.Warning.decode_sbzsbo

TAGS **UNPREDICTABLE** DISPLAY Reserved bits in the instruction are not canonical.

ArchMsg.Warning.decode_transactiontoobig

TAGS **UNPREDICTABLE** DISPLAY Attempt to perform a memory transaction of more than 128 bytes.

ArchMsg.Warning.decode_unpred_other

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable.

ArchMsg.Warning.decode_unpreduseofpc

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable when %{FIELD} is PC. Fields:

FIELD enum

Field name.

ArchMsg.Warning.decode_unpreduseofr13

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable when %{FIELD} is R13. Fields:

FIELD enum

Field name.

ArchMsg.Warning.decode_unpreduseofr13inreglist

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with R13 in register-list.

ArchMsg.Warning.decode_useofd16orhigher

TAGS **UNPREDICTABLE** DISPLAY Attempt to use register D16 or higher where it is not allowed.

ArchMsg.Warning.decode_writebackandbaseinlist

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with write back when the base register is in the transfer list.

ArchMsg.Warning.decode_zeroregistersinlist

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable when no registers are to be transferred.

ArchMsg.Warning.recursive_branch

An instruction is performing a branch that targets the same instruction. If this was intended then a WFI might be more effective. DISPLAY recursive branch detected at PC=%{PC}.

PRIMARY KEY PC. Fields:

PC unsigned int

Address of instruction causing the branch.

ArchMsg.Warning.recursive_exception

An instruction has generated an exception that targets the same instruction. DISPLAY recursive %{TYPE} exception detected at PC=%{PC} (CPSR=%{CPSR}). PRIMARY KEY PC. Fields:

CPSR unsigned int

Processor state.

PC unsigned int

Address of instruction causing the exception.

TYPE enum

Type of exception.

ArchMsg.Warning.reentrant_vector_catch

TAGS **UNPREDICTABLE** DISPLAY Vector catch debug event on Prefetch or Data abort vector %{ADDR}. Fields:

ADDR unsigned int

vector address.

ArchMsg.Warning.reserved_it_state

TAGS **UNPREDICTABLE** DISPLAY Execution with reserved IT state %{ITSTATE}. Fields:

ITSTATE unsigned int

reserved IT state.

ArchMsg.Warning.tlb_contents_unknown

Translation has been enabled without invalidating the TLB. DISPLAY Translation was enabled in the %{PAS:(Secure|Non-secure|Root|Realm)} EL%{EL:d} translation regime (vmid=%{VMID:x}) without invalidating %{INVALIDITY:(|d-side |i-side |)}entries since power-on: %{INVALIDITY:(|D||I)}TLB could contain **UNKNOWN** content. Fields:

EL unsigned int

Exception level of translation regime.

INVALIDITY unsigned int

Which TLB may be invalid (if TLBs are separate).

NON_SECURE bool

Whether the translation regime is non-secure.

PAS unsigned int

Physical address space of translation regime.

VMID unsigned int

Current VMID, if applicable.

ArchMsg.Warning.unknown_DLR_DSPSR

Exiting debug state with unknown DLR or DSPSR.

ArchMsg.Warning.unknown_ELR_SPSR

Returning from debug exception with unknown ELR or SPSR.

ArchMsg.Warning.unpred_FAR_on_LRCPC3_non_atomic_mem_access

TAGS **UNPREDICTABLE** Unpredictable FAR/FSTATUS on LRCPC3 pair instructions resulting in two separate memory accesses
DISPLAY Unpredictable FAR on LRCPC3 non atomic memory access at %{ADDR1} with FSTATUS %{FSTATUS1} or %{ADDR2} with FSTATUS %{FSTATUS2}. Fields:

ADDR1 unsigned int

Possible FAR address 1.

ADDR2 unsigned int

Possible FAR address 2.

FSTATUS1 unsigned int

Possible fault status value 1.

FSTATUS2 unsigned int

Possible fault status value 2.

ArchMsg.Warning.unpred_async_tag_check_fail_on_sve_speculative_load_do_update_ffr

Unpredictable set to FALSE of the FFR element associated to SVE speculative load with an async tag check fail TAGS **UNPREDICTABLE**.

ArchMsg.Warning.unpredictable_a32_breakpoint

TAGS **UNPREDICTABLE** DISPLAY Unpredictable breakpoint on a A32 instruction at %{ADDR1} Offset %{OFFSET} BAS %{BAS} for a %{IS_ADDRESS_MISMATCH_BREAKPOINT:(Mismatch|Match)} Breakpoint number %{N:d} Breakpoint address %{ADDR2}. Fields:

ADDR1 unsigned int

Address.

ADDR2 unsigned int

Breakpoint address.

BAS unsigned int

BAS field.

IS_ADDRESS_MISMATCH_BREAKPOINT bool

Is a Mismatch breakpoint, not a Match breakpoint.

N unsigned int

Breakpoint number.

OFFSET unsigned int

Always has a value of zero.

ArchMsg.Warning.unpredictable_banked_register_access

TAGS **UNPREDICTABLE** DISPLAY Access to banked register (R=%{R:d} SYSm=%{SYSM:2x}) is unpredictable. Fields:

R unsigned int

R bit.

SYSM unsigned int

SYSm.

ArchMsg.Warning.unpredictable_ccfail_undef

TAGS **UNPREDICTABLE** DISPLAY An instruction %{OPCODE} at VA=%{VADDR} failed its condition codes check but would otherwise have caused a non-data-dependent trap or undefined exception. Behaviour is implementation-defined choice of exception or nop. Fields:

OPCODE unsigned int

opcode of the instruction.

VADDR unsigned int

Virtual address of the instruction being executed.

ArchMsg.Warning.unpredictable_ici_bits

TAGS **UNPREDICTABLE** DISPLAY ICI bits do not apply or are architecturally unpredictable for instruction.

ArchMsg.Warning.unpredictable_t32_breakpoint

TAGS **UNPREDICTABLE** DISPLAY Unpredictable breakpoint on a T32 instruction at %{ADDR1} Offset %{OFFSET} BAS %{BAS} for a %{IS_ADDRESS_MISMATCH_BREAKPOINT:(Mismatch|Match)} Breakpoint number %{N:d} Breakpoint address %{ADDR2}. Fields:

ADDR1 unsigned int

Address.

ADDR2 unsigned int

Breakpoint address.

BAS unsigned int

BAS field.

IS_ADDRESS_MISMATCH_BREAKPOINT bool

Is a Mismatch breakpoint, not a Match breakpoint.

N unsigned int

Breakpoint number.

OFFSET unsigned int

Indicates which halfword of a 32-bit Thumb instruction the breakpoint is set on. 0 for the first halfword or if the instruction is 16 bits wide, or 1 for the second halfword.

ArchMsg.Warning.unpredictable_t32_vector_catch

TAGS **UNPREDICTABLE** DISPLAY Unpredictable Vector catch exception on a T32 instruction at %{ADDR} Offset %{OFFSET}. Fields:

ADDR unsigned int

Address.

OFFSET unsigned int

Indicates which halfword of a 32-bit Thumb instruction the vector catch exception is set on. 0 for the first halfword or if the instruction is 16 bits wide, or 1 for the second halfword.

ArchMsg.Warning.unpredictable_watchpoint_far

TAGS **UNPREDICTABLE** DISPLAY The value for FAR/EDWAR for hitting this watchpoint may be anywhere in the range %{LOWER_BOUND} to %{UPPER_BOUND}. Fields:

LOWER_BOUND unsigned int

the lowest address accessed by the instruction that triggered the watchpoint.

UPPER_BOUND unsigned int

the highest watchpointed address accessed by that instruction.

ArchMsg.Warning.unpredictable_watchpoint_far_isscmp

DISPLAY The value for FAR/EDWAR for hitting this watchpoint may be anywhere in the range %{LOWER_BOUND} to %{UPPER_BOUND}. Fields:

LOWER_BOUND unsigned int

the lowest address accessed by the instruction that triggered the watchpoint.

UPPER_BOUND unsigned int

the highest watchpointed address accessed by that instruction.

ArchMsg.Warning.warning_AdvSIMDExpandImmUnexpectedZero

TAGS **UNPREDICTABLE** DISPLAY AdvSIMDExpandImm may treat this immediate value as **UNPREDICTABLE**.

ArchMsg.Warning.warning_ConditionalSMC

TAGS **UNPREDICTABLE** DISPLAY SMC instruction has **UNPREDICTABLE** effects when conditional and when combined with traps.

ArchMsg.Warning.warning_TLBID_out_of_range

TLBID out of range in TLBI according to the configuration of NIS, NVIS, NOS and NVOS. Fields:

NIS unsigned int

NIS.

NOS unsigned int

NOS.

NVIS unsigned int

NVIS.

NVOS unsigned int

NVOS.

TLBID unsigned int

The TLBID value used in the TLBI instruction.

ArchMsg.Warning.warning_TTL_or_TG_mismatches_current_translation_granule

TTL or TG mismatches current translation granule in TLBI VA/RVA/IPA/RIPA. Fields:

CURRENT_TG unsigned int

Current TG as indicated by TCR_ELx or VTCR_EL2.

TLBI_TG unsigned int

TG (or TTL[3:2]) used in TLBI instruction.

ArchMsg.Warning.warning_access_crosses_page_boundary

TAGS **UNPREDICTABLE** DISPLAY Access crosses page boundary.

ArchMsg.Warning.warning_access_crosses_page_boundary_has_fault

TAGS **UNPREDICTABLE** DISPLAY Access that crosses page boundary has fault(s).

ArchMsg.Warning.warning_access_crosses_page_boundary_has_fault_on_both_pages

TAGS **UNPREDICTABLE** DISPLAY Access that crosses page boundary has a fault on each page.
Fields:

ADDR1 unsigned int

FAR address on first page.

ADDR2 unsigned int

FAR address on second page.

FSTATUS1 unsigned int

Fault status on first page.

FSTATUS2 unsigned int

Fault status on second page.

ArchMsg.Warning.warning_access_crosses_page_boundary_spanning_different_memory

TAGS **UNPREDICTABLE** DISPLAY Access at address %{ADDR} crosses page from MEM_TYPE %{MEMTYPE_PAGE1} to %{MEMTYPE_PAGE2}. Fields:

ADDR unsigned int

address of access crossing page.

MEMTYPE_PAGE1 signed int

First Page's memory type (0-Normal, 1-Non-Normal).

MEMTYPE_PAGE2 signed int

Second Page's memory type (0-Normal, 1-Non-Normal).

ArchMsg.Warning.warning_access_wraps_around_memory

TAGS **UNPREDICTABLE** DISPLAY Access wraps around memory in %{WIDTH:d}bit mode.
Fields:

WIDTH unsigned int

Address width.

ArchMsg.Warning.warning_bcr_linking_status

TAGS **UNPREDICTABLE** DISPLAY DBGBCR.{BT2,BT} combination gives unpredictable behaviour.

ArchMsg.Warning.warning_bcr_mask_reserved

TAGS **UNPREDICTABLE** DISPLAY DBGBCR_ELn.MASK is set to a reserved value.

ArchMsg.Warning.warning_bvr_masked_bit_not_zero

TAGS **UNPREDICTABLE** DISPLAY DBGBVRn_EL1.VA contains a bit masked by DBGBCRn_EL1.MASK that is not zero.

ArchMsg.Warning.warning_ccsidr_unimplemented_level

TAGS **UNPREDICTABLE** DISPLAY CCSIDR read while CSSELR %{CSSELR:x} points to an unimplemented cache level. Fields:

CSSELR unsigned int

current effective value of CSSELR.

ArchMsg.Warning.warning_change_to_ns_when_tge_set

TAGS **UNPREDICTABLE** DISPLAY Attempting to change NS to 1 when HCR.TGE = 1.

ArchMsg.Warning.warning_contiguous_bit_check_abort

TAGS **UNPREDICTABLE** DISPLAY An abort occurred whilst attempting to check TLB entry is contiguous at %{ENTRY_ADDR}. Fields:

ENTRY_ADDR unsigned int

address of conflicting TLB entry.

ArchMsg.Warning.warning_contiguous_bit_error

TAGS **UNPREDICTABLE** DISPLAY TLB entry at %{ENTRY_ADDR} was not contiguous with entry at %{CONTIG_ADDR} Contents are %{ENTRY_DATA} but expected %{CONTIG_DATA}. Fields:

CONTIG_ADDR unsigned int

address of the TLB first read, with which this entry is expected to be contiguous.

CONTIG_DATA unsigned int

expected contents based on the entry at CONTIG_ADDR.

ENTRY_ADDR unsigned int

address of conflicting TLB entry.

ENTRY_DATA unsigned int

contents of conflicting TLB entry.

ArchMsg.Warning.warning_contiguous_bit_error_D128

TAGS **UNPREDICTABLE** DISPLAY TLB entry at %{ENTRY_ADDR} was not contiguous with entry at %{CONTIG_ADDR} Contents are %{ENTRY_DATA_HI} %{ENTRY_DATA_LO} but expected %{CONTIG_DATA}. Fields:

CONTIG_ADDR unsigned int

address of the TLB first read, with which this entry is expected to be contiguous.

CONTIG_DATA unsigned int

expected contents based on the entry at CONTIG_ADDR.

ENTRY_ADDR unsigned int

address of conflicting TLB entry.

ENTRY_DATA_HI unsigned int

Higher bit contents of conflicting TLB entry.

ENTRY_DATA_LO unsigned int

Lower bit contents of conflicting TLB entry.

ArchMsg.Warning.warning_cp10_cp11_mismatch

TAGS **UNPREDICTABLE** DISPLAY Unpredictable configuration of CP10 and CP11 controls at EL
%{EL:d}. Fields:

EL unsigned int

EL owning the control.

ArchMsg.Warning.warning_cp10_cp11_reserved_value

TAGS **UNPREDICTABLE** DISPLAY Unpredictable configuration of CPACR CP10 (%{CP10}) or
CP11 (%{CP11}). Fields:

CP10 unsigned int

CP10 access.

CP11 unsigned int

CP11 access.

ArchMsg.Warning.warning_csselr_level_out_of_range

TAGS **UNPREDICTABLE** DISPLAY CSSELR.Level out of range, written value %{LEVEL} with only
%{IMPLEMENTED:d} levels implemented. Fields:

IMPLEMENTED unsigned int

number of cache levels implemented.

LEVEL unsigned int

written value of CSSELR.Level.

TYPE enum

The type of cache selected by CSSELR.InD and TnD.

ArchMsg.Warning.warning_debug_flow_control_bits_not_obeyed

TAGS **UNPREDICTABLE** DISPLAY Flow control bits for %{REGISTER} were not obeyed, causing
%{ERROR}. Fields:

ERROR enum

type of error resulting.

REGISTER enum

register under consideration.

ArchMsg.Warning.warning_debug_register_access_during_reset

TAGS **UNPREDICTABLE** DISPLAY Unpredictable %{IS_WRITE:(read|write)} access to debug register offset %{OFFSET} during reset. Fields:

IS_WRITE unsigned int

Write Not Read.

OFFSET unsigned int

Register Offset.

ArchMsg.Warning.warning_decode_cps_inconsistent_fields

TAGS **UNPREDICTABLE** DISPLAY The mode setting fields in the CPS instruction are inconsistent.

ArchMsg.Warning.warning_decode_invalid_state

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable in current exception-level/security state.

ArchMsg.Warning.warning_default_cacheable_mmu_on

TAGS **UNPREDICTABLE** DISPLAY Default cacheable is enabled (HCR.DC = 1) while MMU is enabled.

ArchMsg.Warning.warning_default_cacheable_vmmu_off

TAGS **UNPREDICTABLE** DISPLAY Default cacheable is enabled (HCR.DC = 1) while HCR.VM = 0.

ArchMsg.Warning.warning_deprecated_wvr_bit_2_set

TAGS **UNPREDICTABLE** DISPLAY DBGWVR%{N:d}_EL1=%{WVR:h} has bit 2 set, deprecated in ARMv8. Fields:

N unsigned int

index.

WVR unsigned int

DBGWVR.

ArchMsg.Warning.warning_eret_while_software_step_active_pending

Missing ISB between setting MDSCR_EL1.SS and ERET TAGS **UNPREDICTABLE**.

ArchMsg.Warning.warning_exclusive_to_non_normal

TAGS **UNPREDICTABLE** DISPLAY Exclusive Access to Non-Normal Memory %{ADDR}. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_exclusive_to_non_writeback

TAGS **UNPREDICTABLE** DISPLAY Exclusive Access to Non-Writeback-Cacheable Memory %{ADDR}. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_execute_from_device_memory

TAGS **UNPREDICTABLE** DISPLAY An attempt was made to execute from Device memory.

ArchMsg.Warning.warning_illegal_cpsr_mode

TAGS **UNPREDICTABLE** DISPLAY Writing an illegal or unimplement mode ({NEW_MODE:h}) to CPSR was unpredictable. Fields:

NEW_MODE unsigned int

New value of CPSR.M.

ArchMsg.Warning.warning_illegal_srs_mode

TAGS **UNPREDICTABLE** DISPLAY Illegal or **UNPREDICTABLE** mode ({MODE:h}) used for SRS instruction. Fields:

MODE unsigned int

mode for Banked SP.

ArchMsg.Warning.warning_implementation_defined_read_debug_register_in_SCS

Read Access to Debug registers from the processor is **IMPLEMENTATION DEFINED** TAGS **IMP_DEF**. Fields:

PPB_OFFSET unsigned int

debug register offset.

ArchMsg.Warning.warning_implementation_defined_sequential_security_transitions_supported

The behavior of sequential instruction fetches that cross from non-secure to secure memory and contain an SG instruction is **CONSTRAINED UNPREDICTABLE** TAGS **UNPREDICTABLE**.

ArchMsg.Warning.warning_implementation_defined_stack_limit_check_supported

It is **IMPLEMENTATION DEFINED** whether stack pointer limit checking is performed for this instruction TAGS **IMP_DEF**. Fields:

ADDRESS unsigned int

Address of instruction.

OPCODE unsigned int

instruction opcode.

ArchMsg.Warning.warning_implementation_defined_taggednonsharedmemorytype

TAGS **IMP_DEF** DISPLAY Memory attribute has Tagged, non shared memory type; it is **IMPLEMENTATION DEFINED** whether the memory location is treated as Tagged or non-Tagged.

ArchMsg.Warning.warning_implementation_defined_write_debug_register_in_SCS

Write Access to Debug registers from the processor is **IMPLEMENTATION DEFINED** TAGS **IMP_DEF**. Fields:

DATA unsigned int

data attempted to be written.

PPB_OFFSET unsigned int

debug register offset.

ArchMsg.Warning.warning_invalid_tcr_granule

TAGS UNPREDICTABLE DISPLAY TCR.TG{%TG_ID:(0|1)} bits have been set to a granule size {%REQUEST:d}K not implemented - using {%SUBSTITUTE:d}K. Fields:

REQUEST signed int

page size requested (or 0 for reserved).

SUBSTITUTE signed int

best guess available page size.

TG_ID bool

bits TG0 or TG1.

ArchMsg.Warning.warning_load_multiple_user_registers_from_user_mode

TAGS UNPREDICTABLE DISPLAY An LDM(user registers) instruction executed from user mode.

ArchMsg.Warning.warning_mte_in_mem_attr_not_supported

TAGS UNPREDICTABLE DISPLAY Memory attribute {%ATTR} has Tagged memory type but Armv8.5-MTE is not implemented. Fields:

ATTR unsigned int

Memory attribute.

ArchMsg.Warning.warning_reserved_breakpoint_state_match

TAGS UNPREDICTABLE DISPLAY {%IS_BREAKPOINT:(Watchpoint|Breakpoint)} programmed with a reserved combination of HMC, SSC and {%IS_BREAKPOINT:(PAC|PMC)}. Fields:

IS_BREAKPOINT unsigned int

Is a breakpoint, not a watchpoint.

ArchMsg.Warning.warning_shareability

TAGS UNPREDICTABLE DISPLAY Unpredictable: combination of the 1st and 2nd stages of translation is Normal Inner Non-Cacheable, Outer Non-Cacheable.

ArchMsg.Warning.warning_software_step_set_while_enabled

TAGS UNPREDICTABLE DISPLAY MDSCR_EL1.SS set to 1 while software step debug exceptions are enabled.

ArchMsg.Warning.warning_thumb_instruction_wraps_around_memory

TAGS UNPREDICTABLE DISPLAY Thumb instruction wraps around memory.

ArchMsg.Warning.warning_ttbr_sbz_bits_are_not_zero

TAGS UNPREDICTABLE DISPLAY {%TTBR}[{%MSB:d}:0] should be zero, but are not. Fields:

MSB unsigned int

x-1.

TTBR enum

which TTBR register.

ArchMsg.Warning.warning_unaligned_address_dbgdtrrx_write

TAGS **UNPREDICTABLE** DISPLAY Unaligned address %{ADDR} in X0/R0 when DBGDTRRX written in memory access mode. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unaligned_address_dbgdtrtx_read

TAGS **UNPREDICTABLE** DISPLAY Unaligned address %{ADDR} in X0/R0 when DBGDTRTX read in memory access mode. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unaligned_to_device

TAGS **UNPREDICTABLE** DISPLAY Unaligned Access to Device Memory %{ADDR}. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unaligned_to_strongly_ordered

TAGS **UNPREDICTABLE** DISPLAY Unaligned Access to Strongly Ordered Memory %{ADDR}. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unknown_sau_rnr

TAGS **UNPREDICTABLE** DISPLAY SAU_RNR was set to an unsupported value.

ArchMsg.Warning.warning_unpredictable_AIRCR_PRIS_and_BFHFNMINs

TAGS **UNPREDICTABLE** DISPLAY The effect of setting both AIRCR.BFHFNMINs and AIRCR.PRIS to 1 is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_AIRCR_VECTCLRACTIVE_when_not_in_debug

TAGS **UNPREDICTABLE** DISPLAY The effect of writing a 1 to AIRCR.VECTCLRACTIVE if the processor is not halted in Debug state is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_AIRCR_VECTRESET_and_SYSRESETREQ

TAGS **UNPREDICTABLE** DISPLAY When the processor is halted in Debug state, if a write to the register writes a 1 to both VECTRESET and SYSRESETREQ, the behavior is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_AIRCR_VECTRESET_when_not_in_debug

TAGS **UNPREDICTABLE** DISPLAY The effect of writing a 1 to AIRCR.VECTRESET if the processor is not halted in Debug state is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_AIRCR_incorrect_VKEY

TAGS **UNPREDICTABLE** DISPLAY The value 0x05FA must be written to AIRCR.VKEY, otherwise the register write is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_EXC_RETURN_Reserved_Bit

EXC_RETURN[23:7] are reserved with the special condition that all bits should be written as one. Values other than all 1s are **UNPREDICTABLE**. TAGS **UNPREDICTABLE**. Fields:

EXC_RETURN unsigned int

Special PC value.

SBO_MASK unsigned int

Which bits Should Be One to not UNPRED.

ArchMsg.Warning.warning_unpredictable_PMEVTYPEPn_ELO_VS_reserved_value

TAGS **UNPREDICTABLE** DISPLAY Unpredictable reserved value of PMEVTYPEN<n>_ELO.VS.

ArchMsg.Warning.warning_unpredictable_change_to_DEMCR_MON_STEP_at_insufficient_priority

TAGS **UNPREDICTABLE** DISPLAY The effect of changing DEMCR.MON_STEP at an execution priority that is lower than the priority of the DebugMonitor exception is **CONSTRAINED UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_change_to_priority_of_active_exception

TAGS **UNPREDICTABLE** DISPLAY Changing the priority of an active exception is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_clear_lspact

TAGS **UNPREDICTABLE** DISPLAY **UNPREDICTABLE** state on context switch. Fields:

DRVEXC_TAKEN unsigned int

Derived exception taken, not pended.

REACHED_LAST_STORE unsigned int

Only the last faulted.

ArchMsg.Warning.warning_unpredictable_dbg_exit_unaligned_dlr

TAGS **UNPREDICTABLE** DISPLAY Debug exit to AArch32 with DLR[0] set to 1. PC[0] can be set to 0 or to DLR[0].

ArchMsg.Warning.warning_unpredictable_exception_catch

TAGS **UNPREDICTABLE** DISPLAY Generation of an exception catch event is unpredictable.

ArchMsg.Warning.warning_unpredictable_exception_return_inconsistent_state

TAGS **UNPREDICTABLE** DISPLAY state on exception return is unpredictable.

ArchMsg.Warning.warning_unpredictable_exception_return_instruction

TAGS **UNPREDICTABLE** DISPLAY exception return instruction is unpredictable.

ArchMsg.Warning.warning_unpredictable_in_debug_state

TAGS **UNPREDICTABLE** DISPLAY Instruction is **UNPREDICTABLE** when executed in debug state.

ArchMsg.Warning.warning_unpredictable_mair_encoding

TAGS **UNPREDICTABLE** DISPLAY Memory attribute %{ATTR} has invalid MAIR encoding; this may lead to unpredictable results. Fields:

ATTR unsigned int

Memory attribute.

ArchMsg.Warning.warning_unpredictable_pmu_counter_access

TAGS **UNPREDICTABLE** DISPLAY Access to PMU counters from non-secure EL0 or EL1 is **UNPREDICTABLE** with MDCR_EL2.HPMN set to 0.

ArchMsg.Warning.warning_unpredictable_prioritization_breakpoint_match_vector_catch

TAGS **UNPREDICTABLE** DISPLAY Two events with the same priority occurred at the same instruction: Address Matching Vector Catch debug event, and Breakpoint debug event. It is constrained unpredictable which is taken.

ArchMsg.Warning.warning_unpredictable_stack_selection

TAGS **UNPREDICTABLE** DISPLAY **UNPREDICTABLE** state on context switch.

ArchMsg.Warning.warning_unpredictable_tsize_out_of_range

TAGS **UNPREDICTABLE** DISPLAY %{TCR}.T%{TTBR:d}SZ (%{TSIZE:d} bits) is out of range. Must be between 25 and %{TSIZE_MAX:d}. Fields:

TCR enum

Which TCR.

TSIZE signed int

Value of TSize.

TSIZE_MAX signed int

The maximum allowed value.

TTBR unsigned int

Which TTBR region.

ArchMsg.Warning.warning_unpredictable_unaligned_pc_as_base_register

TAGS **UNPREDICTABLE** DISPLAY The PC value must be word-aligned, otherwise the behavior of the instruction is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_unaligned_pop_stack

TAGS **UNPREDICTABLE** DISPLAY **CONSTRAINED UNPREDICTABLE** unaligned stack frame %{FRAME_ADDR:x} in exception-return. Fields:

FRAME_ADDR unsigned int

unaligned stack frame address.

ArchMsg.Warning.warning_unpredictable_unaligned_to_non_writeback

TAGS **UNPREDICTABLE** DISPLAY Unaligned Access to Non-Writeback-Cacheable Memory.

ArchMsg.Warning.warning_unpredictable_vmsa_memattrib

TAGS **UNPREDICTABLE** DISPLAY Unpredictable vmsa memory attribute with texcb value programmed to %{TEXCB} when tex remap is %{TEX_REMAP}. Fields:

TEXCB unsigned int

texcb value.

TEX_REMAP bool

use tex remap.

ArchMsg.Warning.warning_unpredictable_vtcr_t0sz_sl0

TAGS **UNPREDICTABLE** DISPLAY The combination of VTCR.TOSZ=%{TOSZ:d} and VTCR.SLO=%{SLO:d} is unpredictable. Fields:

SLO signed int

VTCR.SLO, starting level for stage 2 translation table walks.

TOSZ signed int

VTCR.TOSZ, required input address range.

ArchMsg.Warning.warning_unpredictable_write_DHCSR

TAGS **UNPREDICTABLE** DISPLAY **UNPREDICTABLE** set of bit-changes in DHCSR. Fields:

NEW unsigned int

data attempted to be written.

OLD unsigned int

previous value.

ArchMsg.Warning.warning_unpredictable_write_SHCSR

TAGS **UNPREDICTABLE** DISPLAY **UNPREDICTABLE** setting a pending bit to 1 for an exception with priority >= execution priority.

ArchMsg.Warning.warning_unsupported_access_to_memory_mapped_register

TAGS **UNPREDICTABLE** DISPLAY Access to memory mapped register is unsupported.

ArchMsg.Warning.warning_user_jmcr_access

TAGS **UNPREDICTABLE** DISPLAY User %{WRITE:(write|read)} access to JMCR is unpredictable. Fields:

WRITE unsigned int

access is write.

ArchMsg.Warning.warning_wcr_mask_and_bas

TAGS **UNPREDICTABLE** DISPLAY DBGWCR_EL1.MASK is non-zero and BAS is not 0xff.

ArchMsg.Warning.warning_wcr_mask_reserved

TAGS **UNPREDICTABLE** DISPLAY DBGWCR_ELn.MASK is set to a reserved value.

ArchMsg.Warning.warning_wcr_non_configuous_bas

TAGS **UNPREDICTABLE** DISPLAY DBGWCR_EL1.BAS has non-contiguous set of ones.

ArchMsg.Warning.warning_wvr_masked_bit_not_zero

TAGS **UNPREDICTABLE** DISPLAY DBGWVRn_EL1.VA contains a bit masked by DBGWCRn_EL1.MASK that is not zero.

ArchMsg.Why.why_branch_target_exception

A branch target exception has occurred. DISPLAY branch target exception: target pc=%{PC}, target opcode=%{OPCODE}. PRIMARY KEY PC. Fields:

BTYPE unsigned int

pstate.BTYPE at time branch taken.

OPCODE unsigned int

opcode at branch target.

PC unsigned int

Virtual Address of branch target.

ArchMsg.Why.why_illegal_state

An illegal mode change has occurred. DISPLAY illegal mode change: %{MESSAGE}. Fields:

MESSAGE string

The message.

BRANCH_MISPREDICT

Simulating branch mispredict. Fields:

PC unsigned int

Origin address (or 0 if unavailable).

BRA_DIR

Direct branches, to immediate address. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISSET enum

The instruction set of the branch instruction.

IS_COND bool

Indicates if this is a conditional branch.

IS_HINTED bool

Whether the conditional instruction is hinted.

IS_ISB bool

Whether the instruction is ISB.

IS_LINK bool

Indicates if this is a branch with link.

PC unsigned int

The address of the branch instruction.

TARGET_ISET enum

The instruction set after the branch.

TARGET_PC unsigned int

The address the instruction branches to.

BRA_INDIR

Indirect branches, perhaps to a register. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISET enum

The instruction set of the branch instruction.

IS_COND bool

Indicates if this is a conditional branch.

IS_HINTED bool

Whether the conditional instruction is hinted.

IS_ISB bool

Whether the instruction is ISB.

IS_LINK bool

Indicates if this is a branch with link.

PC unsigned int

The address of the branch instruction.

TARGET_ISET enum

The instruction set after the branch.

TARGET_PC unsigned int

The address the instruction branches to.

BRA_INDIR_CCFAIL

Indirect branches fail, perhaps to a register.

BRA_RET_CCFAIL

Branch return fail, perhaps to a register.

CACHE_MAINTENANCE_OP

Cache Maintenance Operation. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

DATA unsigned int

Specified MVA or Set/way.

FUNCTION enum

Clean / Invalidate.

SCOPE enum

Affected region.

SIDE enum

Inst / Data.

CCFAIL

Conditional instruction condition check fail. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CCFAIL_UNC

Unconditional instruction condition check fail. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CCPASS

Conditional instruction condition check pass. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CCPASS_UNC

Unconditional instruction condition check pass. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CODE_CACHE_MAINT

Code cache maintenance. Fields:

END_INCL unsigned int

Inclusive end address for VA or PA requests.

NS bool

Non-secure world for PA.

START unsigned int

Start address for VA or PA requests.

TYPE enum

Request type.

COMPILE_BLOCK_END

Last instruction of basic block translated. Fields:

VADDR unsigned int

Address of next instruction after this basic block.

COMPILE_BLOCK_START

First instruction of basic block translated. Fields:

VADDR unsigned int

Address of first instruction in basic block.

COMPILE_INST

ARM instruction compiled. Fields:

DISASS string

Disassembly of the instruction.

ISSET enum

The instruction set of this instruction.

ITSTATE unsigned int

The ITSTATE current for the instruction.

OPCODE unsigned int

The opcode of the instruction.

PC unsigned int

The address of the instruction.

SIZE unsigned int

The size of the instruction in bytes.

CONTEXTIDR

Context ID Register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

NS bool

Secure or nonsecure banked register is accessed.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The new value written.

CONTEXT_SYNC

Called for every context synchronization event. Fields:

ARCH_CSE bool

Architectural Context Synchronization Event.

PC unsigned int

The address of the synchronization instruction.

SYSREG_SYNC bool

System register synchronization event.

CORE_ENDIAN

Core BE8 Big-Endian state changed. Fields:

BE8 bool

Core BE8 Big-Endian state.

CORE_INFO

Static processor attributes. Only triggered by a call to DumpState(). Fields:

ARCH_PROFILE enum

The architecture profile of the core.

CLUSTER_ID unsigned int

The cluster ID of this processor.

CORE_NUM unsigned int

The number of this core in an MP processor.

DETAILED_CORE_FEATURES enum

Detailed core features as a bitfield. Enum values are bitmasks to extract the info from the field.

FPU_VERSION enum

The VFP version implemented by the core.

MEM_ARCH enum

The memory architecture of the core.

MIDR_EL1 unsigned int

MIDR_EL1 value.

NUM_CORES unsigned int

The number of cores in this MP processor.

QUANTUM_SIZE unsigned int

The default quantum size of the core (in ticks).

SECURITY_FEATURES bool

Does the core have security features?.

CORE_LOADS

Processor load accesses. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ACQREL enum

Is this an acquire/release.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DATA unsigned int

The data read or written.

ELEMENT_SIZE unsigned int

Width of each element.

ENDIANNESS enum

Byte ordering of this access.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

LOCK enum

Normal, exclusive or locked access.

MECID unsigned int

Memory Encryption Context ID, if applicable.

MECID2 unsigned int

Memory Encryption Context ID of the second page, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PAGE_SIZE unsigned int

Page size as $\log_2(\text{size in bytes})$.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK, 2=Exclusive Failed, 3=Unsupported.

SIZE unsigned int

Width of the access in bytes. Only required if DATA is not traced.

TRANS bool

Is this a translated access.

VADDR unsigned int

The virtual address of the access.

CORE_REGS

Changes of the core registers R0 to R14. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number, 0 to 14.

MODE enum

Bank of the register accessed.

OLD_VALUE unsigned int

The old value overwritten.

PHYS_ID enum

The physical register accessed.

VALUE unsigned int

The new value written to the register.

CORE_REGS64

Changes of the core registers X0..X30, SP_ELn. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ID enum

The register number, 0-30 for X0-X30, >=32 for SP_ELn.

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

CORE_REGS64_READ

General purpose AArch64 register has been read X0..X30, SP_ELn. Fields:

ID enum

The register number, 0-30 for X0-X30, >=32 for SP_ELn.

VALUE unsigned int

The value read from the register.

CORE_STALLED_TRANSITION

Detect when a core is prevented from progressing PChannel states. Fields:

CURRENT_STATE enum

Current power mode of core.

REASON string

The reason a transition was stalled, and what must be done to unblock it.

TARGET_STATE enum

Desired power mode of core after transition.

CORE_STORES

Processor store accesses. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ACQREL enum

Is this an acquire/release.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DATA unsigned int

The data read or written.

ELEMENT_SIZE unsigned int

Width of each element.

ENDIANNESS enum

Byte ordering of this access.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

LOCK enum

Normal, exclusive or locked access.

MECID unsigned int

Memory Encryption Context ID, if applicable.

MECID2 unsigned int

Memory Encryption Context ID of the second page, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PAGE_SIZE unsigned int

Page size as $\log_2(\text{size in bytes})$.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK, 2=Exclusive Failed, 3=Unsupported.

SIZE unsigned int

Width of the access in bytes. Only required if DATA is not traced.

TRANS bool

Is this a translated access.

VADDR unsigned int

The virtual address of the access.

CP14_READ

System Coprocessor register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP14 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc1 unsigned int

opcode 1.

opc2 unsigned int

opcode 2.

CP14_WRITE

System Coprocessor register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP14 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc1 unsigned int

opcode 1.

opc2 unsigned int

opcode 2.

CP15_READ

System Coprocessor register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP15 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc1 unsigned int

opcode 1.

opc2 unsigned int

opcode 2.

CP15_READ64

System Coprocessor register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP15 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc unsigned int

opcode 1.

CP15_WRITE

System Control Coprocessor register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP15 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc1 unsigned int

opcode 1.

opc2 unsigned int

opcode 2.

CP15_WRITE64

System Coprocessor register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP15 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc unsigned int

opcode 1.

CPSR

CPSR change. (In AArch64 PSTATE represented in SPSR format). Fields:

CORE_NUM unsigned int

Core number in a multi processor.

OLD_VALUE unsigned int

The old CPSR value.

UNKNOWN unsigned int

Bits within the register that have unknown value.

VALUE unsigned int

The new CPSR value.

CRYPTO_SPEC

Every crypto instruction speculatively executed.

CorePowerStateChange

Core power-up state has changed. DISPLAY Core powered %{PU:(up|down)}. Fields:

PU bool

Whether the core powered up.

DATA_CACHE_SET_ALLOC_TAGS

Set allocation tags from DC GVA instruction. Fields:

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

NSDESC unsigned int

The security state of the access.

PADDR unsigned int

The physical address of the access.

PAS enum

The physical address space of the page.

RESPONSE enum

0=Aborted, 1=OK.

SIZE unsigned int

The size of memory in bytes.

TAG unsigned int

The value of the tag being set.

TAGGING_TYPE enum

Type of Memory tagging (Physical or Virtual).

TAG_VADDR unsigned int

The virtual address of the Virtual Tag if Virtual tagging is used; otherwise 0.

VADDR unsigned int

Virtual address used for setting allocation tags.

DATA_CACHE_ZERO

Zero and invalidate cache line from DC ZVA instruction. Fields:

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

NSDESC unsigned int

The security state of the access.

PADDR unsigned int

The physical address of the access.

PADDR_TAG unsigned int

If tag checked, the physical address tag for the access.

PAS enum

The physical address space of the page.

RESPONSE enum

0=Aborted, 1=OK.

SIZE unsigned int

The size of zero'd memory in bytes.

TAG_CHECKED bool

Is this access tag checked.

VADDR unsigned int

Virtual address of the zero'd cache line.

DEBUG_EVENT

Hardware debug support event. Fields:

EVENT enum

Description of event.

VALUE unsigned int

data value.

DEBUG_TARGET_ISET

Hardware debug target ISA. Fields:

TARGET_ISET enum

The target instruction set of the debug state.

DMI_ALLOCATE

DMI allocated. Fields:

ACCESS_DIRECTION enum

READ / WRITE.

PADDR unsigned int

The physical address of the DMI.

PTR unsigned int

DMI range pointer.

SIDE enum

DSIDE / ISIDE.

SIZE unsigned int

The size of DMI.

VADDR unsigned int

The virtual address of the DMI..

DMI_HIT

DMI hit, access used DMI (only fire for small DMI regions i.e. less than a full page). Fields:

ACCESS_DIRECTION enum

READ / WRITE.

PADDR unsigned int

The physical address of the DMI.

SIDE enum

DSIDE / ISIDE.

SIZE unsigned int

The size of the DMI.

VADDR unsigned int

The virtual address of the DMI..

DMI_REVOKE

DMI revoked. Fields:

ACCESS_DIRECTION enum

READ / WRITE.

PADDR unsigned int

The physical address of the DMI.

SIDE enum

DSIDE / ISIDE.

SIZE unsigned int

The size of DMI.

VADDR unsigned int

The virtual address of the DMI..

END_COMPILE

Compilation end. Fields:

END_OF_PAGE_COUNT unsigned int

Number of basic blocks exited due to page boundary since START_COMPILE.

FETCHFAIL_COUNT unsigned int

Number of basic blocks exited due to ifetch failure START_COMPILE.

INST_COUNT unsigned int

Number of instructions compiled since START_COMPILE.

NONSEQ_COUNT unsigned int

Number of basic blocks exited due to non-sequential instructions since START_COMPILE.

PAGE_STRADDLE_COUNT unsigned int

Number of basic blocks exited due to unaligned instructions crossing page since START_COMPILE.

ESB

Error Synchronization Barrier hit. Fields:

ASYNC bool

ESB is async.

IMPLICIT bool

ESB is implicit.

EXCEPTION

Exception taken. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ESR unsigned int

The value assigned to ESR when exception is taken.

ESR_EC unsigned int

The Value assigned to ESR.EC when exception is taken.

IS_PHYSICAL bool

Physical or Virtual exception.

IS_TRAP bool

Exception coming from a trap.

LR unsigned int

The value assigned to the link register.

PC unsigned int

The location where the exception occurred.

PREFERRED_RETURN unsigned int

The preferred return address for the exception.

TAKEN_LOCALLY bool

Whether exception is taken locally(same exception level).

TARGET_ISET enum

The instruction set of the exception handler code.

TARGET_PC unsigned int

The address the exception branches to.

VECTOR enum

The exception vector.

VECTOR_OFFSET unsigned int

The offset of an exception vector from the base address.

EXCEPTION_END

Every exception completed.

EXCEPTION_INFO

Trace human-readable reason of the current exception for data/instruction aborts, floating point exceptions and unknown exceptions. Fields:

DESCRIPTION string

Reason for current exception.

ESR_EC unsigned int

The value of ESR.EC when exception is taken.

EXCEPTION_RAISE

Every exception raised.

EXCEPTION_RETURN

Branches on leaving exception. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISET enum

The instruction set of the branch instruction.

IS_COND bool

Indicates if this is a conditional branch.

IS_HINTED bool

Whether the conditional instruction is hinted.

IS_ISB bool

Whether the instruction is ISB.

IS_LINK bool

Indicates if this is a branch with link.

PC unsigned int

The address of the branch instruction.

TARGET_ISET enum

The instruction set after the branch.

TARGET_PC unsigned int

The address the instruction branches to.

EXCEPTION_RETURN_PREBRANCH

Exception return event that must occur before PC and processor state have updated.

EXCEPTION_START

Every exception started.

FIQ_TAKEN

FIQ taken exception.

FP_EXCEPTIONS_SIGNALLED

Trace which FP exceptions were signalled as a part of the FP operations performed by an instruction. Fields:

FP_EXCEPTIONS_SIGNALLED_MASK unsigned int

Indicates which FP exceptions were signalled by an instruction. This uses the same encoding as the cumulative floating-point exception bitfields in FPSR.

FP_STATE

Floating point state. Fields:

FPSR unsigned int

FPSR register value.

MASK unsigned int

The mask indicating updated cumulative bits.

FREQ_CHANGED

Core frequency changed. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The instruction count of this core.

NEW_FREQ signed int

The new frequency of this core (expressed in Hz).

OLD_FREQ signed int

The old frequency of this core (expressed in Hz).

HLT

HLT instruction occurred. Fields:

IMM16 unsigned int

The 16-bit immediate value encoded in HLT instruction.

INST

Every instruction executed. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DEBUG_STATE bool

The instruction is executed in debug state.

DISASS string

Disassembly of instruction.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISSET enum

The current instruction set.

ITSTATE unsigned int

The current ITSTATE.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

MODE enum

The mode the core is in.

NS unsigned int

The core's non-secure bit.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

OPCODE unsigned int

The opcode of the instruction.

PADDR unsigned int

The physical address of the instruction.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the instruction.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the page.

PC unsigned int

The address of the instruction.

SECURITY_STATE enum

The core's security state.

SIZE unsigned int

The size of the instruction in bytes.

INST_END

Every instruction completed.

INST_START

Every instruction started. Fields:

DEBUG_STATE bool

The instruction is executed in debug state.

ISET enum

The current instruction set.

MODE enum

The mode the core is in.

NS enum

The current Secure State.

PC unsigned int

The address of the conditional instruction.

SECURITY_STATE enum

The current Security State.

INST_STRADDLE

Instruction straddles boundary. Fields:

DEBUG_STATE bool

The instruction is executed in debug state.

ISET enum

The current instruction set.

MODE enum

The mode the core is in.

NS enum

The current Secure State.

PC unsigned int

The address of the conditional instruction.

SECURITY_STATE enum

The current Security State.

IRQ_TAKEN

IRQ taken exception.

ISB

ISB instruction occurred. Fields:

PC unsigned int

The address of the ISB instruction.

LOAD_MULTIPLE_REGS_LIST

List of destination registers (not source) of a multiple registers load instruction. Fields:

REG string

Register to add into the list.

LOCAL_MONITOR

Local monitor activity. Fields:

PADDR unsigned int

Local Monitor Address.

State enum

State of the monitor (Open/Exclusive).

MAPPING_CREATE

A memory mapping has been created. This is an implementation detail of the model and so may change without notice. Fields:

ASID unsigned int

The ASID used during translation, or 0 if no ASID was used.

IS_VALID bool

Whether the mapping is valid or not.

PADDR unsigned int

Base physical address of region, or 0 if not known.

SIDE enum

I-side or D-side.

SIZE unsigned int

Size of the page, or 0 if not known.

VADDR unsigned int

Base virtual address of region.

VMID unsigned int

The VMID used during translation, or 0 if no VMID was used.

MAPPING_DESTROY

A memory mapping has been destroyed. This is an implementation detail of the model and so may change without notice. Fields:

ASID unsigned int

The ASID used during translation.

CAUSE enum

The cause of invalidation.

PADDR unsigned int

Base physical address of page.

SIDE enum

I-side or D-side.

SIZE unsigned int

Size of the page.

VADDR unsigned int

Base virtual address of page.

VMID unsigned int

The VMID used during translation.

MEMMAP_DEBUG_READ

Memory mapped reads to the debug registers. Fields:

ADDR unsigned int

address.

CORE_NUM unsigned int

Core number in a multi processor.

EXT bool

Whether access is from an external device (such as the DAP).

NS enum

Secure state of the access.

REG_NAME string

Name of the debug register accessed.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

MEMMAP_DEBUG_WRITE

Memory mapped writes to the debug registers. Fields:

ADDR **unsigned int**

address.

CORE_NUM **unsigned int**

Core number in a multi processor.

EXT **bool**

Whether access is from an external device (such as the DAP).

NS **enum**

Secure state of the access.

REG_NAME **string**

Name of the debug register accessed.

UNDEF **bool**

The register accessed is undefined.

VALUE **unsigned int**

The new value written.

MMU_D128_TTB_READ

This event is triggered by reads caused by a D128 translation table walk. Fields:

ABORTED **bool**

Was the walk successful.

DATA_HIGH **unsigned int**

The data upper bits read.

DATA_LOW **unsigned int**

The data lower bits read.

IPA **unsigned int**

For stage 1, the IPA of the read.

LEVEL **unsigned int**

Translation table level.

MECID **unsigned int**

Memory Encryption Context(MEC) ID, if applicable.

MPAM_PARTID **unsigned int**

MPAM Partition ID.

MPAM_PMG **unsigned int**

MPAM Performance Monitoring Group.

MPAM_SP **unsigned int**

MPAM Partition ID Space.

OUTADDR unsigned int

The output physical address.

PADDR unsigned int

The physical address of the read.

PASmem enum

Physical Address Space of the accessed memory.

PASreq enum

Physical Address Space of the requested lookup.

REGIME_EL enum

Entry matches in this translation regime.

REQUESTER enum

The requester of this table walk.

SIDE enum

Inst / Data.

STAGE unsigned int

Translation stage.

MMU_D128_TTB_WRITE

This event is triggered by writes caused by a D128 translation table walk. Fields:

ABORTED bool

Was the walk successful.

DATA_HIGH unsigned int

The data upper bits written.

DATA_LOW unsigned int

The data lower bits written.

IPA unsigned int

For stage 1, the IPA of the write.

IS_HACDBS_ACCESS bool

If the access was due to HACDBS_CLEAN.

LEVEL unsigned int

Translation table level.

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

OUTADDR unsigned int

The output physical address.

PADDR unsigned int

The physical address of the write.

PASmem enum

Physical Address Space of the accessed memory.

PASreq enum

Physical Address Space of the requested lookup.

REGIME_EL enum

Entry matches in this translation regime.

SIDE enum

Inst / Data.

STAGE unsigned int

Translation stage.

MMU_TRANS

Address translation information. Fields:

ASID unsigned int

Address space identifier.

CORE_NUM unsigned int

Core number in a multi processor.

Hyp bool

Entry matches in Hyp state only.

INNERCACHE_RA bool

Is the inner cache allocate on read.

INNERCACHE_TRANSIENT bool

Is the inner write-through transient.

INNERCACHE_TYPE enum

Inner Caching scheme (NC/MB/WA).

INNERCACHE_WA bool

Is the inner cache allocate on write.

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MEMTYPE enum

Memory type.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC enum

Is secure side supposed to access secure or nonsecure memory.

OUTERCACHE_RA bool

Is the outer cache allocate on read.

OUTERCACHE_TRANSIENT bool

Is the outer write-through transient.

OUTERCACHE_TYPE enum

Outer Caching scheme (NC/MB/WA).

OUTERCACHE_WA bool

Is the outer cache allocate on write.

PADDR unsigned int

Physical address of the access.

PAGESIZE unsigned int

Page size as log2(size).

PAS enum

PAS supposed to be accessed is secure or nonsecure memory or Root/Realm if RME is implemented.

SH enum

Shareability.

SIDE enum

Inst / Data.

STAGE1_PERM unsigned int

Stage 1 permissions, [5:3] Privileged access (XWR) [2:0] User access (XWR).

STAGE2_PERM unsigned int

Stage 2 permission mask [5:3] Privileged access (XWR) [2:0] User access (XWR).

VADDR unsigned int

Virtual address of the access.

VMID unsigned int

Virtual machine identifier.

XS bool

XS Attribute.

nG enum

Flag indicating whether ASID will be matched.

MMU_TTB_READ

This event is triggered by reads caused by a translation table walk. Fields:

ABORTED bool

Was the walk successful.

DATA unsigned int

The data read.

DESC_KIND enum

The encoding scheme of the descriptors used for this walk.

IPA unsigned int

For stage 1, the IPA of the read.

LEVEL unsigned int

Translation table level.

LPAE bool

Is this for an LPAE translation (*DEPRECATED*, use DESC_KIND).

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSmem bool

Non secure memory access.

NSreq bool

Non secure request.

OUTADDR unsigned int

The output physical address.

PADDR unsigned int

The physical address of the read.

PASmem enum

Physical Address Space of the accessed memory.

PASreq enum

Physical Address Space of the requested lookup.

REGIME_EL enum

Entry matches in this translation regime.

REQUESTER enum

The requester of this table walk.

SIDE enum

Inst / Data.

STAGE unsigned int

Translation stage.

MMU_TTB_WRITE

This event is triggered by writes caused by a translation table walk. Fields:

ABORTED bool

Was the walk successful.

DATA unsigned int

The data written.

DESC_KIND enum

The encoding scheme of the descriptors used for this walk.

IPA unsigned int

For stage 1, the IPA of the write.

IS_HACDBS_ACCESS bool

If the access was due to HACDBS_CLEAN.

LEVEL unsigned int

Translation table level.

LPAE bool

Is this for an LPAE translation (*DEPRECATED*, use DESC_KIND).

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSmem bool

Non secure memory access.

NSreq bool

Non secure request.

OUTADDR unsigned int

The output physical address.

PADDR unsigned int

The physical address of the write.

PASmem enum

Physical Address Space of the accessed memory.

PASreq enum

Physical Address Space of the requested lookup.

REGIME_EL enum

Entry matches in this translation regime.

SIDE enum

Inst / Data.

STAGE unsigned int

Translation stage.

MODE_CHANGE

Mode change. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

MODE enum

The new mode.

NON_SECURE enum

The core's new non-secure bit.

OLD_MODE enum

The old mode.

SECURITY_STATE enum

The new security state.

NOT_TAG_CHECKED_INST

MTE Tag not checked for current instruction.

PERIODIC

Called for every quantum. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The instruction count of this CPU.

PC unsigned int

The address of the next instruction to be executed on this CPU.

PMU_COUNTER_OVERFLOW

PMU counter overflow. Fields:

COUNTER_GROUP enum

Counter group.

COUNTER_RANGE enum

Counter range.

EVENT_ID unsigned int

Event Identifier, field is valid only for Event counter group.

INDEX unsigned int

Counter index (as selected by PMSELR), field is valid only for Event/Cycle counter groups.

INTERRUPT bool

Is interrupt enabled on overflow for this counter.

POTINDEX_UPDATE

Value of POT index for current pc changed. Fields:

NEW_POTINDEX unsigned int

New POTIndex.

OLD_POTINDEX unsigned int

Old POTIndex.

PREFETCH_MEMORY64

Prefetch from PRFM or PRFUM instructions. Fields:

PRFOP enum

Prefetch hint.

VADDR unsigned int

Virtual address of the location that should be prefetched.

PRELOAD64_REQUEST

Preload requests from PRFM or PRFUM instructions. Fields:

PRFOP enum

Prefetch hint.

VADDR unsigned int

Virtual address of the location that should be prefetched.

PRELOAD_DATA

Data preload from PLD instruction. Fields:

VADDR unsigned int

Virtual address of the data that should be preloaded.

PRELOAD_INST

Instruction preload from PLI instruction. Fields:

VADDR unsigned int

Virtual address of the instruction that should be preloaded.

PRE_CORE_LOAD

Trace just before a core load. Fields:

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

PRE_CORE_STORE

Trace just before a core store. Fields:

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

PRE_TTB_READ

This event is triggered before reads caused by a translation table walk. Fields:

ADDR unsigned int

The physical address of the read.

PSTATE

PSTATE fields update. Fields:

ALLINT unsigned int

All interrupt mask bit.

BTYPE unsigned int

Branch target identification bit.

CORE_NUM unsigned int

Core number in a multi processor.

DAIF unsigned int

Exception mask bits.

DIT unsigned int

Data Independent Timing bit.

EL unsigned int

Current Exception level field.

EXLOCK unsigned int

Exception return state lock bit.

IL unsigned int

Illegal Execution state bit.

NZCV unsigned int

Condition flags.

PACM unsigned int

PACM bit.

PAN unsigned int

Privileged Access Never state bit.

PM unsigned int

PMU exception mask bit.

PPEND unsigned int

PMU exception pending bit.

SM unsigned int

StreamingMode status bit.

SP unsigned int

Stack pointer register selection bit.

SS unsigned int

Software Step bit.

SSBS unsigned int

Speculative Store Bypass Safe bit.

TCO unsigned int

Tag Check Override bit.

TOUCHED unsigned int

Mask of the fields that have been touched since this trace has last been emitted. Note that touched does not necessarily mean the value changed. The index of the field is the bit index in the mask plus the index of this field plus one.

UAO unsigned int

User Access Override bit.

UINJ unsigned int

Undefined Instruction exceptions bit.

ZA unsigned int

ZA storage and ZT0 register enable bit.

nRW unsigned int

Current Execution state bit.

PSTATE_PACTIVE_OPMODE_TRANSITION

Trace pactive and Opmode mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Operating mode after the transition.

OLD_STATE enum

Operating mode before the transition.

PSTATE_PACTIVE_OPMODE_TRANSITION_DENIED

Trace the denied pactive and Opmode mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Operating mode after the transition.

OLD_STATE enum

Operating mode before the transition.

REASON string

Reason for the transition being denied.

PSTATE_PACTIVE_POWER_TRANSITION

Trace pactive and power mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Power mode after the transition.

OLD_STATE enum

Power mode before the transition.

PSTATE_PACTIVE_POWER_TRANSITION_DENIED

Trace the denied pactive and power mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Power mode after the transition.

OLD_STATE enum

Power mode before the transition.

REASON string

Reason for the transition being denied.

PSTATE_PACTIVE_THREAD_OPMODE_TRANSITION

Trace pactive and Opmode mode transitions for threads on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Operating mode after the transition.

OLD_STATE enum

Operating mode before the transition.

QUANTUM_START

Called at the start of every quantum.

RANGE_PRELOAD_REQUEST

Preload requests from RPRFM. Fields:

COUNT unsigned int

Number of blocks of data to access.

LENGTH unsigned int

Number of contiguous bytes to be accessed in each block.

OP unsigned int

Range prefetch operation .

REUSE unsigned int

Maximum number of bytes to be accessed before next RPRFM .

STRIDE signed int

Stride to advance after accessing LENGTH bytes.

VADDR unsigned int

Base virtual address of range prefetch request.

RUN_STATE

Run state transition. Fields:

INST_COUNT unsigned int

Ticks count at point of transition.

NEW enum

New run state.

OLD enum

Old run state.

SEMIHOSTING_CALL

Call of a semihost function occurred. Fields:

PC unsigned int

The program counter after the semihosting call.

SEMIHOSTING_PRECALL

About to call semihost. Fields:

EL2 bool

Use EL2 translation regime.

NS bool

Is Non-Secure.

REG_WIDTH unsigned int

The current register width in bytes.

SIGNAL

External signal state change. Fields:

SIGNAL enum

Signal that changed.

STATE bool

Signal asserted state.

SOFTWARE_STEP

Return the Debug Software Step state. Fields:

STATE enum

Software Step state.

SPSR

SPSR change. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

MODE enum

Which of the banked SPSR registers is written.

OLD_VALUE unsigned int

The old SPSR value.

VALUE unsigned int

The new SPSR value.

START_COMPILE

Compilation started. Fields:

VADDR unsigned int

Instruction where compilation begins.

STG_MULTIPLE_MICRO_OPS_ADDRESS_RANGE

Address range for MTE stores with multiple uops. Fields:

ADDR_END unsigned int

Final address.

ADDR_START unsigned int

Starting address.

STORE_MULTIPLE_REGS_ADDRESS_RANGE

Address range for stores with multiple registers. Fields:

ADDR_END unsigned int

Final address.

ADDR_START unsigned int

Starting address.

SUBOPTIMAL_LDST_RETIRED

suboptimal load/store. Fields:

ACCESS_SIZE unsigned int

Log2 of the access width in bytes used for alignment checking (1: half-word, 2: word ...).

OPTIMAL_ALIGN_SELECTOR unsigned int

Selector to decide the optimal alignment size (1: access width, 2: 4-bytes, 3: 8-bytes, ..., 12: 4-kbytes).

VADDR unsigned int

The virtual address of the access.

SYNC

Called for every synchronization, that is every quantum break. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The instruction count of this CPU.

LOCAL_QUANTUM unsigned int

The local quantum of this CPU.

LOCAL_TIME unsigned int

The local time of this CPU.

SYNC_REASON enum

Reason for sync.

SYSCALL

System call instruction executed. Fields:

IMM unsigned int

Immediate value of the system call instruction.

TYPE enum

System call type.

VADDR unsigned int

Instruction that caused the system call.

SYSREG_READ128

System Coprocessor 128 bit register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

REGNUM enum

Internal register number.

REGNUM_PHYS enum

Internal register number physically accessed (e.g. in presense of VHE redirection).

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_READ64

System Coprocessor register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REGNUM enum

Internal register number.

REGNUM_OPERAND enum

Register number corresponding to the instruction operand.

REGNUM_PHYS enum

Internal register number physically accessed (e.g. in presense of VHE redirection).

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_UPDATE128

Triggers when the system updates a register. Fields:

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG enum

Register number.

UNKNOWN unsigned int

Bits of the register which became unknown.

VALUE unsigned int

Value written to the register.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_UPDATE32

Triggers when the system updates a register. Fields:

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG enum

Register number.

UNKNOWN unsigned int

Bits of the register which became unknown.

VALUE unsigned int

Value written to the register.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_UPDATE64

Triggers when the system updates a register. Fields:

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG enum

Register number.

UNKNOWN unsigned int

Bits of the register which became unknown.

VALUE unsigned int

Value written to the register.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_WRITE128

System Coprocessor 128 bit register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

REGNUM enum

Internal register number.

REGNUM_PHYS enum

Internal register number physically accessed (e.g. in presense of VHE redirection).

UNDEF bool

The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_WRITE64

System Coprocessor register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REGNUM enum

Internal register number.

REGNUM_OPERAND enum

Register number corresponding to the instruction operand.

REGNUM_PHYS enum

Internal register number physically accessed (e.g. in presense of VHE redirection).

UNDEF bool

The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

UNALIGNED_LDST_RETIRED

Processor unaligned load/store. Fields:

ACCESS_SIZE unsigned int

Log2 of the access width in bytes used for alignment checking (1=>half-word, 2->word ...).

NUMBER_OF_BEATS unsigned int

Number of accesses (beats) in this burst (Total size in bytes is ACCESS_SIZE * NUMBER_OF_BEATS).

VADDR unsigned int

The virtual address of the access.

VFP_D_REGS

VFP/NEON D 64 bit register write. Fields:

ALIASING enum

Type of register aliasing used.

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a write to the corresponding bit in the register. For example, 0x1 would indicate a write to VALUE[0].

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

VFP_D_REGS_READ

VFP/NEON D 64 bit register read. Fields:

ID unsigned int

The register number.

VALUE unsigned int

The current value of the register.

VFP_Q_REGS

VFP/NEON Q 128 bit register write. Fields:

ALIASING enum

Type of register aliasing used.

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a write to the corresponding byte in the register. For example, 0x1 would indicate a write to VALUE[7:0].

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

VFP_Q_REGS_READ

VFP/NEON Q 128 bit register read. Fields:

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a read to the corresponding byte in the register. For example, 0x1 would indicate a read to VALUE[7:0].

VALUE unsigned int

The current value of the register.

VFP_SYS_REGS

Writes to the VFP/NEON units system registers. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ID enum

Which VFP system register is written.

MASK unsigned int

The mask indicating updated cumulative bits.

OLD_VALUE unsigned int

The register's old value overwritten.

VALUE unsigned int

The new value written to the VFP system register.

VFP_S_REGS

VFP/NEON S 32 bit register write. Fields:

ALIASING enum

Type of register aliasing used.

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a write to the corresponding bit in the register. For example, 0x1 would indicate a write to VALUE[0].

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

VFP_S_REGS_READ

VFP/NEON S 32 bit register read. Fields:

ID unsigned int

The register number.

VALUE unsigned int

The current value of the register.

VIRTUAL_TAG_ALLOCATED

Virtual tag information when allocated. Fields:

TAG_VADDR unsigned int

Virtual tag virtual address.

VADDR unsigned int

Data Virtual address.

VIRTUAL_TAG unsigned int

Virtual tag value.

WAYPOINT

Signals end of basic block execution. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ISET enum

Origin instruction set.

IS_COND bool

Indicates if this is a conditional waypoint.

PC unsigned int

Origin address (or 0 if unavailable).

TAKEN bool

Indicates if this waypoint was taken.

TARGET unsigned int

Destination address.

TARGET_ISET enum

Destination instruction set.

WFE_END

WFE ended. Fields:

INST_COUNT unsigned int

Ticks count when leaving WFE.

WFE_EVENT_REGISTER

WFE event register status: set/clear, reason. Fields:

INST_COUNT unsigned int

Ticks count.

REASON enum

Reason for set/clear. Only REASON==0 (Cleared by WFE) clears the bit, all other reasons set it.

WFE_IGNORED

WFE ignored. Fields:

EVENT bool

This WFE was ignored because the event register was set.

INST_COUNT unsigned int

Ticks count when ignoring WFE.

TRAPPED bool

This WFE was trapped.

WFE_START

WFE entered. Fields:

INST_COUNT unsigned int

Ticks count when entering WFE.

WFI_END

WFI ended. Fields:

INST_COUNT unsigned int

Ticks count when leaving WFI.

WFI_IGNORED

WFI ignored. Fields:

DISABLED bool

This WFI was ignored because WFI is disabled.

INST_COUNT unsigned int

Ticks count when ignoring WFI.

REASON enum

specifies reason why WFI trace was ignored.

TRAPPED bool

This WFI was trapped.

WFI_START

WFI entered. Fields:

INST_COUNT unsigned int

Ticks count when entering WFI.

WFI_WAKEUP

WFI wakeup. Fields:

INST_COUNT unsigned int

Ticks count when WFI wakeup occurred.

REASON enum

Reason for wakeup.

2.22 ARM_Cortex-A65

This section describes the trace sources.

AA64_ASE_SVE_REGS

Modification of AArch64 Advanced SIMD or SVE Vector Registers (via SIMD or Scalar operations), SVE Predicate Registers and ZT0. Fields:

ID **enum**

The register identifier.

MASK **unsigned int**

Bitmask of the register to signal the modified bits in the VALUE field.

SM **bool**

Whether the PE is in Streaming Mode.

VALUE **unsigned int**

Value of AArch64 vector register.

ASYNC_MEMORY_FAULT

Context ID Register write. Fields:

FAULT **unsigned int**

Fault status present in ESR.ISS encoding for Serror in ARCH64 and present in DFSR encoding for external abort format in ARCH32.

PADDR **unsigned int**

Physical Address (or 0 if unavailable).

VADDR **signed int**

Virtual Address (or 0 if unavailable).

ATOMIC_END_ACCESS

Bus trace access for atomics. Fields:

ACCESS_FAIL **bool**

Memory access failed.

ADDR **unsigned int**

The virtual address of the access.

ATTR **unsigned int**

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

COMPARE_VALUE **unsigned int**

Compare value for CAS.

LOAD_VALUE unsigned int

Loaded values.

NSDESC unsigned int

The security state of the access.

OPERAND_VALUE unsigned int

Operation's operand.

OPERATION enum

Operation type.

PADDR unsigned int

The physical address of the access.

PRIV bool

Is this a privileged access?.

ATOMIC_START_ACCESS

Bus trace access for atomics. Fields:

ADDR unsigned int

The virtual address of the access.

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

COMPARE_VALUE unsigned int

Compare value for CAS.

NSDESC unsigned int

The security state of the access.

OPERAND_VALUE unsigned int

Operation's operand.

OPERATION enum

Operation type.

PADDR unsigned int

The physical address of the access.

PRIV bool

Is this a privileged access?.

ArchMsg.Error.exit_code

DISPLAY exit code: %{EXITCODE} component kind: %{KIND} reason: %{REASON}. Fields:

EXITCODE signed int

Exit code to be returned.

KIND string

Component kind that invoked the exit code trace.

REASON string

Reason why the exit code trace was invoked (optional).

ArchMsg.Warning.IgnoredWarmResetRequest

Event fires when software attempts to request warm reset through write to deprecated DBGPRCR.CWRR.

ArchMsg.Warning.RAS_ErrselrUnpredictableState

DISPLAY ERRSELR is in unpredictable state due to value greater than number_of_error_records written to it.

ArchMsg.Warning.RAS_UnpredictableStateWrite

DISPLAY %{REG_NAME} write ignored due to unpredictable ERRSELR state. Fields:

REG_NAME string

Name of register being written.

ArchMsg.Warning.ReportedRevisionRegisterRead

DISPLAY Reported revision and patch do not match that of the cpu. Fields:

ORIGINAL_VALUE unsigned int

The value the register would have based on the modelled revision.

REGISTER string

The name of the register that has ben read.

REPORTED_VALUE unsigned int

The value the register has been overridden to.

ArchMsg.Warning.Unpredictably Indexed PM Event Register

TAGS UNPREDICTABLE DISPLAY Event %{IsDirect:(Selector value|Register)} %{SEL} is >= %{N}. Fields:

IsDirect unsigned int

Direct Access rather than indirect.

N unsigned int

Number of Accessible Registers.

SEL unsigned int

Selector.

ArchMsg.Warning.cache_contents_unknown

A cache has been activated without being invalidated and may contain **UNKNOWN** content. DISPLAY %{SIDE:(D|I)}-Cache was enabled in %{REGIME:(SECURE|NON_SECURE|HYP)} regime but was not invalidated since power-on: cache lines could contain **UNKNOWN** content %{SIDE:(Note: D-side caches must invalidate SECURE lines even if the cache is only used by NON_SECURE code in order to ensure that unexpected dirty lines tagged SECURE are not naturally evicted)}. Fields:

REGIME unsigned int

Current translation regime.

SIDE unsigned int

0 d-side cache or unified cache, 1 i-side.

ArchMsg.Warning.dcimvac_matches_watchpoint

TAGS **UNPREDICTABLE** DISPLAY Watchpoints matching an AArch32 DCIMVA are implementation defined.

ArchMsg.Warning.decode_fieldmismatch

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with different fields %{FIELD1} and %{FIELD2}. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_fracbitsnegative

TAGS **UNPREDICTABLE** DISPLAY VCVT Instruction is unpredictable with a negative number of fraction bits.

ArchMsg.Warning.decode_initblock

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable in an IT block.

ArchMsg.Warning.decode_initblocknotlast

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable in an IT block when not the last.

ArchMsg.Warning.decode_invalid_condition

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable when the condition is not AL.

ArchMsg.Warning.decode_invalidfieldcombination

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with the values in %{FIELD1} and %{FIELD2}. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_invalidvalue

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with the value in %{FIELD}. Fields:

FIELD enum

Field name.

ArchMsg.Warning.decode_mem_hint_unallocated

TAGS **UNPREDICTABLE** DISPLAY There is no memory hint allocated to this bit pattern.

ArchMsg.Warning.decode_registermatch

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with %{FIELD1} and %{FIELD2} as the same register. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_registermismatch

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with %{FIELD1} and %{FIELD2} as different registers. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_registeroutofrange

TAGS **UNPREDICTABLE** DISPLAY Floating-point instruction is unpredictable indexing beyond the end of the register bank.

ArchMsg.Warning.decode_sbzsbo

TAGS **UNPREDICTABLE** DISPLAY Reserved bits in the instruction are not canonical.

ArchMsg.Warning.decode_transactiontoobig

TAGS **UNPREDICTABLE** DISPLAY Attempt to perform a memory transaction of more than 128 bytes.

ArchMsg.Warning.decode_unpred_other

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable.

ArchMsg.Warning.decode_unpreduseofpc

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable when %{FIELD} is PC. Fields:

FIELD enum

Field name.

ArchMsg.Warning.decode_unpreduseofr13

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable when %{FIELD} is R13. Fields:

FIELD enum

Field name.

ArchMsg.Warning.decode_unpreduseofr13inreglist

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with R13 in register-list.

ArchMsg.Warning.decode_useofd16orhigher

TAGS **UNPREDICTABLE** DISPLAY Attempt to use register D16 or higher where it is not allowed.

ArchMsg.Warning.decode_writebackandbaseinlist

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with write back when the base register is in the transfer list.

ArchMsg.Warning.decode_zeroregistersinlist

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable when no registers are to be transferred.

ArchMsg.Warning.recursive_branch

An instruction is performing a branch that targets the same instruction. If this was intended then a WFI might be more effective DISPLAY recursive branch detected at PC=%{PC}. PRIMARY KEY PC. Fields:

PC unsigned int

Address of instruction causing the branch.

ArchMsg.Warning.recursive_exception

An instruction has generated an exception that targets the same instruction. DISPLAY recursive %{TYPE} exception detected at PC=%{PC} (CPSR=%{CPSR}). PRIMARY KEY PC. Fields:

CPSR unsigned int

Processor state.

PC unsigned int

Address of instruction causing the exception.

TYPE enum

Type of exception.

ArchMsg.Warning.reentrant_vector_catch

TAGS **UNPREDICTABLE** DISPLAY Vector catch debug event on Prefetch or Data abort vector %{ADDR}. Fields:

ADDR unsigned int

vector address.

ArchMsg.Warning.reserved_it_state

TAGS **UNPREDICTABLE** DISPLAY Execution with reserved IT state %{ITSTATE}. Fields:

ITSTATE unsigned int

reserved IT state.

ArchMsg.Warning.tlb_contents_unknown

Translation has been enabled without invalidating the TLB. DISPLAY Translation was enabled in the %{PAS:(Secure|Non-secure|Root|Realm)} EL%{EL:d} translation regime (vmid=%{VMID:x}) without invalidating %{INVALIDITY:(|d-side |i-side |)}entries since power-on: %{INVALIDITY:(|D|I|)}TLB could contain **UNKNOWN** content. Fields:

EL unsigned int

Exception level of translation regime.

INVALIDITY unsigned int

Which TLB may be invalid (if TLBs are separate).

NON_SECURE bool

Whether the translation regime is non-secure.

PAS unsigned int

Physical address space of translation regime.

VMID unsigned int

Current VMID, if applicable.

ArchMsg.Warning.unknown_DLR_DSPSR

Exiting debug state with unknown DLR or DSPSR.

ArchMsg.Warning.unknown_ELR_SPSR

Returning from debug exception with unknown ELR or SPSR.

ArchMsg.Warning.unpred_FAR_on_LRCPC3_non_atomic_mem_access

TAGS **UNPREDICTABLE** Unpredictable FAR/FSTATUS on LRCPC3 pair instructions resulting in two separate memory accesses
DISPLAY Unpredictable FAR on LRCPC3 non atomic memory access at %{ADDR1} with FSTATUS %{FSTATUS1} or %{ADDR2} with FSTATUS %{FSTATUS2}. Fields:

ADDR1 unsigned int

Possible FAR address 1.

ADDR2 unsigned int

Possible FAR address 2.

FSTATUS1 unsigned int

Possible fault status value 1.

FSTATUS2 unsigned int

Possible fault status value 2.

ArchMsg.Warning.unpred_async_tag_check_fail_on_sve_speculative_load_do_update_ffr

Unpredictable set to FALSE of the FFR element associated to SVE speculative load with an async tag check fail TAGS **UNPREDICTABLE**.

ArchMsg.Warning.unpredictable_a32_breakpoint

TAGS **UNPREDICTABLE** DISPLAY Unpredictable breakpoint on a A32 instruction at %{ADDR1} Offset %{OFFSET} BAS %{BAS} for a %{IS_ADDRESS_MISMATCH_BREAKPOINT:(Mismatch|Match)} Breakpoint number %{N:d} Breakpoint address %{ADDR2}. Fields:

ADDR1 unsigned int

Address.

ADDR2 unsigned int

Breakpoint address.

BAS unsigned int

BAS field.

IS_ADDRESS_MISMATCH_BREAKPOINT bool

Is a Mismatch breakpoint, not a Match breakpoint.

N unsigned int

Breakpoint number.

OFFSET unsigned int

Always has a value of zero.

ArchMsg.Warning.unpredictable_banked_register_access

TAGS **UNPREDICTABLE** DISPLAY Access to banked register (R=%{R:d} SYSm=%{SYSm:2x}) is unpredictable. Fields:

R unsigned int

R bit.

SYSm unsigned int

SYSm.

ArchMsg.Warning.unpredictable_ccfail_undef

TAGS **UNPREDICTABLE** DISPLAY An instruction %{OPCODE} at VA=%{VADDR} failed its condition codes check but would otherwise have caused a non-data-dependent trap or undefined exception. Behaviour is implementation-defined choice of exception or nop. Fields:

OPCODE unsigned int

opcode of the instruction.

VADDR unsigned int

Virtual address of the instruction being executed.

ArchMsg.Warning.unpredictable_ici_bits

TAGS **UNPREDICTABLE** DISPLAY ICI bits do not apply or are architecturally unpredictable for instruction.

ArchMsg.Warning.unpredictable_t32_breakpoint

TAGS **UNPREDICTABLE** DISPLAY Unpredictable breakpoint on a T32 instruction at %{ADDR1} Offset %{OFFSET} BAS %{BAS} for a %{IS_ADDRESS_MISMATCH_BREAKPOINT:(Mismatch|Match)} Breakpoint number %{N:d} Breakpoint address %{ADDR2}. Fields:

ADDR1 unsigned int

Address.

ADDR2 unsigned int

Breakpoint address.

BAS unsigned int

BAS field.

IS_ADDRESS_MISMATCH_BREAKPOINT bool

Is a Mismatch breakpoint, not a Match breakpoint.

N unsigned int

Breakpoint number.

OFFSET unsigned int

Indicates which halfword of a 32-bit Thumb instruction the breakpoint is set on. 0 for the first halfword or if the instruction is 16 bits wide, or 1 for the second halfword.

ArchMsg.Warning.unpredictable_t32_vector_catch

TAGS **UNPREDICTABLE** DISPLAY Unpredictable Vector catch exception on a T32 instruction at **%{ADDR}** Offset **%{OFFSET}**. Fields:

ADDR unsigned int

Address.

OFFSET unsigned int

Indicates which halfword of a 32-bit Thumb instruction the vector catch exception is set on. 0 for the first halfword or if the instruction is 16 bits wide, or 1 for the second halfword.

ArchMsg.Warning.unpredictable_watchpoint_far

TAGS **UNPREDICTABLE** DISPLAY The value for FAR/EDWAR for hitting this watchpoint may be anywhere in the range **%{LOWER_BOUND}** to **%{UPPER_BOUND}**. Fields:

LOWER_BOUND unsigned int

the lowest address accessed by the instruction that triggered the watchpoint.

UPPER_BOUND unsigned int

the highest watchpointed address accessed by that instruction.

ArchMsg.Warning.unpredictable_watchpoint_far_isscmp

DISPLAY The value for FAR/EDWAR for hitting this watchpoint may be anywhere in the range **%{LOWER_BOUND}** to **%{UPPER_BOUND}**. Fields:

LOWER_BOUND unsigned int

the lowest address accessed by the instruction that triggered the watchpoint.

UPPER_BOUND unsigned int

the highest watchpointed address accessed by that instruction.

ArchMsg.Warning.warning_AdvSIMDEExpandImmUnexpectedZero

TAGS **UNPREDICTABLE** DISPLAY AdvSIMDEExpandImm may treat this immediate value as **UNPREDICTABLE**.

ArchMsg.Warning.warning_ConditionalSMC

TAGS **UNPREDICTABLE** DISPLAY SMC instruction has **UNPREDICTABLE** effects when conditional and when combined with traps.

ArchMsg.Warning.warning_TLBID_out_of_range

TLBID out of range in TLBI according to the configuration of NIS, NVIS, NOS and NVOS. Fields:

NIS unsigned int

NIS.

NOS unsigned int

NOS.

NVIS unsigned int

NVIS.

NVOS unsigned int

NVOS.

TLBID unsigned int

The TLBID value used in the TLBI instruction.

ArchMsg.Warning.warning_TTL_or_TG_mismatches_current_translation_granule

TTL or TG mismatches current translation granule in TLBI VA/RVA/IPA/RIPA. Fields:

CURRENT_TG unsigned int

Current TG as indicated by TCR_ELx or VTCR_EL2.

TLBI_TG unsigned int

TG (or TTL[3:2]) used in TLBI instruction.

ArchMsg.Warning.warning_access_crosses_page_boundary

TAGS **UNPREDICTABLE** DISPLAY Access crosses page boundary.

ArchMsg.Warning.warning_access_crosses_page_boundary_has_fault

TAGS **UNPREDICTABLE** DISPLAY Access that crosses page boundary has fault(s).

ArchMsg.Warning.warning_access_crosses_page_boundary_has_fault_on_both_pages

TAGS **UNPREDICTABLE** DISPLAY Access that crosses page boundary has a fault on each page. Fields:

ADDR1 unsigned int

FAR address on first page.

ADDR2 unsigned int

FAR address on second page.

FSTATUS1 unsigned int

Fault status on first page.

FSTATUS2 unsigned int

Fault status on second page.

ArchMsg.Warning.warning_access_crosses_page_boundary_spanning_different_memory

TAGS **UNPREDICTABLE** DISPLAY Access at address %{ADDR} crosses page from MEM_TYPE %{MEMTYPE_PAGE1} to %{MEMTYPE_PAGE2}. Fields:

ADDR unsigned int

address of access crossing page.

MEMTYPE_PAGE1 signed int

First Page's memory type (0-Normal, 1-Non-Normal).

MEMTYPE_PAGE2 signed int

Second Page's memory type (0-Normal, 1-Non-Normal).

ArchMsg.Warning.warning_access_wraps_around_memory

TAGS **UNPREDICTABLE** DISPLAY Access wraps around memory in %{WIDTH:d}bit mode.

Fields:

WIDTH unsigned int

Address width.

ArchMsg.Warning.warning_bcr_linking_status

TAGS **UNPREDICTABLE** DISPLAY DBGBCR.{BT2,BT} combination gives unpredictable behaviour.

ArchMsg.Warning.warning_bcr_mask_reserved

TAGS **UNPREDICTABLE** DISPLAY DBGBCR_ELn.MASK is set to a reserved value.

ArchMsg.Warning.warning_bvr_masked_bit_not_zero

TAGS **UNPREDICTABLE** DISPLAY DBGBVRn_EL1.VA contains a bit masked by DBGBCRn_EL1.MASK that is not zero.

ArchMsg.Warning.warning_ccsidr_unimplemented_level

TAGS **UNPREDICTABLE** DISPLAY CCSIDR read while CSSELR %{CSSELR:x} points to an unimplemented cache level. Fields:

CSSELR unsigned int

current effective value of CSSELR.

ArchMsg.Warning.warning_change_to_ns_when_tge_set

TAGS **UNPREDICTABLE** DISPLAY Attempting to change NS to 1 when HCR.TGE = 1.

ArchMsg.Warning.warning_contiguous_bit_check_abort

TAGS **UNPREDICTABLE** DISPLAY An abort occurred whilst attempting to check TLB entry is contiguous at %{ENTRY_ADDR}. Fields:

ENTRY_ADDR unsigned int

address of conflicting TLB entry.

ArchMsg.Warning.warning_contiguous_bit_error

TAGS **UNPREDICTABLE** DISPLAY TLB entry at %{ENTRY_ADDR} was not contiguous with entry at %{CONTIG_ADDR} Contents are %{ENTRY_DATA} but expected %{CONTIG_DATA}. Fields:

CONTIG_ADDR unsigned int

address of the TLB first read, with which this entry is expected to be contiguous.

CONTIG_DATA unsigned int

expected contents based on the entry at CONTIG_ADDR.

ENTRY_ADDR unsigned int

address of conflicting TLB entry.

ENTRY_DATA unsigned int

contents of conflicting TLB entry.

ArchMsg.Warning.warning_contiguous_bit_error_D128

TAGS **UNPREDICTABLE** DISPLAY TLB entry at %{ENTRY_ADDR} was not contiguous with entry at %{CONTIG_ADDR} Contents are %{ENTRY_DATA_HI} %{ENTRY_DATA_LO} but expected %{CONTIG_DATA}. Fields:

CONTIG_ADDR unsigned int

address of the TLB first read, with which this entry is expected to be contiguous.

CONTIG_DATA unsigned int

expected contents based on the entry at CONTIG_ADDR.

ENTRY_ADDR unsigned int

address of conflicting TLB entry.

ENTRY_DATA_HI unsigned int

Higher bit contents of conflicting TLB entry.

ENTRY_DATA_LO unsigned int

Lower bit contents of conflicting TLB entry.

ArchMsg.Warning.warning_cp10_cp11_mismatch

TAGS **UNPREDICTABLE** DISPLAY Unpredictable configuration of CP10 and CP11 controls at EL %{EL:d}. Fields:

EL unsigned int

EL owning the control.

ArchMsg.Warning.warning_cp10_cp11_reserved_value

TAGS **UNPREDICTABLE** DISPLAY Unpredictable configuration of CPACR CP10 (%{CP10}) or CP11 (%{CP11}). Fields:

CP10 unsigned int

CP10 access.

CP11 unsigned int

CP11 access.

ArchMsg.Warning.warning_csselr_level_out_of_range

TAGS **UNPREDICTABLE** DISPLAY CSSELR.Level out of range, written value %{LEVEL} with only %{IMPLEMENTED:d} levels implemented. Fields:

IMPLEMENTED unsigned int

number of cache levels implemented.

LEVEL unsigned int

written value of CSSELR.Level.

TYPE enum

The type of cache selected by CSSELR.InD and TnD.

ArchMsg.Warning.warning_debug_flow_control_bits_not_obeyed

TAGS **UNPREDICTABLE** DISPLAY Flow control bits for %{REGISTER} were not obeyed, causing %{ERROR}. Fields:

ERROR enum

type of error resulting.

REGISTER enum

register under consideration.

ArchMsg.Warning.warning_debug_register_access_during_reset

TAGS **UNPREDICTABLE** DISPLAY Unpredictable %{IS_WRITE:(read|write)} access to debug register offset %{OFFSET} during reset. Fields:

IS_WRITE unsigned int

Write Not Read.

OFFSET unsigned int

Register Offset.

ArchMsg.Warning.warning_decode_cps_inconsistent_fields

TAGS **UNPREDICTABLE** DISPLAY The mode setting fields in the CPS instruction are inconsistent.

ArchMsg.Warning.warning_decode_invalid_state

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable in current exception-level/security state.

ArchMsg.Warning.warning_default_cacheable_mmu_on

TAGS **UNPREDICTABLE** DISPLAY Default cacheable is enabled (HCR.DC = 1) while MMU is enabled.

ArchMsg.Warning.warning_default_cacheable_vmmu_off

TAGS **UNPREDICTABLE** DISPLAY Default cacheable is enabled (HCR.DC = 1) while HCR.VM = 0.

ArchMsg.Warning.warning_deprecated_wvr_bit_2_set

TAGS **UNPREDICTABLE** DISPLAY DBGWVR%{N:d}_EL1=%{WVR:h} has bit 2 set, deprecated in ARMv8. Fields:

N unsigned int

index.

WVR unsigned int

DBGWVR.

ArchMsg.Warning.warning_eret_while_software_step_active_pending

Missing ISB between setting MDSCR_EL1.SS and ERET TAGS **UNPREDICTABLE**.

ArchMsg.Warning.warning_exclusive_to_non_normal

TAGS **UNPREDICTABLE** DISPLAY Exclusive Access to Non-Normal Memory %{ADDR}. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_exclusive_to_non_writeback

TAGS **UNPREDICTABLE** DISPLAY Exclusive Access to Non-Writeback-Cacheable Memory
%{ADDR}. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_execute_from_device_memory

TAGS **UNPREDICTABLE** DISPLAY An attempt was made to execute from Device memory.

ArchMsg.Warning.warning_illegal_cpsr_mode

TAGS **UNPREDICTABLE** DISPLAY Writing an illegal or unimplement mode (%{NEW_MODE:h})
to CPSR was unpredictable. Fields:

NEW_MODE unsigned int

New value of CPSR.M.

ArchMsg.Warning.warning_illegal_srs_mode

TAGS **UNPREDICTABLE** DISPLAY Illegal or **UNPREDICTABLE** mode (%{MODE:h}) used for SRS
instruction. Fields:

MODE unsigned int

mode for Banked SP.

ArchMsg.Warning.warning_implementation_defined_read_debug_register_in_SCS

Read Access to Debug registers from the processor is IMPLEMENTATION DEFINED TAGS
IMP_DEF. Fields:

PPB_OFFSET unsigned int

debug register offset.

ArchMsg.Warning.warning_implementation_defined_sequential_security_transitions_supported

The behavior of sequential instruction fetches that cross from non-secure to secure memory
and contain an SG instruction is CONSTRAINED **UNPREDICTABLE** TAGS **UNPREDICTABLE**.

ArchMsg.Warning.warning_implementation_defined_stack_limit_check_supported

It is **IMPLEMENTATION DEFINED** whether stack pointer limit checking is performed for this
instruction TAGS IMP_DEF. Fields:

ADDRESS unsigned int

Address of instruction.

OPCODE unsigned int

instruction opcode.

ArchMsg.Warning.warning_implementation_defined_taggednonsharedmemorytype

TAGS IMP_DEF DISPLAY Memory attribute has Tagged, non shared memory type; it is
IMPLEMENTATION DEFINED whether the memory location is treated as Tagged or non-Tagged.

ArchMsg.Warning.warning_implementation_defined_write_debug_register_in_SCS

Write Access to Debug registers from the processor is IMPLEMENTATION DEFINED TAGS IMP_DEF. Fields:

DATA unsigned int

data attempted to be written.

PPB_OFFSET unsigned int

debug register offset.

ArchMsg.Warning.warning_invalid_tcr_granule

TAGS UNPREDICTABLE DISPLAY TCR.TG{%TG_ID:(0|1)} bits have been set to a granule size {%REQUEST:d}K not implemented - using {%SUBSTITUTE:d}K. Fields:

REQUEST signed int

page size requested (or 0 for reserved).

SUBSTITUTE signed int

best guess available page size.

TG_ID bool

bits TG0 or TG1.

ArchMsg.Warning.warning_load_multiple_user_registers_from_user_mode

TAGS UNPREDICTABLE DISPLAY An LDM(user registers) instruction executed from user mode.

ArchMsg.Warning.warning_mte_in_mem_attr_not_supported

TAGS UNPREDICTABLE DISPLAY Memory attribute {%ATTR} has Tagged memory type but Armv8.5-MTE is not implemented. Fields:

ATTR unsigned int

Memory attribute.

ArchMsg.Warning.warning_reserved_breakpoint_state_match

TAGS UNPREDICTABLE DISPLAY {%IS_BREAKPOINT:(Watchpoint|Breakpoint)} programmed with a reserved combination of HMC, SSC and {%IS_BREAKPOINT:(PAC|PMC)}. Fields:

IS_BREAKPOINT unsigned int

Is a breakpoint, not a watchpoint.

ArchMsg.Warning.warning_shareability

TAGS UNPREDICTABLE DISPLAY Unpredictable: combination of the 1st and 2nd stages of translation is Normal Inner Non-Cacheable, Outer Non-Cacheable.

ArchMsg.Warning.warning_software_step_set_while_enabled

TAGS UNPREDICTABLE DISPLAY MDSCR_EL1.SS set to 1 while software step debug exceptions are enabled.

ArchMsg.Warning.warning_thumb_instruction_wraps_around_memory

TAGS UNPREDICTABLE DISPLAY Thumb instruction wraps around memory.

ArchMsg.Warning.warning_ttbr_sbz_bits_are_not_zero

TAGS UNPREDICTABLE DISPLAY %{TTBR}[%{MSB:d}:0] should be zero, but are not. Fields:

MSB unsigned int

x-1.

TTBR enum

which TTBR register.

ArchMsg.Warning.warning_unaligned_address_dbgdttrrx_write

TAGS UNPREDICTABLE DISPLAY Unaligned address %{ADDR} in XO/RO when DBGDTTRRX written in memory access mode. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unaligned_address_dbgdttrtx_read

TAGS UNPREDICTABLE DISPLAY Unaligned address %{ADDR} in XO/RO when DBGDTTRTX read in memory access mode. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unaligned_to_device

TAGS UNPREDICTABLE DISPLAY Unaligned Access to Device Memory %{ADDR}. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unaligned_to_strongly_ordered

TAGS UNPREDICTABLE DISPLAY Unaligned Access to Strongly Ordered Memory %{ADDR}. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unknown_sau_rnr

TAGS UNPREDICTABLE DISPLAY SAU_RNR was set to an unsupported value.

ArchMsg.Warning.warning_unpredictable_AIRCR_PRIS_and_BFHFNMINs

TAGS UNPREDICTABLE DISPLAY The effect of setting both AIRCR.BFHFNMINs and AIRCR.PRIS to 1 is UNPREDICTABLE.

ArchMsg.Warning.warning_unpredictable_AIRCR_VECTCLRACTIVE_when_not_in_debug

TAGS UNPREDICTABLE DISPLAY The effect of writing a 1 to AIRCR.VECTCLRACTIVE if the processor is not halted in Debug state is UNPREDICTABLE.

ArchMsg.Warning.warning_unpredictable_AIRCR_VECTRESET_and_SYSRESETREQ

TAGS UNPREDICTABLE DISPLAY When the processor is halted in Debug state, if a write to the register writes a 1 to both VECTRESET and SYSRESETREQ, the behavior is UNPREDICTABLE.

ArchMsg.Warning.warning_unpredictable_AIRCR_VECTRESET_when_not_in_debug

TAGS **UNPREDICTABLE** DISPLAY The effect of writing a 1 to AIRCR.VECTRESET if the processor is not halted in Debug state is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_AIRCR_incorrect_VKEY

TAGS **UNPREDICTABLE** DISPLAY The value 0x05FA must be written to AIRCR.VKEY, otherwise the register write is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_EXC_RETURN_Reserved_Bit

EXC_RETURN[23:7] are reserved with the special condition that all bits should be written as one. Values other than all 1s are **UNPREDICTABLE**. TAGS **UNPREDICTABLE**. Fields:

EXC_RETURN unsigned int

Special PC value.

SBO_MASK unsigned int

Which bits Should Be One to not UNPRED.

ArchMsg.Warning.warning_unpredictable_PMEVTYPEPn_ELO_VS_reserved_value

TAGS **UNPREDICTABLE** DISPLAY Unpredictable reserved value of PMEVTYPEn<n>_ELO.VS.

ArchMsg.Warning.warning_unpredictable_change_to_DEMCR_MON_STEP_at_insufficient_priority

TAGS **UNPREDICTABLE** DISPLAY The effect of changing DEMCR.MON_STEP at an execution priority that is lower than the priority of the DebugMonitor exception is **CONSTRAINED UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_change_to_priority_of_active_exception

TAGS **UNPREDICTABLE** DISPLAY Changing the priority of an active exception is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_clear_lspact

TAGS **UNPREDICTABLE** DISPLAY **UNPREDICTABLE** state on context switch. Fields:

DRVEXC_TAKEN unsigned int

Derived exception taken, not pended.

REACHED_LAST_STORE unsigned int

Only the last faulted.

ArchMsg.Warning.warning_unpredictable_dbg_exit_unaligned_dlr

TAGS **UNPREDICTABLE** DISPLAY Debug exit to AArch32 with DLR[0] set to 1. PC[0] can be set to 0 or to DLR[0].

ArchMsg.Warning.warning_unpredictable_exception_catch

TAGS **UNPREDICTABLE** DISPLAY Generation of an exception catch event is unpredictable.

ArchMsg.Warning.warning_unpredictable_exception_return_inconsistent_state

TAGS **UNPREDICTABLE** DISPLAY state on exception return is unpredictable.

ArchMsg.Warning.warning_unpredictable_exception_return_instruction

TAGS **UNPREDICTABLE** DISPLAY exception return instruction is unpredictable.

ArchMsg.Warning.warning_unpredictable_in_debug_state

TAGS **UNPREDICTABLE** DISPLAY Instruction is **UNPREDICTABLE** when executed in debug state.

ArchMsg.Warning.warning_unpredictable_mair_encoding

TAGS **UNPREDICTABLE** DISPLAY Memory attribute %{ATTR} has invalid MAIR encoding; this may lead to unpredictable results. Fields:

ATTR unsigned int

Memory attribute.

ArchMsg.Warning.warning_unpredictable_pmu_counter_access

TAGS **UNPREDICTABLE** DISPLAY Access to PMU counters from non-secure EL0 or EL1 is **UNPREDICTABLE** with MDCR_EL2.HPMN set to 0.

ArchMsg.Warning.warning_unpredictable_prioritization_breakpoint_match_vector_catch

TAGS **UNPREDICTABLE** DISPLAY Two events with the same priority occurred at the same instruction: Address Matching Vector Catch debug event, and Breakpoint debug event. It is constrained unpredictable which is taken.

ArchMsg.Warning.warning_unpredictable_stack_selection

TAGS **UNPREDICTABLE** DISPLAY **UNPREDICTABLE** state on context switch.

ArchMsg.Warning.warning_unpredictable_tsize_out_of_range

TAGS **UNPREDICTABLE** DISPLAY %{TCR}.T%{TTBR:d}SZ (%{TSIZE:d} bits) is out of range. Must be between 25 and %{TSIZE_MAX:d}. Fields:

TCR enum

Which TCR.

TSIZE signed int

Value of TSize.

TSIZE_MAX signed int

The maximum allowed value.

TTBR unsigned int

Which TTBR region.

ArchMsg.Warning.warning_unpredictable_unaligned_pc_as_base_register

TAGS **UNPREDICTABLE** DISPLAY The PC value must be word-aligned, otherwise the behavior of the instruction is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_unaligned_pop_stack

TAGS **UNPREDICTABLE** DISPLAY **CONSTRAINED UNPREDICTABLE** unaligned stack frame %{FRAME_ADDR:x} in exception-return. Fields:

FRAME_ADDR unsigned int

unaligned stack frame address.

ArchMsg.Warning.warning_unpredictable_unaligned_to_non_writeback

TAGS **UNPREDICTABLE** DISPLAY Unaligned Access to Non-Writeback-Cacheable Memory.

ArchMsg.Warning.warning_unpredictable_vmsa_memattrib

TAGS **UNPREDICTABLE** DISPLAY Unpredictable vmsa memory attribute with texcb value programmed to %{TEXCB} when tex remap is %{TEX_REMAP}. Fields:

TEXCB unsigned int

texcb value.

TEX_REMAP bool

use tex remap.

ArchMsg.Warning.warning_unpredictable_vtcr_t0sz_sl0

TAGS **UNPREDICTABLE** DISPLAY The combination of VTCR.TOSZ=%{TOSZ:d} and VTCR.SLO=%{SLO:d} is unpredictable. Fields:

SLO signed int

VTCR.SLO, starting level for stage 2 translation table walks.

TOSZ signed int

VTCR.TOSZ, required input address range.

ArchMsg.Warning.warning_unpredictable_write_DHCSR

TAGS **UNPREDICTABLE** DISPLAY **UNPREDICTABLE** set of bit-changes in DHCSR. Fields:

NEW unsigned int

data attempted to be written.

OLD unsigned int

previous value.

ArchMsg.Warning.warning_unpredictable_write_SHCSR

TAGS **UNPREDICTABLE** DISPLAY **UNPREDICTABLE** setting a pending bit to 1 for an exception with priority >= execution priority.

ArchMsg.Warning.warning_unsupported_access_to_memory_mapped_register

TAGS **UNPREDICTABLE** DISPLAY Access to memory mapped register is unsupported.

ArchMsg.Warning.warning_user_jmcr_access

TAGS **UNPREDICTABLE** DISPLAY User %{WRITE:(write|read)} access to JMCR is unpredictable. Fields:

WRITE unsigned int

access is write.

ArchMsg.Warning.warning_wcr_mask_and_bas

TAGS **UNPREDICTABLE** DISPLAY DBGWCR_EL1.MASK is non-zero and BAS is not 0xff.

ArchMsg.Warning.warning_wcr_mask_reserved

TAGS **UNPREDICTABLE** DISPLAY DBGWCR_ELn.MASK is set to a reserved value.

ArchMsg.Warning.warning_wcr_non_configuous_bas

TAGS **UNPREDICTABLE** DISPLAY DBGWCR_EL1.BAS has non-contiguous set of ones.

ArchMsg.Warning.warning_wvr_masked_bit_not_zero

TAGS **UNPREDICTABLE** DISPLAY DBGWVRn_EL1.VA contains a bit masked by DBGWCRn_EL1.MASK that is not zero.

ArchMsg.Why.why_branch_target_exception

A branch target exception has occurred. DISPLAY branch target exception: target pc=%{PC}, target opcode=%{OPCODE}. PRIMARY KEY PC. Fields:

BTTYPE unsigned int

pstate.BTYPE at time branch taken.

OPCODE unsigned int

opcode at branch target.

PC unsigned int

Virtual Address of branch target.

ArchMsg.Why.why_illegal_state

An illegal mode change has occurred. DISPLAY illegal mode change: %{MESSAGE}. Fields:

MESSAGE string

The message.

BRANCH_MISPREDICT

Simulating branch mispredict. Fields:

PC unsigned int

Origin address (or 0 if unavailable).

BRA_DIR

Direct branches, to immediate address. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISET enum

The instruction set of the branch instruction.

IS_COND bool

Indicates if this is a conditional branch.

IS_HINTED bool

Whether the conditional instruction is hinted.

IS_ISB bool

Whether the instruction is ISB.

IS_LINK bool

Indicates if this is a branch with link.

PC unsigned int

The address of the branch instruction.

TARGET_ISET enum

The instruction set after the branch.

TARGET_PC unsigned int

The address the instruction branches to.

BRA_INDIR

Indirect branches, perhaps to a register. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISET enum

The instruction set of the branch instruction.

IS_COND bool

Indicates if this is a conditional branch.

IS_HINTED bool

Whether the conditional instruction is hinted.

IS_ISB bool

Whether the instruction is ISB.

IS_LINK bool

Indicates if this is a branch with link.

PC unsigned int

The address of the branch instruction.

TARGET_ISET enum

The instruction set after the branch.

TARGET_PC unsigned int

The address the instruction branches to.

BRA_INDIR_CCFAIL

Indirect branches fail, perhaps to a register.

BRA_RET_CCFAIL

Branch return fail, perhaps to a register.

CACHE_MAINTENANCE_OP

Cache Maintenance Operation. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

DATA unsigned int

Specified MVA or Set/way.

FUNCTION enum

Clean / Invalidate.

SCOPE enum

Affected region.

SIDE enum

Inst / Data.

CCFAIL

Conditional instruction condition check fail. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CCFAIL_UNC

Unconditional instruction condition check fail. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CCPASS

Conditional instruction condition check pass. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CCPASS_UNC

Unconditional instruction condition check pass. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CODE_CACHE_MAINT

Code cache maintenance. Fields:

END_INCL unsigned int

Inclusive end address for VA or PA requests.

NS bool

Non-secure world for PA.

START unsigned int

Start address for VA or PA requests.

TYPE enum

Request type.

COMPILE_BLOCK_END

Last instruction of basic block translated. Fields:

VADDR unsigned int

Address of next instruction after this basic block.

COMPILE_BLOCK_START

First instruction of basic block translated. Fields:

VADDR unsigned int

Address of first instruction in basic block.

COMPILE_INST

ARM instruction compiled. Fields:

DISASS string

Disassembly of the instruction.

ISSET enum

The instruction set of this instruction.

ITSTATE unsigned int

The ITSTATE current for the instruction.

OPCODE unsigned int

The opcode of the instruction.

PC unsigned int

The address of the instruction.

SIZE unsigned int

The size of the instruction in bytes.

CONTEXTIDR

Context ID Register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

NS bool

Secure or nonsecure banked register is accessed.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The new value written.

CONTEXT_SYNC

Called for every context synchronization event. Fields:

ARCH_CSE bool

Architectural Context Synchronization Event.

PC unsigned int

The address of the synchronization instruction.

SYSREG_SYNC bool

System register synchronization event.

CORE_ENDIAN

Core BE8 Big-Endian state changed. Fields:

BE8 bool

Core BE8 Big-Endian state.

CORE_INFO

Static processor attributes. Only triggered by a call to DumpState(). Fields:

ARCH_PROFILE enum

The architecture profile of the core.

CLUSTER_ID unsigned int

The cluster ID of this processor.

CORE_NUM unsigned int

The number of this core in an MP processor.

DETAILED_CORE_FEATURES enum

Detailed core features as a bitfield. Enum values are bitmasks to extract the info from the field.

FPU_VERSION enum

The VFP version implemented by the core.

MEM_ARCH enum

The memory architecture of the core.

MIDR_EL1 unsigned int

MIDR_EL1 value.

NUM_CORES unsigned int

The number of cores in this MP processor.

QUANTUM_SIZE unsigned int

The default quantum size of the core (in ticks).

SECURITY_FEATURES bool

Does the core have security features?.

CORE_LOADS

Processor load accesses. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ACQREL enum

Is this an acquire/release.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DATA unsigned int

The data read or written.

ELEMENT_SIZE unsigned int

Width of each element.

ENDIANNESS enum

Byte ordering of this access.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

LOCK enum

Normal, exclusive or locked access.

MECID unsigned int

Memory Encryption Context ID, if applicable.

MECID2 unsigned int

Memory Encryption Context ID of the second page, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PAGE_SIZE unsigned int

Page size as $\log_2(\text{size in bytes})$.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK, 2=Exclusive Failed, 3=Unsupported.

SIZE unsigned int

Width of the access in bytes. Only required if DATA is not traced.

TRANS bool

Is this a translated access.

VADDR unsigned int

The virtual address of the access.

CORE_REGS

Changes of the core registers R0 to R14. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number, 0 to 14.

MODE enum

Bank of the register accessed.

OLD_VALUE unsigned int

The old value overwritten.

PHYS_ID enum

The physical register accessed.

VALUE unsigned int

The new value written to the register.

CORE_REGS64

Changes of the core registers X0..X30, SP_ELn. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ID enum

The register number, 0-30 for X0-X30, >=32 for SP_ELn.

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

CORE_REGS64_READ

General purpose AArch64 register has been read X0..X30, SP_ELn. Fields:

ID enum

The register number, 0-30 for X0-X30, >=32 for SP_ELn.

VALUE unsigned int

The value read from the register.

CORE_STALLED_TRANSITION

Detect when a core is prevented from progressing PChannel states. Fields:

CURRENT_STATE enum

Current power mode of core.

REASON string

The reason a transition was stalled, and what must be done to unblock it.

TARGET_STATE enum

Desired power mode of core after transition.

CORE_STORES

Processor store accesses. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ACQREL enum

Is this an acquire/release.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DATA unsigned int

The data read or written.

ELEMENT_SIZE unsigned int

Width of each element.

ENDIANNESS enum

Byte ordering of this access.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

LOCK enum

Normal, exclusive or locked access.

MECID unsigned int

Memory Encryption Context ID, if applicable.

MECID2 unsigned int

Memory Encryption Context ID of the second page, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PAGE_SIZE unsigned int

Page size as $\log_2(\text{size in bytes})$.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK, 2=Exclusive Failed, 3=Unsupported.

SIZE unsigned int

Width of the access in bytes. Only required if DATA is not traced.

TRANS bool

Is this a translated access.

VADDR unsigned int

The virtual address of the access.

CP14_READ

System Coprocessor register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP14 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc1 unsigned int

opcode 1.

opc2 unsigned int
opcode 2.

CP14_WRITE

System Coprocessor register write. Fields:

CORE_NUM unsigned int
Core number in a multi processor.

CRm unsigned int
CRm.

CRn unsigned int
CRn.

NS bool
Secure or nonsecure banked register is accessed.

REG_NAME string
Name of the CP14 register accessed.

SECURITY_STATE enum
The current security state.

UNDEF bool
The register accessed is undefined.

UPDATED_VALUE unsigned int
Updated value of the register now it has been written.

VALUE unsigned int
The new value written.

opc1 unsigned int
opcode 1.

opc2 unsigned int
opcode 2.

CP15_READ

System Coprocessor register read. Fields:

CORE_NUM unsigned int
Core number in a multi processor.

CRm unsigned int
CRm.

CRn unsigned int
CRn.

NS bool
Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP15 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc1 unsigned int

opcode 1.

opc2 unsigned int

opcode 2.

CP15_READ64

System Coprocessor register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP15 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc unsigned int

opcode 1.

CP15_WRITE

System Control Coprocessor register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP15 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc1 unsigned int

opcode 1.

opc2 unsigned int

opcode 2.

CP15_WRITE64

System Coprocessor register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP15 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc unsigned int

opcode 1.

CPSR

CPSR change. (In AArch64 PSTATE represented in SPSR format). Fields:

CORE_NUM unsigned int

Core number in a multi processor.

OLD_VALUE unsigned int

The old CPSR value.

UNKNOWN unsigned int

Bits within the register that have unknown value.

VALUE unsigned int

The new CPSR value.

CRYPTO_SPEC

Every crypto instruction speculatively executed.

CorePowerStateChange

Core power-up state has changed. DISPLAY Core powered %{PU:(up|down)}. Fields:

PU bool

Whether the core powered up.

DATA_CACHE_SET_ALLOC_TAGS

Set allocation tags from DC GVA instruction. Fields:

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

NSDESC unsigned int

The security state of the access.

PADDR unsigned int

The physical address of the access.

PAS enum

The physical address space of the page.

RESPONSE enum

0=Aborted, 1=OK.

SIZE unsigned int

The size of memory in bytes.

TAG unsigned int

The value of the tag being set.

TAGGING_TYPE enum

Type of Memory tagging (Physical or Virtual).

TAG_VADDR unsigned int

The virtual address of the Virtual Tag if Virtual tagging is used; otherwise 0.

VADDR unsigned int

Virtual address used for setting allocation tags.

DATA_CACHE_ZERO

Zero and invalidate cache line from DC ZVA instruction. Fields:

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

NSDESC unsigned int

The security state of the access.

PADDR unsigned int

The physical address of the access.

PADDR_TAG unsigned int

If tag checked, the physical address tag for the access.

PAS enum

The physical address space of the page.

RESPONSE enum

0=Aborted, 1=OK.

SIZE unsigned int

The size of zero'd memory in bytes.

TAG_CHECKED bool

Is this access tag checked.

VADDR unsigned int

Virtual address of the zero'd cache line.

DEBUG_EVENT

Hardware debug support event. Fields:

EVENT enum

Description of event.

VALUE unsigned int

data value.

DEBUG_TARGET_ISET

Hardware debug target ISA. Fields:

TARGET_ISET enum

The target instruction set of the debug state.

DMI_ALLOCATE

DMI allocated. Fields:

ACCESS_DIRECTION enum

READ / WRITE.

PADDR unsigned int

The physical address of the DMI.

PTR unsigned int

DMI range pointer.

SIDE enum

DSIDE / ISIDE.

SIZE unsigned int

The size of DMI.

VADDR unsigned int

The virtual address of the DMI..

DMI_HIT

DMI hit, access used DMI (only fire for small DMI regions i.e. less than a full page). Fields:

ACCESS_DIRECTION enum

READ / WRITE.

PADDR unsigned int

The physical address of the DMI.

SIDE enum

DSIDE / ISIDE.

SIZE unsigned int

The size of the DMI.

VADDR unsigned int

The virtual address of the DMI..

DMI_REVOKE

DMI revoked. Fields:

ACCESS_DIRECTION enum

READ / WRITE.

PADDR unsigned int

The physical address of the DMI.

SIDE enum

DSIDE / ISIDE.

SIZE unsigned int

The size of DMI.

VADDR unsigned int

The virtual address of the DMI..

END_COMPILE

Compilation end. Fields:

END_OF_PAGE_COUNT unsigned int

Number of basic blocks exited due to page boundary since START_COMPILE.

FETCHFAIL_COUNT unsigned int

Number of basic blocks exited due to ifetch failure START_COMPILE.

INST_COUNT unsigned int

Number of instructions compiled since START_COMPILE.

NONSEQ_COUNT unsigned int

Number of basic blocks exited due to non-sequential instructions since START_COMPILE.

PAGE_STRADDLE_COUNT unsigned int

Number of basic blocks exited due to unaligned instructions crossing page since START_COMPILE.

ESB

Error Synchronization Barrier hit. Fields:

ASYNC bool

ESB is async.

IMPLICIT bool

ESB is implicit.

EXCEPTION

Exception taken. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ESR unsigned int

The value assigned to ESR when exception is taken.

ESR_EC unsigned int

The Value assigned to ESR.EC when exception is taken.

IS_PHYSICAL bool

Physical or Virtual exception.

IS_TRAP bool

Exception coming from a trap.

LR unsigned int

The value assigned to the link register.

PC unsigned int

The location where the exception occurred.

PREFERRED_RETURN unsigned int

The preferred return address for the exception.

TAKEN_LOCALLY bool

Whether exception is taken locally(same exception level).

TARGET_ISET enum

The instruction set of the exception handler code.

TARGET_PC unsigned int

The address the exception branches to.

VECTOR enum

The exception vector.

VECTOR_OFFSET unsigned int

The offset of an exception vector from the base address.

EXCEPTION_END

Every exception completed.

EXCEPTION_INFO

Trace human-readable reason of the current exception for data/instruction aborts, floating point exceptions and unknown exceptions. Fields:

DESCRIPTION string

Reason for current exception.

ESR_EC unsigned int

The value of ESR.EC when exception is taken.

EXCEPTION_RAISE

Every exception raised.

EXCEPTION_RETURN

Branches on leaving exception. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISSET enum

The instruction set of the branch instruction.

IS_COND bool

Indicates if this is a conditional branch.

IS_HINTED bool

Whether the conditional instruction is hinted.

IS_ISB bool

Whether the instruction is ISB.

IS_LINK bool

Indicates if this is a branch with link.

PC unsigned int

The address of the branch instruction.

TARGET_ISET enum

The instruction set after the branch.

TARGET_PC unsigned int

The address the instruction branches to.

EXCEPTION_RETURN_PREBRANCH

Exception return event that must occur before PC and processor state have updated.

EXCEPTION_START

Every exception started.

FIQ_TAKEN

FIQ taken exception.

FP_EXCEPTIONS_SIGNALLED

Trace which FP exceptions were signalled as a part of the FP operations performed by an instruction. Fields:

FP_EXCEPTIONS_SIGNALLED_MASK unsigned int

Indicates which FP exceptions were signalled by an instruction. This uses the same encoding as the cumulative floating-point exception bitfields in FPSR.

FP_STATE

Floating point state. Fields:

FPSR unsigned int

FPSR register value.

MASK unsigned int

The mask indicating updated cumulative bits.

FREQ_CHANGED

Core frequency changed. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The instruction count of this core.

NEW_FREQ signed int

The new frequency of this core (expressed in Hz).

OLD_FREQ signed int

The old frequency of this core (expressed in Hz).

HLT

HLT instruction occurred. Fields:

IMM16 unsigned int

The 16-bit immediate value encoded in HLT instruction.

INST

Every instruction executed. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DEBUG_STATE bool

The instruction is executed in debug state.

DISASS string

Disassembly of instruction.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISSET enum

The current instruction set.

ITSTATE unsigned int

The current ITSTATE.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

MODE enum

The mode the core is in.

NS unsigned int

The core's non-secure bit.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

OPCODE unsigned int

The opcode of the instruction.

PADDR unsigned int

The physical address of the instruction.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the instruction.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the page.

PC unsigned int

The address of the instruction.

SECURITY_STATE enum

The core's security state.

SIZE unsigned int

The size of the instruction in bytes.

INST_END

Every instruction completed.

INST_START

Every instruction started. Fields:

DEBUG_STATE bool

The instruction is executed in debug state.

ISSET enum

The current instruction set.

MODE enum

The mode the core is in.

NS enum

The current Secure State.

PC unsigned int

The address of the conditional instruction.

SECURITY_STATE enum

The current Security State.

INST_STRADDLE

Instruction straddles boundary. Fields:

DEBUG_STATE bool

The instruction is executed in debug state.

ISSET enum

The current instruction set.

MODE enum

The mode the core is in.

NS enum

The current Secure State.

PC unsigned int

The address of the conditional instruction.

SECURITY_STATE enum

The current Security State.

IRQ_TAKEN

IRQ taken exception.

ISB

ISB instruction occurred. Fields:

PC unsigned int

The address of the ISB instruction.

LOAD_MULTIPLE_REGS_LIST

List of destination registers (not source) of a multiple registers load instruction. Fields:

REG string

Register to add into the list.

LOCAL_MONITOR

Local monitor activity. Fields:

PADDR unsigned int

Local Monitor Address.

State enum

State of the monitor (Open/Exclusive).

MAPPING_CREATE

A memory mapping has been created. This is an implementation detail of the model and so may change without notice. Fields:

ASID unsigned int

The ASID used during translation, or 0 if no ASID was used.

IS_VALID bool

Whether the mapping is valid or not.

PADDR unsigned int

Base physical address of region, or 0 if not known.

SIDE enum

I-side or D-side.

SIZE unsigned int

Size of the page, or 0 if not known.

VADDR unsigned int

Base virtual address of region.

VMID unsigned int

The VMID used during translation, or 0 if no VMID was used.

MAPPING_DESTROY

A memory mapping has been destroyed. This is an implementation detail of the model and so may change without notice. Fields:

ASID unsigned int

The ASID used during translation.

CAUSE enum

The cause of invalidation.

PADDR unsigned int

Base physical address of page.

SIDE enum

I-side or D-side.

SIZE unsigned int

Size of the page.

VADDR unsigned int

Base virtual address of page.

VMID unsigned int

The VMID used during translation.

MEMMAP_DEBUG_READ

Memory mapped reads to the debug registers. Fields:

ADDR unsigned int

address.

CORE_NUM unsigned int

Core number in a multi processor.

EXT bool

Whether access is from an external device (such as the DAP).

NS enum

Secure state of the access.

REG_NAME string

Name of the debug register accessed.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

MEMMAP_DEBUG_WRITE

Memory mapped writes to the debug registers. Fields:

ADDR unsigned int

address.

CORE_NUM unsigned int

Core number in a multi processor.

EXT bool

Whether access is from an external device (such as the DAP).

NS enum

Secure state of the access.

REG_NAME string

Name of the debug register accessed.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The new value written.

MMU_D128_TTB_READ

This event is triggered by reads caused by a D128 translation table walk. Fields:

ABORTED bool

Was the walk successful.

DATA_HIGH unsigned int

The data upper bits read.

DATA_LOW unsigned int

The data lower bits read.

IPA unsigned int

For stage 1, the IPA of the read.

LEVEL unsigned int

Translation table level.

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

OUTADDR unsigned int

The output physical address.

PADDR unsigned int

The physical address of the read.

PASmem enum

Physical Address Space of the accessed memory.

PASreq enum

Physical Address Space of the requested lookup.

REGIME_EL enum

Entry matches in this translation regime.

REQUESTER enum

The requester of this table walk.

SIDE enum

Inst / Data.

STAGE unsigned int

Translation stage.

MMU_D128_TTB_WRITE

This event is triggered by writes caused by a D128 translation table walk. Fields:

ABORTED bool

Was the walk successful.

DATA_HIGH unsigned int

The data upper bits written.

DATA_LOW unsigned int

The data lower bits written.

IPA unsigned int

For stage 1, the IPA of the write.

IS_HACDBS_ACCESS bool

If the access was due to HACDBS_CLEAN.

LEVEL unsigned int

Translation table level.

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

OUTADDR unsigned int

The output physical address.

PADDR unsigned int

The physical address of the write.

PASmem enum

Physical Address Space of the accessed memory.

PASreq enum

Physical Address Space of the requested lookup.

REGIME_EL enum

Entry matches in this translation regime.

SIDE enum

Inst / Data.

STAGE unsigned int

Translation stage.

MMU_TRANS

Address translation information. Fields:

ASID unsigned int

Address space identifier.

CORE_NUM unsigned int

Core number in a multi processor.

Hyp bool

Entry matches in Hyp state only.

INNERCACHE_RA bool

Is the inner cache allocate on read.

INNERCACHE_TRANSIENT bool

Is the inner write-through transient.

INNERCACHE_TYPE enum

Inner Caching scheme (NC/MB/WA).

INNERCACHE_WA bool

Is the inner cache allocate on write.

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MEMTYPE enum

Memory type.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC enum

Is secure side supposed to access secure or nonsecure memory.

OUTERCACHE_RA bool

Is the outer cache allocate on read.

OUTERCACHE_TRANSIENT bool

Is the outer write-through transient.

OUTERCACHE_TYPE enum

Outer Caching scheme (NC/MB/WA).

OUTERCACHE_WA bool

Is the outer cache allocate on write.

PADDR unsigned int

Physical address of the access.

PAGESIZE unsigned int

Page size as log2(size).

PAS enum

PAS supposed to be accessed is secure or nonsecure memory or Root/Realm if RME is implemented.

SH enum

Shareability.

SIDE enum

Inst / Data.

STAGE1_PERM unsigned int

Stage 1 permissions, [5:3] Privileged access (XWR) [2:0] User access (XWR).

STAGE2_PERM unsigned int

Stage 2 permission mask [5:3] Privileged access (XWR) [2:0] User access (XWR).

VADDR unsigned int

Virtual address of the access.

VMID unsigned int

Virtual machine identifier.

XS bool

XS Attribute.

nG enum

Flag indicating whether ASID will be matched.

MMU_TTB_READ

This event is triggered by reads caused by a translation table walk. Fields:

ABORTED bool

Was the walk successful.

DATA unsigned int

The data read.

DESC_KIND enum

The encoding scheme of the descriptors used for this walk.

IPA unsigned int

For stage 1, the IPA of the read.

LEVEL unsigned int

Translation table level.

LPAE bool

Is this for an LPAE translation (*DEPRECATED*, use DESC_KIND).

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSmem bool

Non secure memory access.

NSreq bool

Non secure request.

OUTADDR unsigned int

The output physical address.

PADDR unsigned int

The physical address of the read.

PASmem enum

Physical Address Space of the accessed memory.

PASreq enum

Physical Address Space of the requested lookup.

REGIME_EL enum

Entry matches in this translation regime.

REQUESTER enum

The requester of this table walk.

SIDE enum

Inst / Data.

STAGE unsigned int

Translation stage.

MMU_TTB_WRITE

This event is triggered by writes caused by a translation table walk. Fields:

ABORTED bool

Was the walk successful.

DATA unsigned int

The data written.

DESC_KIND enum

The encoding scheme of the descriptors used for this walk.

IPA unsigned int

For stage 1, the IPA of the write.

IS_HACDBS_ACCESS bool

If the access was due to HACDBS_CLEAN.

LEVEL unsigned int

Translation table level.

LPAE bool

Is this for an LPAE translation (*DEPRECATED*, use DESC_KIND).

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSmem bool

Non secure memory access.

NSreq bool

Non secure request.

OUTADDR unsigned int

The output physical address.

PADDR unsigned int

The physical address of the write.

PASmem enum

Physical Address Space of the accessed memory.

PASreq enum

Physical Address Space of the requested lookup.

REGIME_EL enum

Entry matches in this translation regime.

SIDE enum

Inst / Data.

STAGE unsigned int

Translation stage.

MODE_CHANGE

Mode change. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

MODE enum

The new mode.

NON_SECURE enum

The core's new non-secure bit.

OLD_MODE enum

The old mode.

SECURITY_STATE enum

The new security state.

NOT_TAG_CHECKED_INST

MTE Tag not checked for current instruction.

PERIODIC

Called for every quantum. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The instruction count of this CPU.

PC unsigned int

The address of the next instruction to be executed on this CPU.

PMU_COUNTER_OVERFLOW

PMU counter overflow. Fields:

COUNTER_GROUP enum

Counter group.

COUNTER_RANGE enum

Counter range.

EVENT_ID unsigned int

Event Identifier, field is valid only for Event counter group.

INDEX unsigned int

Counter index (as selected by PMSELR), field is valid only for Event/Cycle counter groups.

INTERRUPT bool

Is interrupt enabled on overflow for this counter.

POTINDEX_UPDATE

Value of POT index for current pc changed. Fields:

NEW_POTINDEX unsigned int

New POTIndex.

OLD_POTINDEX unsigned int

Old POTIndex.

PREFETCH_MEMORY64

Prefetch from PRFM or PRFUM instructions. Fields:

PRFOP enum

Prefetch hint.

VADDR unsigned int

Virtual address of the location that should be prefetched.

PRELOAD64_REQUEST

Preload requests from PRFM or PRFUM instructions. Fields:

PRFOP enum

Prefetch hint.

VADDR unsigned int

Virtual address of the location that should be prefetched.

PRELOAD_DATA

Data preload from PLD instruction. Fields:

VADDR unsigned int

Virtual address of the data that should be preloaded.

PRELOAD_INST

Instruction preload from PLI instruction. Fields:

VADDR unsigned int

Virtual address of the instruction that should be preloaded.

PRE_CORE_LOAD

Trace just before a core load. Fields:

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

PRE_CORE_STORE

Trace just before a core store. Fields:

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

PRE_TTB_READ

This event is triggered before reads caused by a translation table walk. Fields:

ADDR unsigned int

The physical address of the read.

PSTATE

PSTATE fields update. Fields:

ALLINT unsigned int

All interrupt mask bit.

BTYPE unsigned int

Branch target identification bit.

CORE_NUM unsigned int

Core number in a multi processor.

DAIF unsigned int

Exception mask bits.

DIT unsigned int

Data Independent Timing bit.

EL unsigned int

Current Exception level field.

EXLOCK unsigned int

Exception return state lock bit.

IL unsigned int

Illegal Execution state bit.

NZCV unsigned int

Condition flags.

PACM unsigned int

PACM bit.

PAN unsigned int

Privileged Access Never state bit.

PM unsigned int

PMU exception mask bit.

PPEND unsigned int

PMU exception pending bit.

SM unsigned int

StreamingMode status bit.

SP unsigned int

Stack pointer register selection bit.

SS unsigned int

Software Step bit.

SSBS unsigned int

Speculative Store Bypass Safe bit.

TCO unsigned int

Tag Check Override bit.

TOUCHED unsigned int

Mask of the fields that have been touched since this trace has last been emitted. Note that touched does not necessarily mean the value changed. The index of the field is the bit index in the mask plus the index of this field plus one.

UAO unsigned int

User Access Override bit.

UINJ unsigned int

Undefined Instruction exceptions bit.

ZA unsigned int

ZA storage and ZT0 register enable bit.

nRW unsigned int

Current Execution state bit.

PSTATE_PACTIVE_OPMODE_TRANSITION

Trace pactive and Opmode mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Operating mode after the transition.

OLD_STATE enum

Operating mode before the transition.

PSTATE_PACTIVE_OPMODE_TRANSITION_DENIED

Trace the denied pactive and Opmode mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Operating mode after the transition.

OLD_STATE enum

Operating mode before the transition.

REASON string

Reason for the transition being denied.

PSTATE_PACTIVE_POWER_TRANSITION

Trace pactive and power mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Power mode after the transition.

OLD_STATE enum

Power mode before the transition.

PSTATE_PACTIVE_POWER_TRANSITION_DENIED

Trace the denied pactive and power mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Power mode after the transition.

OLD_STATE enum

Power mode before the transition.

REASON string

Reason for the transition being denied.

PSTATE_PACTIVE_THREAD_OPMODE_TRANSITION

Trace pactive and Opmode mode transitions for threads on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Operating mode after the transition.

OLD_STATE enum

Operating mode before the transition.

QUANTUM_START

Called at the start of every quantum.

RANGE_PRELOAD_REQUEST

Preload requests from RPRFM. Fields:

COUNT unsigned int

Number of blocks of data to access.

LENGTH unsigned int

Number of contiguous bytes to be accessed in each block.

OP unsigned int

Range prefetch operation .

REUSE unsigned int

Maximum number of bytes to be accessed before next RPRFM .

STRIDE signed int

Stride to advance after accessing LENGTH bytes.

VADDR unsigned int

Base virtual address of range prefetch request.

RUN_STATE

Run state transition. Fields:

INST_COUNT unsigned int

Ticks count at point of transition.

NEW enum

New run state.

OLD enum

Old run state.

SEMIHOSTING_CALL

Call of a semihost function occurred. Fields:

PC unsigned int

The program counter after the semihosting call.

SEMIHOSTING_PRECALL

About to call semihost. Fields:

EL2 **bool**

Use EL2 translation regime.

NS **bool**

Is Non-Secure.

REG_WIDTH **unsigned int**

The current register width in bytes.

SIGNAL

External signal state change. Fields:

SIGNAL **enum**

Signal that changed.

STATE **bool**

Signal asserted state.

SOFTWARE_STEP

Return the Debug Software Step state. Fields:

STATE **enum**

Software Step state.

SPSR

SPSR change. Fields:

CORE_NUM **unsigned int**

Core number in a multi processor.

MODE **enum**

Which of the banked SPSR registers is written.

OLD_VALUE **unsigned int**

The old SPSR value.

VALUE **unsigned int**

The new SPSR value.

START_COMPILE

Compilation started. Fields:

VADDR **unsigned int**

Instruction where compilation begins.

STG_MULTIPLE_MICRO_OPS_ADDRESS_RANGE

Address range for MTE stores with multiple uops. Fields:

ADDR_END **unsigned int**

Final address.

ADDR_START unsigned int

Starting address.

STORE_MULTIPLE_REGS_ADDRESS_RANGE

Address range for stores with multiple registers. Fields:

ADDR_END unsigned int

Final address.

ADDR_START unsigned int

Starting address.

SUBOPTIMAL_LDST_RETIRED

suboptimal load/store. Fields:

ACCESS_SIZE unsigned int

Log2 of the access width in bytes used for alignment checking (1: half-word, 2: word ...).

OPTIMAL_ALIGN_SELECTOR unsigned int

Selector to decide the optimal alignment size (1: access width, 2: 4-bytes, 3: 8-bytes, ..., 12: 4-kbytes).

VADDR unsigned int

The virtual address of the access.

SYNC

Called for every synchronization, that is every quantum break. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The instruction count of this CPU.

LOCAL_QUANTUM unsigned int

The local quantum of this CPU.

LOCAL_TIME unsigned int

The local time of this CPU.

SYNC_REASON enum

Reason for sync.

SYSCALL

System call instruction executed. Fields:

IMM unsigned int

Immediate value of the system call instruction.

TYPE enum

System call type.

VADDR unsigned int

Instruction that caused the system call.

SYSREG_READ128

System Coprocessor 128 bit register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

REGNUM enum

Internal register number.

REGNUM_PHYS enum

Internal register number physically accessed (e.g. in presense of VHE redirection).

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_READ64

System Coprocessor register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REGNUM enum

Internal register number.

REGNUM_OPERAND enum

Register number corresponding to the instruction operand.

REGNUM_PHYS enum

Internal register number physically accessed (e.g. in presense of VHE redirection).

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_UPDATE128

Triggers when the system updates a register. Fields:

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG enum

Register number.

UNKNOWN unsigned int

Bits of the register which became unknown.

VALUE unsigned int

Value written to the register.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_UPDATE32

Triggers when the system updates a register. Fields:

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG enum

Register number.

UNKNOWN unsigned int

Bits of the register which became unknown.

VALUE unsigned int

Value written to the register.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_UPDATE64

Triggers when the system updates a register. Fields:

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG enum

Register number.

UNKNOWN unsigned int

Bits of the register which became unknown.

VALUE unsigned int

Value written to the register.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int
opcode 2.

SYSREG_WRITE128

System Coprocessor 128 bit register write. Fields:

CORE_NUM unsigned int
Core number in a multi processor.

CRm unsigned int
CRm.

CRn unsigned int
CRn.

REGNUM enum
Internal register number.

REGNUM_PHYS enum
Internal register number physically accessed (e.g. in presense of VHE redirection).

UNDEF bool
The register accessed is undefined.

UPDATED_VALUE unsigned int
Updated value of the register now it has been written.

VALUE unsigned int
The new value written.

opc unsigned int
opcode 1.

opc0 unsigned int
opcode 0.

opc2 unsigned int
opcode 2.

SYSREG_WRITE64

System Coprocessor register write. Fields:

CORE_NUM unsigned int
Core number in a multi processor.

CRm unsigned int
CRm.

CRn unsigned int
CRn.

NS bool
Secure or nonsecure banked register is accessed.

REGNUM enum

Internal register number.

REGNUM_OPERAND enum

Register number corresponding to the instruction operand.

REGNUM_PHYS enum

Internal register number physically accessed (e.g. in presense of VHE redirection).

UNDEF bool

The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

UNALIGNED_LDST_RETIRED

Processor unaligned load/store. Fields:

ACCESS_SIZE unsigned int

Log2 of the access width in bytes used for alignment checking (1=>half-word, 2->word ...).

NUMBER_OF_BEATS unsigned int

Number of accesses (beats) in this burst (Total size in bytes is ACCESS_SIZE * NUMBER_OF_BEATS).

VADDR unsigned int

The virtual address of the access.

VFP_D_REGS

VFP/NEON D 64 bit register write. Fields:

ALIASING enum

Type of register aliasing used.

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a write to the corresponding bit in the register. For example, 0x1 would indicate a write to VALUE[0].

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

VFP_D_REGS_READ

VFP/NEON D 64 bit register read. Fields:

ID unsigned int

The register number.

VALUE unsigned int

The current value of the register.

VFP_Q_REGS

VFP/NEON Q 128 bit register write. Fields:

ALIASING enum

Type of register aliasing used.

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a write to the corresponding byte in the register. For example, 0x1 would indicate a write to VALUE[7:0].

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

VFP_Q_REGS_READ

VFP/NEON Q 128 bit register read. Fields:

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a read to the corresponding byte in the register. For example, 0x1 would indicate a read to VALUE[7:0].

VALUE unsigned int

The current value of the register.

VFP_SYS_REGS

Writes to the VFP/NEON units system registers. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ID enum

Which VFP system register is written.

MASK unsigned int

The mask indicating updated cumulative bits.

OLD_VALUE unsigned int

The register's old value overwritten.

VALUE unsigned int

The new value written to the VFP system register.

VFP_S_REGS

VFP/NEON S 32 bit register write. Fields:

ALIASING enum

Type of register aliasing used.

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a write to the corresponding bit in the register. For example, 0x1 would indicate a write to VALUE[0].

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

VFP_S_REGS_READ

VFP/NEON S 32 bit register read. Fields:

ID unsigned int

The register number.

VALUE unsigned int

The current value of the register.

VIRTUAL_TAG_ALLOCATED

Virtual tag information when allocated. Fields:

TAG_VADDR unsigned int

Virtual tag virtual address.

VADDR unsigned int

Data Virtual address.

VIRTUAL_TAG unsigned int

Virtual tag value.

WAYPOINT

Signals end of basic block execution. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ISET enum

Origin instruction set.

IS_COND bool

Indicates if this is a conditional waypoint.

PC unsigned int

Origin address (or 0 if unavailable).

TAKEN bool

Indicates if this waypoint was taken.

TARGET unsigned int

Destination address.

TARGET_ISET enum

Destination instruction set.

WFE_END

WFE ended. Fields:

INST_COUNT unsigned int

Ticks count when leaving WFE.

WFE_EVENT_REGISTER

WFE event register status: set/clear, reason. Fields:

INST_COUNT unsigned int

Ticks count.

REASON enum

Reason for set/clear. Only REASON==0 (Cleared by WFE) clears the bit, all other reasons set it.

WFE_IGNORED

WFE ignored. Fields:

EVENT bool

This WFE was ignored because the event register was set.

INST_COUNT unsigned int

Ticks count when ignoring WFE.

TRAPPED bool

This WFE was trapped.

WFE_START

WFE entered. Fields:

INST_COUNT unsigned int

Ticks count when entering WFE.

WFI_END

WFI ended. Fields:

INST_COUNT unsigned int

Ticks count when leaving WFI.

WFI_IGNORED

WFI ignored. Fields:

DISABLED bool

This WFI was ignored because WFI is disabled.

INST_COUNT unsigned int

Ticks count when ignoring WFI.

REASON enum

specifies reason why WFI trace was ignored.

TRAPPED bool

This WFI was trapped.

WFI_START

WFI entered. Fields:

INST_COUNT unsigned int

Ticks count when entering WFI.

WFI_WAKEUP

WFI wakeup. Fields:

INST_COUNT unsigned int

Ticks count when WFI wakeup occurred.

REASON enum

Reason for wakeup.

2.23 ARM_Cortex-A65AE

This section describes the trace sources.

AA64_ASE_SVE_REGS

Modification of AArch64 Advanced SIMD or SVE Vector Registers (via SIMD or Scalar operations), SVE Predicate Registers and ZT0. Fields:

ID **enum**

The register identifier.

MASK **unsigned int**

Bitmask of the register to signal the modified bits in the VALUE field.

SM **bool**

Whether the PE is in Streaming Mode.

VALUE **unsigned int**

Value of AArch64 vector register.

ASYNC_MEMORY_FAULT

Context ID Register write. Fields:

FAULT **unsigned int**

Fault status present in ESR.ISS encoding for Serror in ARCH64 and present in DFSR encoding for external abort format in ARCH32.

PADDR **unsigned int**

Physical Address (or 0 if unavailable).

VADDR **signed int**

Virtual Address (or 0 if unavailable).

ATOMIC_END_ACCESS

Bus trace access for atomics. Fields:

ACCESS_FAIL **bool**

Memory access failed.

ADDR **unsigned int**

The virtual address of the access.

ATTR **unsigned int**

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

COMPARE_VALUE **unsigned int**

Compare value for CAS.

LOAD_VALUE unsigned int

Loaded values.

NSDESC unsigned int

The security state of the access.

OPERAND_VALUE unsigned int

Operation's operand.

OPERATION enum

Operation type.

PADDR unsigned int

The physical address of the access.

PRIV bool

Is this a privileged access?.

ATOMIC_START_ACCESS

Bus trace access for atomics. Fields:

ADDR unsigned int

The virtual address of the access.

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

COMPARE_VALUE unsigned int

Compare value for CAS.

NSDESC unsigned int

The security state of the access.

OPERAND_VALUE unsigned int

Operation's operand.

OPERATION enum

Operation type.

PADDR unsigned int

The physical address of the access.

PRIV bool

Is this a privileged access?.

ArchMsg.Error.exit_code

DISPLAY exit code: %{EXITCODE} component kind: %{KIND} reason: %{REASON}. Fields:

EXITCODE signed int

Exit code to be returned.

KIND string

Component kind that invoked the exit code trace.

REASON string

Reason why the exit code trace was invoked (optional).

ArchMsg.Warning.IgnoredWarmResetRequest

Event fires when software attempts to request warm reset through write to deprecated DBGPRCR.CWRR.

ArchMsg.Warning.RAS_ErrselrUnpredictableState

DISPLAY ERRSELR is in unpredictable state due to value greater than number_of_error_records written to it.

ArchMsg.Warning.RAS_UnpredictableStateWrite

DISPLAY %{REG_NAME} write ignored due to unpredictable ERRSELR state. Fields:

REG_NAME string

Name of register being written.

ArchMsg.Warning.ReportedRevisionRegisterRead

DISPLAY Reported revision and patch do not match that of the cpu. Fields:

ORIGINAL_VALUE unsigned int

The value the register would have based on the modelled revision.

REGISTER string

The name of the register that has ben read.

REPORTED_VALUE unsigned int

The value the register has been overridden to.

ArchMsg.Warning.Unpredictably Indexed PM Event Register

TAGS UNPREDICTABLE DISPLAY Event %{IsDirect:(Selector value|Register)} %{SEL} is >= %{N}. Fields:

IsDirect unsigned int

Direct Access rather than indirect.

N unsigned int

Number of Accessible Registers.

SEL unsigned int

Selector.

ArchMsg.Warning.cache_contents_unknown

A cache has been activated without being invalidated and may contain **UNKNOWN** content. DISPLAY %{SIDE:(D|I)}-Cache was enabled in %{REGIME:(SECURE|NON_SECURE|HYP)} regime but was not invalidated since power-on: cache lines could contain **UNKNOWN** content %{SIDE:(Note: D-side caches must invalidate SECURE lines even if the cache is only used by NON_SECURE code in order to ensure that unexpected dirty lines tagged SECURE are not naturally evicted)}. Fields:

REGIME unsigned int

Current translation regime.

SIDE unsigned int

0 d-side cache or unified cache, 1 i-side.

ArchMsg.Warning.dcimvac_matches_watchpoint

TAGS **UNPREDICTABLE** DISPLAY Watchpoints matching an AArch32 DCIMVA are implementation defined.

ArchMsg.Warning.decode_fieldmismatch

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with different fields %{FIELD1} and %{FIELD2}. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_fracbitsnegative

TAGS **UNPREDICTABLE** DISPLAY VCVT Instruction is unpredictable with a negative number of fraction bits.

ArchMsg.Warning.decode_initblock

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable in an IT block.

ArchMsg.Warning.decode_initblocknotlast

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable in an IT block when not the last.

ArchMsg.Warning.decode_invalid_condition

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable when the condition is not AL.

ArchMsg.Warning.decode_invalidfieldcombination

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with the values in %{FIELD1} and %{FIELD2}. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_invalidvalue

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with the value in %{FIELD}. Fields:

FIELD enum

Field name.

ArchMsg.Warning.decode_mem_hint_unallocated

TAGS **UNPREDICTABLE** DISPLAY There is no memory hint allocated to this bit pattern.

ArchMsg.Warning.decode_registermatch

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with %{FIELD1} and %{FIELD2} as the same register. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_registermismatch

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with %{FIELD1} and %{FIELD2} as different registers. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_registeroutofrange

TAGS **UNPREDICTABLE** DISPLAY Floating-point instruction is unpredictable indexing beyond the end of the register bank.

ArchMsg.Warning.decode_sbzsbo

TAGS **UNPREDICTABLE** DISPLAY Reserved bits in the instruction are not canonical.

ArchMsg.Warning.decode_transactiontoobig

TAGS **UNPREDICTABLE** DISPLAY Attempt to perform a memory transaction of more than 128 bytes.

ArchMsg.Warning.decode_unpred_other

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable.

ArchMsg.Warning.decode_unpreduseofpc

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable when %{FIELD} is PC. Fields:

FIELD enum

Field name.

ArchMsg.Warning.decode_unpreduseofr13

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable when %{FIELD} is R13. Fields:

FIELD enum

Field name.

ArchMsg.Warning.decode_unpreduseofr13inreglist

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with R13 in register-list.

ArchMsg.Warning.decode_useofd16orhigher

TAGS **UNPREDICTABLE** DISPLAY Attempt to use register D16 or higher where it is not allowed.

ArchMsg.Warning.decode_writebackandbaseinlist

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with write back when the base register is in the transfer list.

ArchMsg.Warning.decode_zeroregistersinlist

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable when no registers are to be transferred.

ArchMsg.Warning.recursive_branch

An instruction is performing a branch that targets the same instruction. If this was intended then a WFI might be more effective DISPLAY recursive branch detected at PC=%{PC}. PRIMARY KEY PC. Fields:

PC unsigned int

Address of instruction causing the branch.

ArchMsg.Warning.recursive_exception

An instruction has generated an exception that targets the same instruction. DISPLAY recursive %{TYPE} exception detected at PC=%{PC} (CPSR=%{CPSR}). PRIMARY KEY PC. Fields:

CPSR unsigned int

Processor state.

PC unsigned int

Address of instruction causing the exception.

TYPE enum

Type of exception.

ArchMsg.Warning.reentrant_vector_catch

TAGS **UNPREDICTABLE** DISPLAY Vector catch debug event on Prefetch or Data abort vector %{ADDR}. Fields:

ADDR unsigned int

vector address.

ArchMsg.Warning.reserved_it_state

TAGS **UNPREDICTABLE** DISPLAY Execution with reserved IT state %{ITSTATE}. Fields:

ITSTATE unsigned int

reserved IT state.

ArchMsg.Warning.tlb_contents_unknown

Translation has been enabled without invalidating the TLB. DISPLAY Translation was enabled in the %{PAS:(Secure|Non-secure|Root|Realm)} EL%{EL:d} translation regime (vmid=%{VMID:x}) without invalidating %{INVALIDITY:(|d-side |i-side |)}entries since power-on: %{INVALIDITY:(|D|I|)}TLB could contain **UNKNOWN** content. Fields:

EL unsigned int

Exception level of translation regime.

INVALIDITY unsigned int

Which TLB may be invalid (if TLBs are separate).

NON_SECURE bool

Whether the translation regime is non-secure.

PAS unsigned int

Physical address space of translation regime.

VMID unsigned int

Current VMID, if applicable.

ArchMsg.Warning.unknown_DLR_DSPSR

Exiting debug state with unknown DLR or DSPSR.

ArchMsg.Warning.unknown_ELR_SPSR

Returning from debug exception with unknown ELR or SPSR.

ArchMsg.Warning.unpred_FAR_on_LRCPC3_non_atomic_mem_access

TAGS **UNPREDICTABLE** Unpredictable FAR/FSTATUS on LRCPC3 pair instructions resulting in two separate memory accesses
DISPLAY Unpredictable FAR on LRCPC3 non atomic memory access at %{ADDR1} with FSTATUS %{FSTATUS1} or %{ADDR2} with FSTATUS %{FSTATUS2}. Fields:

ADDR1 unsigned int

Possible FAR address 1.

ADDR2 unsigned int

Possible FAR address 2.

FSTATUS1 unsigned int

Possible fault status value 1.

FSTATUS2 unsigned int

Possible fault status value 2.

ArchMsg.Warning.unpred_async_tag_check_fail_on_sve_speculative_load_do_update_ffr

Unpredictable set to FALSE of the FFR element associated to SVE speculative load with an async tag check fail TAGS **UNPREDICTABLE**.

ArchMsg.Warning.unpredictable_a32_breakpoint

TAGS **UNPREDICTABLE** DISPLAY Unpredictable breakpoint on a A32 instruction at %{ADDR1} Offset %{OFFSET} BAS %{BAS} for a %{IS_ADDRESS_MISMATCH_BREAKPOINT:(Mismatch|Match)} Breakpoint number %{N:d} Breakpoint address %{ADDR2}. Fields:

ADDR1 unsigned int

Address.

ADDR2 unsigned int

Breakpoint address.

BAS unsigned int

BAS field.

IS_ADDRESS_MISMATCH_BREAKPOINT bool

Is a Mismatch breakpoint, not a Match breakpoint.

N unsigned int

Breakpoint number.

OFFSET unsigned int

Always has a value of zero.

ArchMsg.Warning.unpredictable_banked_register_access

TAGS **UNPREDICTABLE** DISPLAY Access to banked register (R=%{R:d} SYSm=%{SYSM:2x}) is unpredictable. Fields:

R unsigned int

R bit.

SYSM unsigned int

SYSm.

ArchMsg.Warning.unpredictable_ccfail_undef

TAGS **UNPREDICTABLE** DISPLAY An instruction %{OPCODE} at VA=%{VADDR} failed its condition codes check but would otherwise have caused a non-data-dependent trap or undefined exception. Behaviour is implementation-defined choice of exception or nop. Fields:

OPCODE unsigned int

opcode of the instruction.

VADDR unsigned int

Virtual address of the instruction being executed.

ArchMsg.Warning.unpredictable_ici_bits

TAGS **UNPREDICTABLE** DISPLAY ICI bits do not apply or are architecturally unpredictable for instruction.

ArchMsg.Warning.unpredictable_t32_breakpoint

TAGS **UNPREDICTABLE** DISPLAY Unpredictable breakpoint on a T32 instruction at %{ADDR1} Offset %{OFFSET} BAS %{BAS} for a %{IS_ADDRESS_MISMATCH_BREAKPOINT:(Mismatch|Match)} Breakpoint number %{N:d} Breakpoint address %{ADDR2}. Fields:

ADDR1 unsigned int

Address.

ADDR2 unsigned int

Breakpoint address.

BAS unsigned int

BAS field.

IS_ADDRESS_MISMATCH_BREAKPOINT bool

Is a Mismatch breakpoint, not a Match breakpoint.

N unsigned int

Breakpoint number.

OFFSET unsigned int

Indicates which halfword of a 32-bit Thumb instruction the breakpoint is set on. 0 for the first halfword or if the instruction is 16 bits wide, or 1 for the second halfword.

ArchMsg.Warning.unpredictable_t32_vector_catch

TAGS **UNPREDICTABLE** DISPLAY Unpredictable Vector catch exception on a T32 instruction at %{ADDR} Offset %{OFFSET}. Fields:

ADDR unsigned int

Address.

OFFSET unsigned int

Indicates which halfword of a 32-bit Thumb instruction the vector catch exception is set on. 0 for the first halfword or if the instruction is 16 bits wide, or 1 for the second halfword.

ArchMsg.Warning.unpredictable_watchpoint_far

TAGS **UNPREDICTABLE** DISPLAY The value for FAR/EDWAR for hitting this watchpoint may be anywhere in the range %{LOWER_BOUND} to %{UPPER_BOUND}. Fields:

LOWER_BOUND unsigned int

the lowest address accessed by the instruction that triggered the watchpoint.

UPPER_BOUND unsigned int

the highest watchpointed address accessed by that instruction.

ArchMsg.Warning.unpredictable_watchpoint_far_isscmp

DISPLAY The value for FAR/EDWAR for hitting this watchpoint may be anywhere in the range %{LOWER_BOUND} to %{UPPER_BOUND}. Fields:

LOWER_BOUND unsigned int

the lowest address accessed by the instruction that triggered the watchpoint.

UPPER_BOUND unsigned int

the highest watchpointed address accessed by that instruction.

ArchMsg.Warning.warning_AdvSIMDEExpandImmUnexpectedZero

TAGS **UNPREDICTABLE** DISPLAY AdvSIMDEExpandImm may treat this immediate value as **UNPREDICTABLE**.

ArchMsg.Warning.warning_ConditionalSMC

TAGS **UNPREDICTABLE** DISPLAY SMC instruction has **UNPREDICTABLE** effects when conditional and when combined with traps.

ArchMsg.Warning.warning_TLBID_out_of_range

TLBID out of range in TLBI according to the configuration of NIS, NVIS, NOS and NVOS. Fields:

NIS unsigned int

NIS.

NOS unsigned int

NOS.

NVIS unsigned int

NVIS.

NVOS unsigned int

NVOS.

TLBID unsigned int

The TLBID value used in the TLBI instruction.

ArchMsg.Warning.warning_TTL_or_TG_mismatches_current_translation_granule

TTL or TG mismatches current translation granule in TLBI VA/RVA/IPA/RIPA. Fields:

CURRENT_TG unsigned int

Current TG as indicated by TCR_ELx or VTCR_EL2.

TLBI_TG unsigned int

TG (or TTL[3:2]) used in TLBI instruction.

ArchMsg.Warning.warning_access_crosses_page_boundary

TAGS **UNPREDICTABLE** DISPLAY Access crosses page boundary.

ArchMsg.Warning.warning_access_crosses_page_boundary_has_fault

TAGS **UNPREDICTABLE** DISPLAY Access that crosses page boundary has fault(s).

ArchMsg.Warning.warning_access_crosses_page_boundary_has_fault_on_both_pages

TAGS **UNPREDICTABLE** DISPLAY Access that crosses page boundary has a fault on each page.
Fields:

ADDR1 unsigned int

FAR address on first page.

ADDR2 unsigned int

FAR address on second page.

FSTATUS1 unsigned int

Fault status on first page.

FSTATUS2 unsigned int

Fault status on second page.

ArchMsg.Warning.warning_access_crosses_page_boundary_spanning_different_memory

TAGS **UNPREDICTABLE** DISPLAY Access at address %{ADDR} crosses page from MEM_TYPE %{MEMTYPE_PAGE1} to %{MEMTYPE_PAGE2}. Fields:

ADDR unsigned int

address of access crossing page.

MEMTYPE_PAGE1 signed int

First Page's memory type (0-Normal, 1-Non-Normal).

MEMTYPE_PAGE2 signed int

Second Page's memory type (0-Normal, 1-Non-Normal).

ArchMsg.Warning.warning_access_wraps_around_memory

TAGS **UNPREDICTABLE** DISPLAY Access wraps around memory in %{WIDTH:d}bit mode.

Fields:

WIDTH unsigned int

Address width.

ArchMsg.Warning.warning_bcr_linking_status

TAGS **UNPREDICTABLE** DISPLAY DBGBCR.{BT2,BT} combination gives unpredictable behaviour.

ArchMsg.Warning.warning_bcr_mask_reserved

TAGS **UNPREDICTABLE** DISPLAY DBGBCR_ELn.MASK is set to a reserved value.

ArchMsg.Warning.warning_bvr_masked_bit_not_zero

TAGS **UNPREDICTABLE** DISPLAY DBGBVRn_EL1.VA contains a bit masked by DBGBCRn_EL1.MASK that is not zero.

ArchMsg.Warning.warning_ccsidr_unimplemented_level

TAGS **UNPREDICTABLE** DISPLAY CCSIDR read while CSSELR %{CSSELR:x} points to an unimplemented cache level. Fields:

CSSELR unsigned int

current effective value of CSSELR.

ArchMsg.Warning.warning_change_to_ns_when_tge_set

TAGS **UNPREDICTABLE** DISPLAY Attempting to change NS to 1 when HCR.TGE = 1.

ArchMsg.Warning.warning_contiguous_bit_check_abort

TAGS **UNPREDICTABLE** DISPLAY An abort occurred whilst attempting to check TLB entry is contiguous at %{ENTRY_ADDR}. Fields:

ENTRY_ADDR unsigned int

address of conflicting TLB entry.

ArchMsg.Warning.warning_contiguous_bit_error

TAGS **UNPREDICTABLE** DISPLAY TLB entry at %{ENTRY_ADDR} was not contiguous with entry at %{CONTIG_ADDR} Contents are %{ENTRY_DATA} but expected %{CONTIG_DATA}. Fields:

CONTIG_ADDR unsigned int

address of the TLB first read, with which this entry is expected to be contiguous.

CONTIG_DATA unsigned int

expected contents based on the entry at CONTIG_ADDR.

ENTRY_ADDR unsigned int

address of conflicting TLB entry.

ENTRY_DATA unsigned int

contents of conflicting TLB entry.

ArchMsg.Warning.warning_contiguous_bit_error_D128

TAGS **UNPREDICTABLE** DISPLAY TLB entry at %{ENTRY_ADDR} was not contiguous with entry at %{CONTIG_ADDR} Contents are %{ENTRY_DATA_HI} %{ENTRY_DATA_LO} but expected %{CONTIG_DATA}. Fields:

CONTIG_ADDR unsigned int

address of the TLB first read, with which this entry is expected to be contiguous.

CONTIG_DATA unsigned int

expected contents based on the entry at CONTIG_ADDR.

ENTRY_ADDR unsigned int

address of conflicting TLB entry.

ENTRY_DATA_HI unsigned int

Higher bit contents of conflicting TLB entry.

ENTRY_DATA_LO unsigned int

Lower bit contents of conflicting TLB entry.

ArchMsg.Warning.warning_cp10_cp11_mismatch

TAGS **UNPREDICTABLE** DISPLAY Unpredictable configuration of CP10 and CP11 controls at EL %{EL:d}. Fields:

EL unsigned int

EL owning the control.

ArchMsg.Warning.warning_cp10_cp11_reserved_value

TAGS **UNPREDICTABLE** DISPLAY Unpredictable configuration of CPACR CP10 (%{CP10}) or CP11 (%{CP11}). Fields:

CP10 unsigned int

CP10 access.

CP11 unsigned int

CP11 access.

ArchMsg.Warning.warning_csselr_level_out_of_range

TAGS **UNPREDICTABLE** DISPLAY CSSELR.Level out of range, written value %{LEVEL} with only %{IMPLEMENTED:d} levels implemented. Fields:

IMPLEMENTED unsigned int

number of cache levels implemented.

LEVEL unsigned int

written value of CSSELR.Level.

TYPE enum

The type of cache selected by CSSELR.InD and TnD.

ArchMsg.Warning.warning_debug_flow_control_bits_not_obeyed

TAGS **UNPREDICTABLE** DISPLAY Flow control bits for %{REGISTER} were not obeyed, causing %{ERROR}. Fields:

ERROR enum

type of error resulting.

REGISTER enum

register under consideration.

ArchMsg.Warning.warning_debug_register_access_during_reset

TAGS **UNPREDICTABLE** DISPLAY Unpredictable %{IS_WRITE:(read|write)} access to debug register offset %{OFFSET} during reset. Fields:

IS_WRITE unsigned int

Write Not Read.

OFFSET unsigned int

Register Offset.

ArchMsg.Warning.warning_decode_cps_inconsistent_fields

TAGS **UNPREDICTABLE** DISPLAY The mode setting fields in the CPS instruction are inconsistent.

ArchMsg.Warning.warning_decode_invalid_state

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable in current exception-level/security state.

ArchMsg.Warning.warning_default_cacheable_mmu_on

TAGS **UNPREDICTABLE** DISPLAY Default cacheable is enabled (HCR.DC = 1) while MMU is enabled.

ArchMsg.Warning.warning_default_cacheable_vmmu_off

TAGS **UNPREDICTABLE** DISPLAY Default cacheable is enabled (HCR.DC = 1) while HCR.VM = 0.

ArchMsg.Warning.warning_deprecated_wvr_bit_2_set

TAGS **UNPREDICTABLE** DISPLAY DBGWVR%{N:d}_EL1=%{WVR:h} has bit 2 set, deprecated in ARMv8. Fields:

N unsigned int

index.

WVR unsigned int

DBGWVR.

ArchMsg.Warning.warning_eret_while_software_step_active_pending

Missing ISB between setting MDSCR_EL1.SS and ERET TAGS **UNPREDICTABLE**.

ArchMsg.Warning.warning_exclusive_to_non_normal

TAGS **UNPREDICTABLE** DISPLAY Exclusive Access to Non-Normal Memory %{ADDR}. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_exclusive_to_non_writeback

TAGS **UNPREDICTABLE** DISPLAY Exclusive Access to Non-Writeback-Cacheable Memory
%{ADDR}. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_execute_from_device_memory

TAGS **UNPREDICTABLE** DISPLAY An attempt was made to execute from Device memory.

ArchMsg.Warning.warning_illegal_cpsr_mode

TAGS **UNPREDICTABLE** DISPLAY Writing an illegal or unimplement mode (%{NEW_MODE:h})
to CPSR was unpredictable. Fields:

NEW_MODE unsigned int

New value of CPSR.M.

ArchMsg.Warning.warning_illegal_srs_mode

TAGS **UNPREDICTABLE** DISPLAY Illegal or **UNPREDICTABLE** mode (%{MODE:h}) used for SRS
instruction. Fields:

MODE unsigned int

mode for Banked SP.

ArchMsg.Warning.warning_implementation_defined_read_debug_register_in_SCS

Read Access to Debug registers from the processor is IMPLEMENTATION DEFINED TAGS
IMP_DEF. Fields:

PPB_OFFSET unsigned int

debug register offset.

ArchMsg.Warning.warning_implementation_defined_sequential_security_transitions_supported

The behavior of sequential instruction fetches that cross from non-secure to secure memory
and contain an SG instruction is CONSTRAINED **UNPREDICTABLE** TAGS **UNPREDICTABLE**.

ArchMsg.Warning.warning_implementation_defined_stack_limit_check_supported

It is **IMPLEMENTATION DEFINED** whether stack pointer limit checking is performed for this
instruction TAGS IMP_DEF. Fields:

ADDRESS unsigned int

Address of instruction.

OPCODE unsigned int

instruction opcode.

ArchMsg.Warning.warning_implementation_defined_taggednonsharedmemorytype

TAGS IMP_DEF DISPLAY Memory attribute has Tagged, non shared memory type; it is
IMPLEMENTATION DEFINED whether the memory location is treated as Tagged or non-Tagged.

ArchMsg.Warning.warning_implementation_defined_write_debug_register_in_SCS

Write Access to Debug registers from the processor is IMPLEMENTATION DEFINED TAGS IMP_DEF. Fields:

DATA unsigned int

data attempted to be written.

PPB_OFFSET unsigned int

debug register offset.

ArchMsg.Warning.warning_invalid_tcr_granule

TAGS UNPREDICTABLE DISPLAY TCR.TG{%TG_ID:(0|1)} bits have been set to a granule size {%REQUEST:d}K not implemented - using {%SUBSTITUTE:d}K. Fields:

REQUEST signed int

page size requested (or 0 for reserved).

SUBSTITUTE signed int

best guess available page size.

TG_ID bool

bits TG0 or TG1.

ArchMsg.Warning.warning_load_multiple_user_registers_from_user_mode

TAGS UNPREDICTABLE DISPLAY An LDM(user registers) instruction executed from user mode.

ArchMsg.Warning.warning_mte_in_mem_attr_not_supported

TAGS UNPREDICTABLE DISPLAY Memory attribute {%ATTR} has Tagged memory type but Armv8.5-MTE is not implemented. Fields:

ATTR unsigned int

Memory attribute.

ArchMsg.Warning.warning_reserved_breakpoint_state_match

TAGS UNPREDICTABLE DISPLAY {%IS_BREAKPOINT:(Watchpoint|Breakpoint)} programmed with a reserved combination of HMC, SSC and {%IS_BREAKPOINT:(PAC|PMC)}. Fields:

IS_BREAKPOINT unsigned int

Is a breakpoint, not a watchpoint.

ArchMsg.Warning.warning_shareability

TAGS UNPREDICTABLE DISPLAY Unpredictable: combination of the 1st and 2nd stages of translation is Normal Inner Non-Cacheable, Outer Non-Cacheable.

ArchMsg.Warning.warning_software_step_set_while_enabled

TAGS UNPREDICTABLE DISPLAY MDSCR_EL1.SS set to 1 while software step debug exceptions are enabled.

ArchMsg.Warning.warning_thumb_instruction_wraps_around_memory

TAGS UNPREDICTABLE DISPLAY Thumb instruction wraps around memory.

ArchMsg.Warning.warning_ttbr_sbz_bits_are_not_zero

TAGS UNPREDICTABLE DISPLAY %{TTBR}[%{MSB:d}:0] should be zero, but are not. Fields:

MSB unsigned int

x-1.

TTBR enum

which TTBR register.

ArchMsg.Warning.warning_unaligned_address_dbgdttrrx_write

TAGS UNPREDICTABLE DISPLAY Unaligned address %{ADDR} in XO/RO when DBGDTTRRX written in memory access mode. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unaligned_address_dbgdttrtx_read

TAGS UNPREDICTABLE DISPLAY Unaligned address %{ADDR} in XO/RO when DBGDTTRTX read in memory access mode. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unaligned_to_device

TAGS UNPREDICTABLE DISPLAY Unaligned Access to Device Memory %{ADDR}. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unaligned_to_strongly_ordered

TAGS UNPREDICTABLE DISPLAY Unaligned Access to Strongly Ordered Memory %{ADDR}. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unknown_sau_rnr

TAGS UNPREDICTABLE DISPLAY SAU_RNR was set to an unsupported value.

ArchMsg.Warning.warning_unpredictable_AIRCR_PRIS_and_BFHFNMINs

TAGS UNPREDICTABLE DISPLAY The effect of setting both AIRCR.BFHFNMINs and AIRCR.PRIS to 1 is UNPREDICTABLE.

ArchMsg.Warning.warning_unpredictable_AIRCR_VECTCLRACTIVE_when_not_in_debug

TAGS UNPREDICTABLE DISPLAY The effect of writing a 1 to AIRCR.VECTCLRACTIVE if the processor is not halted in Debug state is UNPREDICTABLE.

ArchMsg.Warning.warning_unpredictable_AIRCR_VECTRESET_and_SYSRESETREQ

TAGS UNPREDICTABLE DISPLAY When the processor is halted in Debug state, if a write to the register writes a 1 to both VECTRESET and SYSRESETREQ, the behavior is UNPREDICTABLE.

ArchMsg.Warning.warning_unpredictable_AIRCR_VECTRESET_when_not_in_debug

TAGS **UNPREDICTABLE** DISPLAY The effect of writing a 1 to AIRCR.VECTRESET if the processor is not halted in Debug state is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_AIRCR_incorrect_VKEY

TAGS **UNPREDICTABLE** DISPLAY The value 0x05FA must be written to AIRCR.VKEY, otherwise the register write is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_EXC_RETURN_Reserved_Bit

EXC_RETURN[23:7] are reserved with the special condition that all bits should be written as one. Values other than all 1s are **UNPREDICTABLE**. TAGS **UNPREDICTABLE**. Fields:

EXC_RETURN unsigned int

Special PC value.

SBO_MASK unsigned int

Which bits Should Be One to not UNPRED.

ArchMsg.Warning.warning_unpredictable_PMEVTYPEPn_ELO_VS_reserved_value

TAGS **UNPREDICTABLE** DISPLAY Unpredictable reserved value of PMEVTYPEn<n>_ELO.VS.

ArchMsg.Warning.warning_unpredictable_change_to_DEMCR_MON_STEP_at_insufficient_priority

TAGS **UNPREDICTABLE** DISPLAY The effect of changing DEMCR.MON_STEP at an execution priority that is lower than the priority of the DebugMonitor exception is **CONSTRAINED UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_change_to_priority_of_active_exception

TAGS **UNPREDICTABLE** DISPLAY Changing the priority of an active exception is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_clear_lspact

TAGS **UNPREDICTABLE** DISPLAY **UNPREDICTABLE** state on context switch. Fields:

DRVEXC_TAKEN unsigned int

Derived exception taken, not pended.

REACHED_LAST_STORE unsigned int

Only the last faulted.

ArchMsg.Warning.warning_unpredictable_dbg_exit_unaligned_dlr

TAGS **UNPREDICTABLE** DISPLAY Debug exit to AArch32 with DLR[0] set to 1. PC[0] can be set to 0 or to DLR[0].

ArchMsg.Warning.warning_unpredictable_exception_catch

TAGS **UNPREDICTABLE** DISPLAY Generation of an exception catch event is unpredictable.

ArchMsg.Warning.warning_unpredictable_exception_return_inconsistent_state

TAGS **UNPREDICTABLE** DISPLAY state on exception return is unpredictable.

ArchMsg.Warning.warning_unpredictable_exception_return_instruction

TAGS **UNPREDICTABLE** DISPLAY exception return instruction is unpredictable.

ArchMsg.Warning.warning_unpredictable_in_debug_state

TAGS **UNPREDICTABLE** DISPLAY Instruction is **UNPREDICTABLE** when executed in debug state.

ArchMsg.Warning.warning_unpredictable_mair_encoding

TAGS **UNPREDICTABLE** DISPLAY Memory attribute %{ATTR} has invalid MAIR encoding; this may lead to unpredictable results. Fields:

ATTR unsigned int

Memory attribute.

ArchMsg.Warning.warning_unpredictable_pmu_counter_access

TAGS **UNPREDICTABLE** DISPLAY Access to PMU counters from non-secure EL0 or EL1 is **UNPREDICTABLE** with MDCR_EL2.HPMN set to 0.

ArchMsg.Warning.warning_unpredictable_prioritization_breakpoint_match_vector_catch

TAGS **UNPREDICTABLE** DISPLAY Two events with the same priority occurred at the same instruction: Address Matching Vector Catch debug event, and Breakpoint debug event. It is constrained unpredictable which is taken.

ArchMsg.Warning.warning_unpredictable_stack_selection

TAGS **UNPREDICTABLE** DISPLAY **UNPREDICTABLE** state on context switch.

ArchMsg.Warning.warning_unpredictable_tsize_out_of_range

TAGS **UNPREDICTABLE** DISPLAY %{TCR}.T%{TTBR:d}SZ (%{TSIZE:d} bits) is out of range. Must be between 25 and %{TSIZE_MAX:d}. Fields:

TCR enum

Which TCR.

TSIZE signed int

Value of TSize.

TSIZE_MAX signed int

The maximum allowed value.

TTBR unsigned int

Which TTBR region.

ArchMsg.Warning.warning_unpredictable_unaligned_pc_as_base_register

TAGS **UNPREDICTABLE** DISPLAY The PC value must be word-aligned, otherwise the behavior of the instruction is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_unaligned_pop_stack

TAGS **UNPREDICTABLE** DISPLAY **CONSTRAINED UNPREDICTABLE** unaligned stack frame %{FRAME_ADDR:x} in exception-return. Fields:

FRAME_ADDR unsigned int

unaligned stack frame address.

ArchMsg.Warning.warning_unpredictable_unaligned_to_non_writeback

TAGS **UNPREDICTABLE** DISPLAY Unaligned Access to Non-Writeback-Cacheable Memory.

ArchMsg.Warning.warning_unpredictable_vmsa_memattrib

TAGS **UNPREDICTABLE** DISPLAY Unpredictable vmsa memory attribute with texcb value programmed to %{TEXCB} when tex remap is %{TEX_REMAP}. Fields:

TEXCB unsigned int

texcb value.

TEX_REMAP bool

use tex remap.

ArchMsg.Warning.warning_unpredictable_vtcr_t0sz_sl0

TAGS **UNPREDICTABLE** DISPLAY The combination of VTCR.TOSZ=%{TOSZ:d} and VTCR.SL0=%{SL0:d} is unpredictable. Fields:

SL0 signed int

VTCR.SL0, starting level for stage 2 translation table walks.

TOSZ signed int

VTCR.TOSZ, required input address range.

ArchMsg.Warning.warning_unpredictable_write_DHCSR

TAGS **UNPREDICTABLE** DISPLAY **UNPREDICTABLE** set of bit-changes in DHCSR. Fields:

NEW unsigned int

data attempted to be written.

OLD unsigned int

previous value.

ArchMsg.Warning.warning_unpredictable_write_SHCSR

TAGS **UNPREDICTABLE** DISPLAY **UNPREDICTABLE** setting a pending bit to 1 for an exception with priority >= execution priority.

ArchMsg.Warning.warning_unsupported_access_to_memory_mapped_register

TAGS **UNPREDICTABLE** DISPLAY Access to memory mapped register is unsupported.

ArchMsg.Warning.warning_user_jmcr_access

TAGS **UNPREDICTABLE** DISPLAY User %{WRITE:(write|read)} access to JMCR is unpredictable. Fields:

WRITE unsigned int

access is write.

ArchMsg.Warning.warning_wcr_mask_and_bas

TAGS **UNPREDICTABLE** DISPLAY DBGWCR_EL1.MASK is non-zero and BAS is not 0xff.

ArchMsg.Warning.warning_wcr_mask_reserved

TAGS **UNPREDICTABLE** DISPLAY DBGWCR_ELn.MASK is set to a reserved value.

ArchMsg.Warning.warning_wcr_non_configuous_bas

TAGS **UNPREDICTABLE** DISPLAY DBGWCR_EL1.BAS has non-contiguous set of ones.

ArchMsg.Warning.warning_wvr_masked_bit_not_zero

TAGS **UNPREDICTABLE** DISPLAY DBGWVRn_EL1.VA contains a bit masked by DBGWCRn_EL1.MASK that is not zero.

ArchMsg.Why.why_branch_target_exception

A branch target exception has occurred. DISPLAY branch target exception: target pc={PC}, target opcode={OPCODE}. PRIMARY KEY PC. Fields:

BTTYPE unsigned int

pstate.BTYPE at time branch taken.

OPCODE unsigned int

opcode at branch target.

PC unsigned int

Virtual Address of branch target.

ArchMsg.Why.why_illegal_state

An illegal mode change has occurred. DISPLAY illegal mode change: {MESSAGE}. Fields:

MESSAGE string

The message.

BRANCH_MISPREDICT

Simulating branch mispredict. Fields:

PC unsigned int

Origin address (or 0 if unavailable).

BRA_DIR

Direct branches, to immediate address. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISET enum

The instruction set of the branch instruction.

IS_COND bool

Indicates if this is a conditional branch.

IS_HINTED bool

Whether the conditional instruction is hinted.

IS_ISB bool

Whether the instruction is ISB.

IS_LINK bool

Indicates if this is a branch with link.

PC unsigned int

The address of the branch instruction.

TARGET_ISET enum

The instruction set after the branch.

TARGET_PC unsigned int

The address the instruction branches to.

BRA_INDIR

Indirect branches, perhaps to a register. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISET enum

The instruction set of the branch instruction.

IS_COND bool

Indicates if this is a conditional branch.

IS_HINTED bool

Whether the conditional instruction is hinted.

IS_ISB bool

Whether the instruction is ISB.

IS_LINK bool

Indicates if this is a branch with link.

PC unsigned int

The address of the branch instruction.

TARGET_ISET enum

The instruction set after the branch.

TARGET_PC unsigned int

The address the instruction branches to.

BRA_INDIR_CCFAIL

Indirect branches fail, perhaps to a register.

BRA_RET_CCFAIL

Branch return fail, perhaps to a register.

CACHE_MAINTENANCE_OP

Cache Maintenance Operation. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

DATA unsigned int

Specified MVA or Set/way.

FUNCTION enum

Clean / Invalidate.

SCOPE enum

Affected region.

SIDE enum

Inst / Data.

CCFAIL

Conditional instruction condition check fail. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CCFAIL_UNC

Unconditional instruction condition check fail. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CCPASS

Conditional instruction condition check pass. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CCPASS_UNC

Unconditional instruction condition check pass. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CODE_CACHE_MAINT

Code cache maintenance. Fields:

END_INCL unsigned int

Inclusive end address for VA or PA requests.

NS bool

Non-secure world for PA.

START unsigned int

Start address for VA or PA requests.

TYPE enum

Request type.

COMPILE_BLOCK_END

Last instruction of basic block translated. Fields:

VADDR unsigned int

Address of next instruction after this basic block.

COMPILE_BLOCK_START

First instruction of basic block translated. Fields:

VADDR unsigned int

Address of first instruction in basic block.

COMPILE_INST

ARM instruction compiled. Fields:

DISASS string

Disassembly of the instruction.

ISSET enum

The instruction set of this instruction.

ITSTATE unsigned int

The ITSTATE current for the instruction.

OPCODE unsigned int

The opcode of the instruction.

PC unsigned int

The address of the instruction.

SIZE unsigned int

The size of the instruction in bytes.

CONTEXTIDR

Context ID Register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

NS bool

Secure or nonsecure banked register is accessed.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The new value written.

CONTEXT_SYNC

Called for every context synchronization event. Fields:

ARCH_CSE bool

Architectural Context Synchronization Event.

PC unsigned int

The address of the synchronization instruction.

SYSREG_SYNC bool

System register synchronization event.

CORE_ENDIAN

Core BE8 Big-Endian state changed. Fields:

BE8 bool

Core BE8 Big-Endian state.

CORE_INFO

Static processor attributes. Only triggered by a call to DumpState(). Fields:

ARCH_PROFILE enum

The architecture profile of the core.

CLUSTER_ID unsigned int

The cluster ID of this processor.

CORE_NUM unsigned int

The number of this core in an MP processor.

DETAILED_CORE_FEATURES enum

Detailed core features as a bitfield. Enum values are bitmasks to extract the info from the field.

FPU_VERSION enum

The VFP version implemented by the core.

MEM_ARCH enum

The memory architecture of the core.

MIDR_EL1 unsigned int

MIDR_EL1 value.

NUM_CORES unsigned int

The number of cores in this MP processor.

QUANTUM_SIZE unsigned int

The default quantum size of the core (in ticks).

SECURITY_FEATURES bool

Does the core have security features?.

CORE_LOADS

Processor load accesses. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ACQREL enum

Is this an acquire/release.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DATA unsigned int

The data read or written.

ELEMENT_SIZE unsigned int

Width of each element.

ENDIANNESS enum

Byte ordering of this access.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

LOCK enum

Normal, exclusive or locked access.

MECID unsigned int

Memory Encryption Context ID, if applicable.

MECID2 unsigned int

Memory Encryption Context ID of the second page, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PAGE_SIZE unsigned int

Page size as $\log_2(\text{size in bytes})$.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK, 2=Exclusive Failed, 3=Unsupported.

SIZE unsigned int

Width of the access in bytes. Only required if DATA is not traced.

TRANS bool

Is this a translated access.

VADDR unsigned int

The virtual address of the access.

CORE_REGS

Changes of the core registers R0 to R14. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number, 0 to 14.

MODE enum

Bank of the register accessed.

OLD_VALUE unsigned int

The old value overwritten.

PHYS_ID enum

The physical register accessed.

VALUE unsigned int

The new value written to the register.

CORE_REGS64

Changes of the core registers X0..X30, SP_ELn. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ID enum

The register number, 0-30 for X0-X30, >=32 for SP_ELn.

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

CORE_REGS64_READ

General purpose AArch64 register has been read X0..X30, SP_ELn. Fields:

ID enum

The register number, 0-30 for X0-X30, >=32 for SP_ELn.

VALUE unsigned int

The value read from the register.

CORE_STALLED_TRANSITION

Detect when a core is prevented from progressing PChannel states. Fields:

CURRENT_STATE enum

Current power mode of core.

REASON string

The reason a transition was stalled, and what must be done to unblock it.

TARGET_STATE enum

Desired power mode of core after transition.

CORE_STORES

Processor store accesses. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ACQREL enum

Is this an acquire/release.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DATA unsigned int

The data read or written.

ELEMENT_SIZE unsigned int

Width of each element.

ENDIANNESS enum

Byte ordering of this access.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

LOCK enum

Normal, exclusive or locked access.

MECID unsigned int

Memory Encryption Context ID, if applicable.

MECID2 unsigned int

Memory Encryption Context ID of the second page, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PAGE_SIZE unsigned int

Page size as $\log_2(\text{size in bytes})$.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK, 2=Exclusive Failed, 3=Unsupported.

SIZE unsigned int

Width of the access in bytes. Only required if DATA is not traced.

TRANS bool

Is this a translated access.

VADDR unsigned int

The virtual address of the access.

CP14_READ

System Coprocessor register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP14 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc1 unsigned int

opcode 1.

opc2 unsigned int
opcode 2.

CP14_WRITE

System Coprocessor register write. Fields:

CORE_NUM unsigned int
Core number in a multi processor.

CRm unsigned int
CRm.

CRn unsigned int
CRn.

NS bool
Secure or nonsecure banked register is accessed.

REG_NAME string
Name of the CP14 register accessed.

SECURITY_STATE enum
The current security state.

UNDEF bool
The register accessed is undefined.

UPDATED_VALUE unsigned int
Updated value of the register now it has been written.

VALUE unsigned int
The new value written.

opc1 unsigned int
opcode 1.

opc2 unsigned int
opcode 2.

CP15_READ

System Coprocessor register read. Fields:

CORE_NUM unsigned int
Core number in a multi processor.

CRm unsigned int
CRm.

CRn unsigned int
CRn.

NS bool
Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP15 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc1 unsigned int

opcode 1.

opc2 unsigned int

opcode 2.

CP15_READ64

System Coprocessor register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP15 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc unsigned int

opcode 1.

CP15_WRITE

System Control Coprocessor register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP15 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc1 unsigned int

opcode 1.

opc2 unsigned int

opcode 2.

CP15_WRITE64

System Coprocessor register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP15 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc unsigned int

opcode 1.

CPSR

CPSR change. (In AArch64 PSTATE represented in SPSR format). Fields:

CORE_NUM unsigned int

Core number in a multi processor.

OLD_VALUE unsigned int

The old CPSR value.

UNKNOWN unsigned int

Bits within the register that have unknown value.

VALUE unsigned int

The new CPSR value.

CRYPTO_SPEC

Every crypto instruction speculatively executed.

CorePowerStateChange

Core power-up state has changed. DISPLAY Core powered %{PU:(up|down)}. Fields:

PU bool

Whether the core powered up.

DATA_CACHE_SET_ALLOC_TAGS

Set allocation tags from DC GVA instruction. Fields:

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

NSDESC unsigned int

The security state of the access.

PADDR unsigned int

The physical address of the access.

PAS enum

The physical address space of the page.

RESPONSE enum

0=Aborted, 1=OK.

SIZE unsigned int

The size of memory in bytes.

TAG unsigned int

The value of the tag being set.

TAGGING_TYPE enum

Type of Memory tagging (Physical or Virtual).

TAG_VADDR unsigned int

The virtual address of the Virtual Tag if Virtual tagging is used; otherwise 0.

VADDR unsigned int

Virtual address used for setting allocation tags.

DATA_CACHE_ZERO

Zero and invalidate cache line from DC ZVA instruction. Fields:

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

NSDESC unsigned int

The security state of the access.

PADDR unsigned int

The physical address of the access.

PADDR_TAG unsigned int

If tag checked, the physical address tag for the access.

PAS enum

The physical address space of the page.

RESPONSE enum

0=Aborted, 1=OK.

SIZE unsigned int

The size of zero'd memory in bytes.

TAG_CHECKED bool

Is this access tag checked.

VADDR unsigned int

Virtual address of the zero'd cache line.

DEBUG_EVENT

Hardware debug support event. Fields:

EVENT enum

Description of event.

VALUE unsigned int

data value.

DEBUG_TARGET_ISET

Hardware debug target ISA. Fields:

TARGET_ISET enum

The target instruction set of the debug state.

DMI_ALLOCATE

DMI allocated. Fields:

ACCESS_DIRECTION enum

READ / WRITE.

PADDR unsigned int

The physical address of the DMI.

PTR unsigned int

DMI range pointer.

SIDE enum

DSIDE / ISIDE.

SIZE unsigned int

The size of DMI.

VADDR unsigned int

The virtual address of the DMI..

DMI_HIT

DMI hit, access used DMI (only fire for small DMI regions i.e. less than a full page). Fields:

ACCESS_DIRECTION enum

READ / WRITE.

PADDR unsigned int

The physical address of the DMI.

SIDE enum

DSIDE / ISIDE.

SIZE unsigned int

The size of the DMI.

VADDR unsigned int

The virtual address of the DMI..

DMI_REVOKE

DMI revoked. Fields:

ACCESS_DIRECTION enum

READ / WRITE.

PADDR unsigned int

The physical address of the DMI.

SIDE enum

DSIDE / ISIDE.

SIZE unsigned int

The size of DMI.

VADDR unsigned int

The virtual address of the DMI..

END_COMPILE

Compilation end. Fields:

END_OF_PAGE_COUNT unsigned int

Number of basic blocks exited due to page boundary since START_COMPILE.

FETCHFAIL_COUNT unsigned int

Number of basic blocks exited due to ifetch failure START_COMPILE.

INST_COUNT unsigned int

Number of instructions compiled since START_COMPILE.

NONSEQ_COUNT unsigned int

Number of basic blocks exited due to non-sequential instructions since START_COMPILE.

PAGE_STRADDLE_COUNT unsigned int

Number of basic blocks exited due to unaligned instructions crossing page since START_COMPILE.

ESB

Error Synchronization Barrier hit. Fields:

ASYNC bool

ESB is async.

IMPLICIT bool

ESB is implicit.

EXCEPTION

Exception taken. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ESR unsigned int

The value assigned to ESR when exception is taken.

ESR_EC unsigned int

The Value assigned to ESR.EC when exception is taken.

IS_PHYSICAL bool

Physical or Virtual exception.

IS_TRAP bool

Exception coming from a trap.

LR unsigned int

The value assigned to the link register.

PC unsigned int

The location where the exception occurred.

PREFERRED_RETURN unsigned int

The preferred return address for the exception.

TAKEN_LOCALLY bool

Whether exception is taken locally(same exception level).

TARGET_ISET enum

The instruction set of the exception handler code.

TARGET_PC unsigned int

The address the exception branches to.

VECTOR enum

The exception vector.

VECTOR_OFFSET unsigned int

The offset of an exception vector from the base address.

EXCEPTION_END

Every exception completed.

EXCEPTION_INFO

Trace human-readable reason of the current exception for data/instruction aborts, floating point exceptions and unknown exceptions. Fields:

DESCRIPTION string

Reason for current exception.

ESR_EC unsigned int

The value of ESR.EC when exception is taken.

EXCEPTION_RAISE

Every exception raised.

EXCEPTION_RETURN

Branches on leaving exception. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISET enum

The instruction set of the branch instruction.

IS_COND bool

Indicates if this is a conditional branch.

IS_HINTED bool

Whether the conditional instruction is hinted.

IS_ISB bool

Whether the instruction is ISB.

IS_LINK bool

Indicates if this is a branch with link.

PC unsigned int

The address of the branch instruction.

TARGET_ISET enum

The instruction set after the branch.

TARGET_PC unsigned int

The address the instruction branches to.

EXCEPTION_RETURN_PREBRANCH

Exception return event that must occur before PC and processor state have updated.

EXCEPTION_START

Every exception started.

FIQ_TAKEN

FIQ taken exception.

FP_EXCEPTIONS_SIGNALLED

Trace which FP exceptions were signalled as a part of the FP operations performed by an instruction. Fields:

FP_EXCEPTIONS_SIGNALLED_MASK unsigned int

Indicates which FP exceptions were signalled by an instruction. This uses the same encoding as the cumulative floating-point exception bitfields in FPSR.

FP_STATE

Floating point state. Fields:

FPSR unsigned int

FPSR register value.

MASK unsigned int

The mask indicating updated cumulative bits.

FREQ_CHANGED

Core frequency changed. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The instruction count of this core.

NEW_FREQ signed int

The new frequency of this core (expressed in Hz).

OLD_FREQ signed int

The old frequency of this core (expressed in Hz).

HLT

HLT instruction occurred. Fields:

IMM16 unsigned int

The 16-bit immediate value encoded in HLT instruction.

INST

Every instruction executed. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DEBUG_STATE bool

The instruction is executed in debug state.

DISASS string

Disassembly of instruction.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISSET enum

The current instruction set.

ITSTATE unsigned int

The current ITSTATE.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

MODE enum

The mode the core is in.

NS unsigned int

The core's non-secure bit.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

OPCODE unsigned int

The opcode of the instruction.

PADDR unsigned int

The physical address of the instruction.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the instruction.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the page.

PC unsigned int

The address of the instruction.

SECURITY_STATE enum

The core's security state.

SIZE unsigned int

The size of the instruction in bytes.

INST_END

Every instruction completed.

INST_START

Every instruction started. Fields:

DEBUG_STATE bool

The instruction is executed in debug state.

ISSET enum

The current instruction set.

MODE enum

The mode the core is in.

NS enum

The current Secure State.

PC unsigned int

The address of the conditional instruction.

SECURITY_STATE enum

The current Security State.

INST_STRADDLE

Instruction straddles boundary. Fields:

DEBUG_STATE bool

The instruction is executed in debug state.

ISSET enum

The current instruction set.

MODE enum

The mode the core is in.

NS enum

The current Secure State.

PC unsigned int

The address of the conditional instruction.

SECURITY_STATE enum

The current Security State.

IRQ_TAKEN

IRQ taken exception.

ISB

ISB instruction occurred. Fields:

PC unsigned int

The address of the ISB instruction.

LOAD_MULTIPLE_REGS_LIST

List of destination registers (not source) of a multiple registers load instruction. Fields:

REG string

Register to add into the list.

LOCAL_MONITOR

Local monitor activity. Fields:

PADDR unsigned int

Local Monitor Address.

State enum

State of the monitor (Open/Exclusive).

MAPPING_CREATE

A memory mapping has been created. This is an implementation detail of the model and so may change without notice. Fields:

ASID unsigned int

The ASID used during translation, or 0 if no ASID was used.

IS_VALID bool

Whether the mapping is valid or not.

PADDR unsigned int

Base physical address of region, or 0 if not known.

SIDE enum

I-side or D-side.

SIZE unsigned int

Size of the page, or 0 if not known.

VADDR unsigned int

Base virtual address of region.

VMID unsigned int

The VMID used during translation, or 0 if no VMID was used.

MAPPING_DESTROY

A memory mapping has been destroyed. This is an implementation detail of the model and so may change without notice. Fields:

ASID unsigned int

The ASID used during translation.

CAUSE enum

The cause of invalidation.

PADDR unsigned int

Base physical address of page.

SIDE enum

I-side or D-side.

SIZE unsigned int

Size of the page.

VADDR unsigned int

Base virtual address of page.

VMID unsigned int

The VMID used during translation.

MEMMAP_DEBUG_READ

Memory mapped reads to the debug registers. Fields:

ADDR unsigned int

address.

CORE_NUM unsigned int

Core number in a multi processor.

EXT bool

Whether access is from an external device (such as the DAP).

NS enum

Secure state of the access.

REG_NAME string

Name of the debug register accessed.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

MEMMAP_DEBUG_WRITE

Memory mapped writes to the debug registers. Fields:

ADDR unsigned int

address.

CORE_NUM unsigned int

Core number in a multi processor.

EXT bool

Whether access is from an external device (such as the DAP).

NS enum

Secure state of the access.

REG_NAME string

Name of the debug register accessed.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The new value written.

MMU_D128_TTB_READ

This event is triggered by reads caused by a D128 translation table walk. Fields:

ABORTED bool

Was the walk successful.

DATA_HIGH unsigned int

The data upper bits read.

DATA_LOW unsigned int

The data lower bits read.

IPA unsigned int

For stage 1, the IPA of the read.

LEVEL unsigned int

Translation table level.

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

OUTADDR unsigned int

The output physical address.

PADDR unsigned int

The physical address of the read.

PASmem enum

Physical Address Space of the accessed memory.

PASreq enum

Physical Address Space of the requested lookup.

REGIME_EL enum

Entry matches in this translation regime.

REQUESTER enum

The requester of this table walk.

SIDE enum

Inst / Data.

STAGE unsigned int

Translation stage.

MMU_D128_TTB_WRITE

This event is triggered by writes caused by a D128 translation table walk. Fields:

ABORTED bool

Was the walk successful.

DATA_HIGH unsigned int

The data upper bits written.

DATA_LOW unsigned int

The data lower bits written.

IPA unsigned int

For stage 1, the IPA of the write.

IS_HACDBS_ACCESS bool

If the access was due to HACDBS_CLEAN.

LEVEL unsigned int

Translation table level.

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

OUTADDR unsigned int

The output physical address.

PADDR unsigned int

The physical address of the write.

PASmem enum

Physical Address Space of the accessed memory.

PASreq enum

Physical Address Space of the requested lookup.

REGIME_EL enum

Entry matches in this translation regime.

SIDE enum

Inst / Data.

STAGE unsigned int

Translation stage.

MMU_TRANS

Address translation information. Fields:

ASID unsigned int

Address space identifier.

CORE_NUM unsigned int

Core number in a multi processor.

Hyp bool

Entry matches in Hyp state only.

INNERCACHE_RA bool

Is the inner cache allocate on read.

INNERCACHE_TRANSIENT bool

Is the inner write-through transient.

INNERCACHE_TYPE enum

Inner Caching scheme (NC/MB/WA).

INNERCACHE_WA bool

Is the inner cache allocate on write.

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MEMTYPE enum

Memory type.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC enum

Is secure side supposed to access secure or nonsecure memory.

OUTERCACHE_RA bool

Is the outer cache allocate on read.

OUTERCACHE_TRANSIENT bool

Is the outer write-through transient.

OUTERCACHE_TYPE enum

Outer Caching scheme (NC/MB/WA).

OUTERCACHE_WA bool

Is the outer cache allocate on write.

PADDR unsigned int

Physical address of the access.

PAGESIZE unsigned int

Page size as log2(size).

PAS enum

PAS supposed to be accessed is secure or nonsecure memory or Root/Realm if RME is implemented.

SH enum

Shareability.

SIDE enum

Inst / Data.

STAGE1_PERM unsigned int

Stage 1 permissions, [5:3] Privileged access (XWR) [2:0] User access (XWR).

STAGE2_PERM unsigned int

Stage 2 permission mask [5:3] Privileged access (XWR) [2:0] User access (XWR).

VADDR unsigned int

Virtual address of the access.

VMID unsigned int

Virtual machine identifier.

XS bool

XS Attribute.

nG enum

Flag indicating whether ASID will be matched.

MMU_TTB_READ

This event is triggered by reads caused by a translation table walk. Fields:

ABORTED bool

Was the walk successful.

DATA unsigned int

The data read.

DESC_KIND enum

The encoding scheme of the descriptors used for this walk.

IPA unsigned int

For stage 1, the IPA of the read.

LEVEL unsigned int

Translation table level.

LPAE bool

Is this for an LPAE translation (*DEPRECATED*, use DESC_KIND).

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSmem bool

Non secure memory access.

NSreq bool

Non secure request.

OUTADDR unsigned int

The output physical address.

PADDR unsigned int

The physical address of the read.

PASmem enum

Physical Address Space of the accessed memory.

PASreq enum

Physical Address Space of the requested lookup.

REGIME_EL enum

Entry matches in this translation regime.

REQUESTER enum

The requester of this table walk.

SIDE enum

Inst / Data.

STAGE unsigned int

Translation stage.

MMU_TTB_WRITE

This event is triggered by writes caused by a translation table walk. Fields:

ABORTED bool

Was the walk successful.

DATA unsigned int

The data written.

DESC_KIND enum

The encoding scheme of the descriptors used for this walk.

IPA unsigned int

For stage 1, the IPA of the write.

IS_HACDBS_ACCESS bool

If the access was due to HACDBS_CLEAN.

LEVEL unsigned int

Translation table level.

LPAE bool

Is this for an LPAE translation (*DEPRECATED*, use DESC_KIND).

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSmem bool

Non secure memory access.

NSreq bool

Non secure request.

OUTADDR unsigned int

The output physical address.

PADDR unsigned int

The physical address of the write.

PASmem enum

Physical Address Space of the accessed memory.

PASreq enum

Physical Address Space of the requested lookup.

REGIME_EL enum

Entry matches in this translation regime.

SIDE enum

Inst / Data.

STAGE unsigned int

Translation stage.

MODE_CHANGE

Mode change. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

MODE enum

The new mode.

NON_SECURE enum

The core's new non-secure bit.

OLD_MODE enum

The old mode.

SECURITY_STATE enum

The new security state.

NOT_TAG_CHECKED_INST

MTE Tag not checked for current instruction.

PERIODIC

Called for every quantum. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The instruction count of this CPU.

PC unsigned int

The address of the next instruction to be executed on this CPU.

PMU_COUNTER_OVERFLOW

PMU counter overflow. Fields:

COUNTER_GROUP enum

Counter group.

COUNTER_RANGE enum

Counter range.

EVENT_ID unsigned int

Event Identifier, field is valid only for Event counter group.

INDEX unsigned int

Counter index (as selected by PMSELR), field is valid only for Event/Cycle counter groups.

INTERRUPT bool

Is interrupt enabled on overflow for this counter.

POTINDEX_UPDATE

Value of POT index for current pc changed. Fields:

NEW_POTINDEX unsigned int

New POTIndex.

OLD_POTINDEX unsigned int

Old POTIndex.

PREFETCH_MEMORY64

Prefetch from PRFM or PRFUM instructions. Fields:

PRFOP enum

Prefetch hint.

VADDR unsigned int

Virtual address of the location that should be prefetched.

PRELOAD64_REQUEST

Preload requests from PRFM or PRFUM instructions. Fields:

PRFOP enum

Prefetch hint.

VADDR unsigned int

Virtual address of the location that should be prefetched.

PRELOAD_DATA

Data preload from PLD instruction. Fields:

VADDR unsigned int

Virtual address of the data that should be preloaded.

PRELOAD_INST

Instruction preload from PLI instruction. Fields:

VADDR unsigned int

Virtual address of the instruction that should be preloaded.

PRE_CORE_LOAD

Trace just before a core load. Fields:

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

PRE_CORE_STORE

Trace just before a core store. Fields:

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

PRE_TTB_READ

This event is triggered before reads caused by a translation table walk. Fields:

ADDR unsigned int

The physical address of the read.

PSTATE

PSTATE fields update. Fields:

ALLINT unsigned int

All interrupt mask bit.

BTYPE unsigned int

Branch target identification bit.

CORE_NUM unsigned int

Core number in a multi processor.

DAIF unsigned int

Exception mask bits.

DIT unsigned int

Data Independent Timing bit.

EL unsigned int

Current Exception level field.

EXLOCK unsigned int

Exception return state lock bit.

IL unsigned int

Illegal Execution state bit.

NZCV unsigned int

Condition flags.

PACM unsigned int

PACM bit.

PAN unsigned int

Privileged Access Never state bit.

PM unsigned int

PMU exception mask bit.

PPEND unsigned int

PMU exception pending bit.

SM unsigned int

StreamingMode status bit.

SP unsigned int

Stack pointer register selection bit.

SS unsigned int

Software Step bit.

SSBS unsigned int

Speculative Store Bypass Safe bit.

TCO unsigned int

Tag Check Override bit.

TOUCHED unsigned int

Mask of the fields that have been touched since this trace has last been emitted. Note that touched does not necessarily mean the value changed. The index of the field is the bit index in the mask plus the index of this field plus one.

UAO unsigned int

User Access Override bit.

UINJ unsigned int

Undefined Instruction exceptions bit.

ZA unsigned int

ZA storage and ZT0 register enable bit.

nRW unsigned int

Current Execution state bit.

PSTATE_PACTIVE_OPMODE_TRANSITION

Trace active and Opmode mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Operating mode after the transition.

OLD_STATE enum

Operating mode before the transition.

PSTATE_PACTIVE_OPMODE_TRANSITION_DENIED

Trace the denied active and Opmode mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Operating mode after the transition.

OLD_STATE enum

Operating mode before the transition.

REASON string

Reason for the transition being denied.

PSTATE_PACTIVE_POWER_TRANSITION

Trace active and power mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Power mode after the transition.

OLD_STATE enum

Power mode before the transition.

PSTATE_PACTIVE_POWER_TRANSITION_DENIED

Trace the denied active and power mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Power mode after the transition.

OLD_STATE enum

Power mode before the transition.

REASON string

Reason for the transition being denied.

PSTATE_PACTIVE_THREAD_OPMODE_TRANSITION

Trace pactive and Opmode mode transitions for threads on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Operating mode after the transition.

OLD_STATE enum

Operating mode before the transition.

QUANTUM_START

Called at the start of every quantum.

RANGE_PRELOAD_REQUEST

Preload requests from RPRFM. Fields:

COUNT unsigned int

Number of blocks of data to access.

LENGTH unsigned int

Number of contiguous bytes to be accessed in each block.

OP unsigned int

Range prefetch operation .

REUSE unsigned int

Maximum number of bytes to be accessed before next RPRFM .

STRIDE signed int

Stride to advance after accessing LENGTH bytes.

VADDR unsigned int

Base virtual address of range prefetch request.

RUN_STATE

Run state transition. Fields:

INST_COUNT unsigned int

Ticks count at point of transition.

NEW enum

New run state.

OLD enum

Old run state.

SEMIHOSTING_CALL

Call of a semihost function occurred. Fields:

PC unsigned int

The program counter after the semihosting call.

SEMIHOSTING_PRECALL

About to call semihost. Fields:

EL2 **bool**

Use EL2 translation regime.

NS **bool**

Is Non-Secure.

REG_WIDTH **unsigned int**

The current register width in bytes.

SIGNAL

External signal state change. Fields:

SIGNAL **enum**

Signal that changed.

STATE **bool**

Signal asserted state.

SOFTWARE_STEP

Return the Debug Software Step state. Fields:

STATE **enum**

Software Step state.

SPSR

SPSR change. Fields:

CORE_NUM **unsigned int**

Core number in a multi processor.

MODE **enum**

Which of the banked SPSR registers is written.

OLD_VALUE **unsigned int**

The old SPSR value.

VALUE **unsigned int**

The new SPSR value.

START_COMPILE

Compilation started. Fields:

VADDR **unsigned int**

Instruction where compilation begins.

STG_MULTIPLE_MICRO_OPS_ADDRESS_RANGE

Address range for MTE stores with multiple uops. Fields:

ADDR_END **unsigned int**

Final address.

ADDR_START unsigned int

Starting address.

STORE_MULTIPLE_REGS_ADDRESS_RANGE

Address range for stores with multiple registers. Fields:

ADDR_END unsigned int

Final address.

ADDR_START unsigned int

Starting address.

SUBOPTIMAL_LDST_RETIRED

suboptimal load/store. Fields:

ACCESS_SIZE unsigned int

Log2 of the access width in bytes used for alignment checking (1: half-word, 2: word ...).

OPTIMAL_ALIGN_SELECTOR unsigned int

Selector to decide the optimal alignment size (1: access width, 2: 4-bytes, 3: 8-bytes, ..., 12: 4-kbytes).

VADDR unsigned int

The virtual address of the access.

SYNC

Called for every synchronization, that is every quantum break. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The instruction count of this CPU.

LOCAL_QUANTUM unsigned int

The local quantum of this CPU.

LOCAL_TIME unsigned int

The local time of this CPU.

SYNC_REASON enum

Reason for sync.

SYSCALL

System call instruction executed. Fields:

IMM unsigned int

Immediate value of the system call instruction.

TYPE enum

System call type.

VADDR unsigned int

Instruction that caused the system call.

SYSREG_READ128

System Coprocessor 128 bit register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

REGNUM enum

Internal register number.

REGNUM_PHYS enum

Internal register number physically accessed (e.g. in presense of VHE redirection).

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_READ64

System Coprocessor register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REGNUM enum

Internal register number.

REGNUM_OPERAND enum

Register number corresponding to the instruction operand.

REGNUM_PHYS enum

Internal register number physically accessed (e.g. in presense of VHE redirection).

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_UPDATE128

Triggers when the system updates a register. Fields:

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG enum

Register number.

UNKNOWN unsigned int

Bits of the register which became unknown.

VALUE unsigned int

Value written to the register.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_UPDATE32

Triggers when the system updates a register. Fields:

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG enum

Register number.

UNKNOWN unsigned int

Bits of the register which became unknown.

VALUE unsigned int

Value written to the register.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_UPDATE64

Triggers when the system updates a register. Fields:

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG enum

Register number.

UNKNOWN unsigned int

Bits of the register which became unknown.

VALUE unsigned int

Value written to the register.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int
opcode 2.

SYSREG_WRITE128

System Coprocessor 128 bit register write. Fields:

CORE_NUM unsigned int
Core number in a multi processor.

CRm unsigned int
CRm.

CRn unsigned int
CRn.

REGNUM enum
Internal register number.

REGNUM_PHYS enum
Internal register number physically accessed (e.g. in presense of VHE redirection).

UNDEF bool
The register accessed is undefined.

UPDATED_VALUE unsigned int
Updated value of the register now it has been written.

VALUE unsigned int
The new value written.

opc unsigned int
opcode 1.

opc0 unsigned int
opcode 0.

opc2 unsigned int
opcode 2.

SYSREG_WRITE64

System Coprocessor register write. Fields:

CORE_NUM unsigned int
Core number in a multi processor.

CRm unsigned int
CRm.

CRn unsigned int
CRn.

NS bool
Secure or nonsecure banked register is accessed.

REGNUM enum

Internal register number.

REGNUM_OPERAND enum

Register number corresponding to the instruction operand.

REGNUM_PHYS enum

Internal register number physically accessed (e.g. in presense of VHE redirection).

UNDEF bool

The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

UNALIGNED_LDST_RETIRED

Processor unaligned load/store. Fields:

ACCESS_SIZE unsigned int

Log2 of the access width in bytes used for alignment checking (1=>half-word, 2->word ...).

NUMBER_OF_BEATS unsigned int

Number of accesses (beats) in this burst (Total size in bytes is ACCESS_SIZE * NUMBER_OF_BEATS).

VADDR unsigned int

The virtual address of the access.

VFP_D_REGS

VFP/NEON D 64 bit register write. Fields:

ALIASING enum

Type of register aliasing used.

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a write to the corresponding bit in the register. For example, 0x1 would indicate a write to VALUE[0].

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

VFP_D_REGS_READ

VFP/NEON D 64 bit register read. Fields:

ID unsigned int

The register number.

VALUE unsigned int

The current value of the register.

VFP_Q_REGS

VFP/NEON Q 128 bit register write. Fields:

ALIASING enum

Type of register aliasing used.

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a write to the corresponding byte in the register. For example, 0x1 would indicate a write to VALUE[7:0].

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

VFP_Q_REGS_READ

VFP/NEON Q 128 bit register read. Fields:

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a read to the corresponding byte in the register. For example, 0x1 would indicate a read to VALUE[7:0].

VALUE unsigned int

The current value of the register.

VFP_SYS_REGS

Writes to the VFP/NEON units system registers. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ID enum

Which VFP system register is written.

MASK unsigned int

The mask indicating updated cumulative bits.

OLD_VALUE unsigned int

The register's old value overwritten.

VALUE unsigned int

The new value written to the VFP system register.

VFP_S_REGS

VFP/NEON S 32 bit register write. Fields:

ALIASING enum

Type of register aliasing used.

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a write to the corresponding bit in the register. For example, 0x1 would indicate a write to VALUE[0].

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

VFP_S_REGS_READ

VFP/NEON S 32 bit register read. Fields:

ID unsigned int

The register number.

VALUE unsigned int

The current value of the register.

VIRTUAL_TAG_ALLOCATED

Virtual tag information when allocated. Fields:

TAG_VADDR unsigned int

Virtual tag virtual address.

VADDR unsigned int

Data Virtual address.

VIRTUAL_TAG unsigned int

Virtual tag value.

WAYPOINT

Signals end of basic block execution. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ISET enum

Origin instruction set.

IS_COND bool

Indicates if this is a conditional waypoint.

PC unsigned int

Origin address (or 0 if unavailable).

TAKEN bool

Indicates if this waypoint was taken.

TARGET unsigned int

Destination address.

TARGET_ISET enum

Destination instruction set.

WFE_END

WFE ended. Fields:

INST_COUNT unsigned int

Ticks count when leaving WFE.

WFE_EVENT_REGISTER

WFE event register status: set/clear, reason. Fields:

INST_COUNT unsigned int

Ticks count.

REASON enum

Reason for set/clear. Only REASON==0 (Cleared by WFE) clears the bit, all other reasons set it.

WFE_IGNORED

WFE ignored. Fields:

EVENT bool

This WFE was ignored because the event register was set.

INST_COUNT unsigned int

Ticks count when ignoring WFE.

TRAPPED bool

This WFE was trapped.

WFE_START

WFE entered. Fields:

INST_COUNT unsigned int

Ticks count when entering WFE.

WFI_END

WFI ended. Fields:

INST_COUNT unsigned int

Ticks count when leaving WFI.

WFI_IGNORED

WFI ignored. Fields:

DISABLED bool

This WFI was ignored because WFI is disabled.

INST_COUNT unsigned int

Ticks count when ignoring WFI.

REASON enum

specifies reason why WFI trace was ignored.

TRAPPED bool

This WFI was trapped.

WFI_START

WFI entered. Fields:

INST_COUNT unsigned int

Ticks count when entering WFI.

WFI_WAKEUP

WFI wakeup. Fields:

INST_COUNT unsigned int

Ticks count when WFI wakeup occurred.

REASON enum

Reason for wakeup.

2.24 ARM_Cortex-A710

This section describes the trace sources.

AA64_ASE_SVE_REGS

Modification of AArch64 Advanced SIMD or SVE Vector Registers (via SIMD or Scalar operations), SVE Predicate Registers and ZT0. Fields:

ID **enum**

The register identifier.

MASK **unsigned int**

Bitmask of the register to signal the modified bits in the VALUE field.

SM **bool**

Whether the PE is in Streaming Mode.

VALUE **unsigned int**

Value of AArch64 vector register.

ASYNC_MEMORY_FAULT

Context ID Register write. Fields:

FAULT **unsigned int**

Fault status present in ESR.ISS encoding for Serror in ARCH64 and present in DFSR encoding for external abort format in ARCH32.

PADDR **unsigned int**

Physical Address (or 0 if unavailable).

VADDR **signed int**

Virtual Address (or 0 if unavailable).

ATOMIC_END_ACCESS

Bus trace access for atomics. Fields:

ACCESS_FAIL **bool**

Memory access failed.

ADDR **unsigned int**

The virtual address of the access.

ATTR **unsigned int**

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

COMPARE_VALUE **unsigned int**

Compare value for CAS.

LOAD_VALUE unsigned int

Loaded values.

NSDESC unsigned int

The security state of the access.

OPERAND_VALUE unsigned int

Operation's operand.

OPERATION enum

Operation type.

PADDR unsigned int

The physical address of the access.

PRIV bool

Is this a privileged access?.

ATOMIC_START_ACCESS

Bus trace access for atomics. Fields:

ADDR unsigned int

The virtual address of the access.

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

COMPARE_VALUE unsigned int

Compare value for CAS.

NSDESC unsigned int

The security state of the access.

OPERAND_VALUE unsigned int

Operation's operand.

OPERATION enum

Operation type.

PADDR unsigned int

The physical address of the access.

PRIV bool

Is this a privileged access?.

ArchMsg.Error.exit_code

DISPLAY exit code: %{EXITCODE} component kind: %{KIND} reason: %{REASON}. Fields:

EXITCODE signed int

Exit code to be returned.

KIND string

Component kind that invoked the exit code trace.

REASON string

Reason why the exit code trace was invoked (optional).

ArchMsg.Warning.IgnoredWarmResetRequest

Event fires when software attempts to request warm reset through write to deprecated DBGPRCR.CWRR.

ArchMsg.Warning.RAS_ErrselrUnpredictableState

DISPLAY ERRSELR is in unpredictable state due to value greater than number_of_error_records written to it.

ArchMsg.Warning.RAS_UnpredictableStateWrite

DISPLAY %{REG_NAME} write ignored due to unpredictable ERRSELR state. Fields:

REG_NAME string

Name of register being written.

ArchMsg.Warning.ReportedRevisionRegisterRead

DISPLAY Reported revision and patch do not match that of the cpu. Fields:

ORIGINAL_VALUE unsigned int

The value the register would have based on the modelled revision.

REGISTER string

The name of the register that has ben read.

REPORTED_VALUE unsigned int

The value the register has been overridden to.

ArchMsg.Warning.Unpredictably Indexed PM Event Register

TAGS UNPREDICTABLE DISPLAY Event %{IsDirect:(Selector value|Register)} %{SEL} is >= %{N}. Fields:

IsDirect unsigned int

Direct Access rather than indirect.

N unsigned int

Number of Accessible Registers.

SEL unsigned int

Selector.

ArchMsg.Warning.cache_contents_unknown

A cache has been activated without being invalidated and may contain **UNKNOWN** content. DISPLAY %{SIDE:(D|I)}-Cache was enabled in %{REGIME:(SECURE|NON_SECURE|HYP)} regime but was not invalidated since power-on: cache lines could contain **UNKNOWN** content %{SIDE:(Note: D-side caches must invalidate SECURE lines even if the cache is only used by NON_SECURE code in order to ensure that unexpected dirty lines tagged SECURE are not naturally evicted)}. Fields:

REGIME unsigned int

Current translation regime.

SIDE unsigned int

0 d-side cache or unified cache, 1 i-side.

ArchMsg.Warning.dcimvac_matches_watchpoint

TAGS **UNPREDICTABLE** DISPLAY Watchpoints matching an AArch32 DCIMVA are implementation defined.

ArchMsg.Warning.decode_fieldmismatch

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with different fields %{FIELD1} and %{FIELD2}. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_fracbitsnegative

TAGS **UNPREDICTABLE** DISPLAY VCVT Instruction is unpredictable with a negative number of fraction bits.

ArchMsg.Warning.decode_initblock

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable in an IT block.

ArchMsg.Warning.decode_initblocknotlast

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable in an IT block when not the last.

ArchMsg.Warning.decode_invalid_condition

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable when the condition is not AL.

ArchMsg.Warning.decode_invalidfieldcombination

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with the values in %{FIELD1} and %{FIELD2}. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_invalidvalue

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with the value in %{FIELD}. Fields:

FIELD enum

Field name.

ArchMsg.Warning.decode_mem_hint_unallocated

TAGS **UNPREDICTABLE** DISPLAY There is no memory hint allocated to this bit pattern.

ArchMsg.Warning.decode_registermatch

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with %{FIELD1} and %{FIELD2} as the same register. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_registermismatch

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with %{FIELD1} and %{FIELD2} as different registers. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_registeroutofrange

TAGS **UNPREDICTABLE** DISPLAY Floating-point instruction is unpredictable indexing beyond the end of the register bank.

ArchMsg.Warning.decode_sbzsbo

TAGS **UNPREDICTABLE** DISPLAY Reserved bits in the instruction are not canonical.

ArchMsg.Warning.decode_transactiontoobig

TAGS **UNPREDICTABLE** DISPLAY Attempt to perform a memory transaction of more than 128 bytes.

ArchMsg.Warning.decode_unpred_other

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable.

ArchMsg.Warning.decode_unpreduseofpc

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable when %{FIELD} is PC. Fields:

FIELD enum

Field name.

ArchMsg.Warning.decode_unpreduseofr13

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable when %{FIELD} is R13. Fields:

FIELD enum

Field name.

ArchMsg.Warning.decode_unpreduseofr13inreglist

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with R13 in register-list.

ArchMsg.Warning.decode_useofd16orhigher

TAGS **UNPREDICTABLE** DISPLAY Attempt to use register D16 or higher where it is not allowed.

ArchMsg.Warning.decode_writebackandbaseinlist

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with write back when the base register is in the transfer list.

ArchMsg.Warning.decode_zeroregistersinlist

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable when no registers are to be transferred.

ArchMsg.Warning.misaligned_ptrs

DISPLAY TRBPTR_EL1(%{TRBPTR_EL1}) or TRBTRG_EL1(%{TRBTRG_EL1}) are not aligned as per TRBIDR_EL1.Align(%{TRBIDR_EL1}). Fields:

TRBIDR_EL1 unsigned int

TRBIDR_EL1.Align value.

TRBPTR_EL1 unsigned int

Current write pointer.

TRBTRG_EL1 unsigned int

Trigger pointer.

ArchMsg.Warning.out_of_range_ptrs

DISPLAY TRBPTR_EL1(%{TRBPTR_EL1}) is out of range as per TRBBASER_EL1(%{TRBBASER_EL1}) or TRBLIMITR_EL1.LIMIT(%{TRBLIMITR_EL1}). Fields:

TRBBASER_EL1 unsigned int

Base pointer.

TRBLIMITR_EL1 unsigned int

Limit pointer.

TRBPTR_EL1 unsigned int

Current write pointer.

ArchMsg.Warning.recursive_branch

An instruction is performing a branch that targets the same instruction. If this was intended then a WFI might be more effective DISPLAY recursive branch detected at PC=%{PC}.

PRIMARY KEY PC. Fields:

PC unsigned int

Address of instruction causing the branch.

ArchMsg.Warning.recursive_exception

An instruction has generated an exception that targets the same instruction. DISPLAY recursive %{TYPE} exception detected at PC=%{PC} (CPSR=%{CPSR}). PRIMARY KEY PC. Fields:

CPSR unsigned int

Processor state.

PC unsigned int

Address of instruction causing the exception.

TYPE enum

Type of exception.

ArchMsg.Warning.reentrant_vector_catch

TAGS **UNPREDICTABLE** DISPLAY Vector catch debug event on Prefetch or Data abort vector %{ADDR}. Fields:

ADDR unsigned int

vector address.

ArchMsg.Warning.reserved_it_state

TAGS **UNPREDICTABLE** DISPLAY Execution with reserved IT state %{ITSTATE}. Fields:

ITSTATE unsigned int

reserved IT state.

ArchMsg.Warning.tlb_contents_unknown

Translation has been enabled without invalidating the TLB. DISPLAY Translation was enabled in the %{PAS:(Secure|Non-secure|Root|Realm)} EL%{EL:d} translation regime (vmid=%{VMID:x}) without invalidating %{INVALIDITY:(|d-side |i-side |)}entries since power-on: %{INVALIDITY:(|D|I|)}TLB could contain **UNKNOWN** content. Fields:

EL unsigned int

Exception level of translation regime.

INVALIDITY unsigned int

Which TLB may be invalid (if TLBs are separate).

NON_SECURE bool

Whether the translation regime is non-secure.

PAS unsigned int

Physical address space of translation regime.

VMID unsigned int

Current VMID, if applicable.

ArchMsg.Warning.unknown_DLR_DSPSR

Exiting debug state with unknown DLR or DSPSR.

ArchMsg.Warning.unknown_ELR_SPSR

Returning from debug exception with unknown ELR or SPSR.

ArchMsg.Warning.unpred_FAR_on_LRCPC3_non_atomic_mem_access

TAGS **UNPREDICTABLE** Unpredictable FAR/FSTATUS on LRCPC3 pair instructions resulting in two separate memory accessesDISPLAY Unpredictable FAR on LRCPC3 non atomic memory access at %{ADDR1} with FSTATUS %{FSTATUS1} or %{ADDR2} with FSTATUS %{FSTATUS2}. Fields:

ADDR1 unsigned int

Possible FAR address 1.

ADDR2 unsigned int

Possible FAR address 2.

FSTATUS1 unsigned int

Possible fault status value 1.

FSTATUS2 unsigned int

Possible fault status value 2.

ArchMsg.Warning.unpred_async_tag_check_fail_on_sve_speculative_load_do_update_ffr

Unpredictable set to FALSE of the FFR element associated to SVE speculative load with an async tag check fail TAGS **UNPREDICTABLE**.

ArchMsg.Warning.unpredictable_a32_breakpoint

TAGS **UNPREDICTABLE** DISPLAY Unpredictable breakpoint on a A32 instruction at %{ADDR1} Offset %{OFFSET} BAS %{BAS} for a %{IS_ADDRESS_MISMATCH_BREAKPOINT:(Mismatch|Match)} Breakpoint number %{N:d} Breakpoint address %{ADDR2}. Fields:

ADDR1 unsigned int

Address.

ADDR2 unsigned int

Breakpoint address.

BAS unsigned int

BAS field.

IS_ADDRESS_MISMATCH_BREAKPOINT bool

Is a Mismatch breakpoint, not a Match breakpoint.

N unsigned int

Breakpoint number.

OFFSET unsigned int

Always has a value of zero.

ArchMsg.Warning.unpredictable_banked_register_access

TAGS **UNPREDICTABLE** DISPLAY Access to banked register (R=%{R:d} SYSm=%{SYSM:2x}) is unpredictable. Fields:

R unsigned int

R bit.

SYSM unsigned int

SYSm.

ArchMsg.Warning.unpredictable_ccfail_undef

TAGS **UNPREDICTABLE** DISPLAY An instruction %{OPCODE} at VA=%{VADDR} failed its condition codes check but would otherwise have caused a non-data-dependent trap or undefined exception. Behaviour is implementation-defined choice of exception or nop. Fields:

OPCODE unsigned int

opcode of the instruction.

VADDR unsigned int

Virtual address of the instruction being executed.

ArchMsg.Warning.unpredictable_ici_bits

TAGS **UNPREDICTABLE** DISPLAY ICI bits do not apply or are architecturally unpredictable for instruction.

ArchMsg.Warning.unpredictable_t32_breakpoint

TAGS **UNPREDICTABLE** DISPLAY Unpredictable breakpoint on a T32 instruction at %{ADDR1} Offset %{OFFSET} BAS %{BAS} for a %{IS_ADDRESS_MISMATCH_BREAKPOINT:(Mismatch|Match)} Breakpoint number %{N:d} Breakpoint address %{ADDR2}. Fields:

ADDR1 unsigned int

Address.

ADDR2 unsigned int

Breakpoint address.

BAS unsigned int

BAS field.

IS_ADDRESS_MISMATCH_BREAKPOINT bool

Is a Mismatch breakpoint, not a Match breakpoint.

N unsigned int

Breakpoint number.

OFFSET unsigned int

Indicates which halfword of a 32-bit Thumb instruction the breakpoint is set on. 0 for the first halfword or if the instruction is 16 bits wide, or 1 for the second halfword.

ArchMsg.Warning.unpredictable_t32_vector_catch

TAGS **UNPREDICTABLE** DISPLAY Unpredictable Vector catch exception on a T32 instruction at %{ADDR} Offset %{OFFSET}. Fields:

ADDR unsigned int

Address.

OFFSET unsigned int

Indicates which halfword of a 32-bit Thumb instruction the vector catch exception is set on. 0 for the first halfword or if the instruction is 16 bits wide, or 1 for the second halfword.

ArchMsg.Warning.unpredictable_watchpoint_far

TAGS **UNPREDICTABLE** DISPLAY The value for FAR/EDWAR for hitting this watchpoint may be anywhere in the range %{LOWER_BOUND} to %{UPPER_BOUND}. Fields:

LOWER_BOUND unsigned int

the lowest address accessed by the instruction that triggered the watchpoint.

UPPER_BOUND unsigned int

the highest watchpointed address accessed by that instruction.

ArchMsg.Warning.unpredictable_watchpoint_far_isscmp

DISPLAY The value for FAR/EDWAR for hitting this watchpoint may be anywhere in the range %{LOWER_BOUND} to %{UPPER_BOUND}. Fields:

LOWER_BOUND unsigned int

the lowest address accessed by the instruction that triggered the watchpoint.

UPPER_BOUND unsigned int

the highest watchpointed address accessed by that instruction.

ArchMsg.Warning.warning_AdvSIMDExpandImmUnexpectedZero

TAGS UNPREDICTABLE DISPLAY AdvSIMDExpandImm may treat this immediate value as UNPREDICTABLE.

ArchMsg.Warning.warning_ConditionalSMC

TAGS UNPREDICTABLE DISPLAY SMC instruction has UNPREDICTABLE effects when conditional and when combined with traps.

ArchMsg.Warning.warning_TLBID_out_of_range

TLBID out of range in TLBI according to the configuration of NIS, NVIS, NOS and NVOS. Fields:

NIS unsigned int

NIS.

NOS unsigned int

NOS.

NVIS unsigned int

NVIS.

NVOS unsigned int

NVOS.

TLBID unsigned int

The TLBID value used in the TLBI instruction.

ArchMsg.Warning.warning_TTL_or_TG_mismatches_current_translation_granule

TTL or TG mismatches current translation granule in TLBI VA/RVA/IPA/RIPA. Fields:

CURRENT_TG unsigned int

Current TG as indicated by TCR_ELx or VTCR_EL2.

TLBI_TG unsigned int

TG (or TTL[3:2]) used in TLBI instruction.

ArchMsg.Warning.warning_access_crosses_page_boundary

TAGS UNPREDICTABLE DISPLAY Access crosses page boundary.

ArchMsg.Warning.warning_access_crosses_page_boundary_has_fault

TAGS UNPREDICTABLE DISPLAY Access that crosses page boundary has fault(s).

ArchMsg.Warning.warning_access_crosses_page_boundary_has_fault_on_both_pages

TAGS **UNPREDICTABLE** DISPLAY Access that crosses page boundary has a fault on each page.

Fields:

ADDR1 unsigned int

FAR address on first page.

ADDR2 unsigned int

FAR address on second page.

FSTATUS1 unsigned int

Fault status on first page.

FSTATUS2 unsigned int

Fault status on second page.

ArchMsg.Warning.warning_access_crosses_page_boundary_spanning_different_memory

TAGS **UNPREDICTABLE** DISPLAY Access at address %{ADDR} crosses page from MEM_TYPE %{MEMTYPE_PAGE1} to %{MEMTYPE_PAGE2}. Fields:

ADDR unsigned int

address of access crossing page.

MEMTYPE_PAGE1 signed int

First Page's memory type (0-Normal, 1-Non-Normal).

MEMTYPE_PAGE2 signed int

Second Page's memory type (0-Normal, 1-Non-Normal).

ArchMsg.Warning.warning_access_wraps_around_memory

TAGS **UNPREDICTABLE** DISPLAY Access wraps around memory in %{WIDTH:d}bit mode.

Fields:

WIDTH unsigned int

Address width.

ArchMsg.Warning.warning_bcr_linking_status

TAGS **UNPREDICTABLE** DISPLAY DBGBCR.{BT2,BT} combination gives unpredictable behaviour.

ArchMsg.Warning.warning_bcr_mask_reserved

TAGS **UNPREDICTABLE** DISPLAY DBGBCR_ELn.MASK is set to a reserved value.

ArchMsg.Warning.warning_bvr_masked_bit_not_zero

TAGS **UNPREDICTABLE** DISPLAY DBGBVRn_EL1.VA contains a bit masked by DBGBCRn_EL1.MASK that is not zero.

ArchMsg.Warning.warning_ccsidr_unimplemented_level

TAGS **UNPREDICTABLE** DISPLAY CCSIDR read while CSSELR %{CSSELR:x} points to an unimplemented cache level. Fields:

CSSELR unsigned int

current effective value of CSSELR.

ArchMsg.Warning.warning_change_to_ns_when_tge_set

TAGS **UNPREDICTABLE** DISPLAY Attempting to change NS to 1 when HCR.TGE = 1.

ArchMsg.Warning.warning_contiguous_bit_check_abort

TAGS **UNPREDICTABLE** DISPLAY An abort occurred whilst attempting to check TLB entry is contiguous at %{ENTRY_ADDR}. Fields:

ENTRY_ADDR unsigned int

address of conflicting TLB entry.

ArchMsg.Warning.warning_contiguous_bit_error

TAGS **UNPREDICTABLE** DISPLAY TLB entry at %{ENTRY_ADDR} was not contiguous with entry at %{CONTIG_ADDR} Contents are %{ENTRY_DATA} but expected %{CONTIG_DATA}. Fields:

CONTIG_ADDR unsigned int

address of the TLB first read, with which this entry is expected to be contiguous.

CONTIG_DATA unsigned int

expected contents based on the entry at CONTIG_ADDR.

ENTRY_ADDR unsigned int

address of conflicting TLB entry.

ENTRY_DATA unsigned int

contents of conflicting TLB entry.

ArchMsg.Warning.warning_contiguous_bit_error_D128

TAGS **UNPREDICTABLE** DISPLAY TLB entry at %{ENTRY_ADDR} was not contiguous with entry at %{CONTIG_ADDR} Contents are %{ENTRY_DATA_HI} %{ENTRY_DATA_LO} but expected %{CONTIG_DATA}. Fields:

CONTIG_ADDR unsigned int

address of the TLB first read, with which this entry is expected to be contiguous.

CONTIG_DATA unsigned int

expected contents based on the entry at CONTIG_ADDR.

ENTRY_ADDR unsigned int

address of conflicting TLB entry.

ENTRY_DATA_HI unsigned int

Higher bit contents of conflicting TLB entry.

ENTRY_DATA_LO unsigned int

Lower bit contents of conflicting TLB entry.

ArchMsg.Warning.warning_cp10_cp11_mismatch

TAGS **UNPREDICTABLE** DISPLAY Unpredictable configuration of CP10 and CP11 controls at EL %{EL:d}. Fields:

EL unsigned int

EL owning the control.

ArchMsg.Warning.warning_cp10_cp11_reserved_value

TAGS **UNPREDICTABLE** DISPLAY Unpredictable configuration of CPACR CP10 ({CP10}) or CP11 ({CP11}). Fields:

CP10 unsigned int

CP10 access.

CP11 unsigned int

CP11 access.

ArchMsg.Warning.warning_csselr_level_out_of_range

TAGS **UNPREDICTABLE** DISPLAY CSSELR.Level out of range, written value {LEVEL} with only {IMPLEMENTED:d} levels implemented. Fields:

IMPLEMENTED unsigned int

number of cache levels implemented.

LEVEL unsigned int

written value of CSSELR.Level.

TYPE enum

The type of cache selected by CSSELR.InD and TnD.

ArchMsg.Warning.warning_debug_flow_control_bits_not_obeyed

TAGS **UNPREDICTABLE** DISPLAY Flow control bits for {REGISTER} were not obeyed, causing {ERROR}. Fields:

ERROR enum

type of error resulting.

REGISTER enum

register under consideration.

ArchMsg.Warning.warning_debug_register_access_during_reset

TAGS **UNPREDICTABLE** DISPLAY Unpredictable {IS_WRITE:(read|write)} access to debug register offset {OFFSET} during reset. Fields:

IS_WRITE unsigned int

Write Not Read.

OFFSET unsigned int

Register Offset.

ArchMsg.Warning.warning_decode_cps_inconsistent_fields

TAGS **UNPREDICTABLE** DISPLAY The mode setting fields in the CPS instruction are inconsistent.

ArchMsg.Warning.warning_decode_invalid_state

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable in current exception-level/security state.

ArchMsg.Warning.warning_default_cacheable_mmu_on

TAGS **UNPREDICTABLE** DISPLAY Default cacheable is enabled (HCR.DC = 1) while MMU is enabled.

ArchMsg.Warning.warning_default_cacheable_vmmu_off

TAGS **UNPREDICTABLE** DISPLAY Default cacheable is enabled (HCR.DC = 1) while HCR.VM = 0.

ArchMsg.Warning.warning_deprecated_wvr_bit_2_set

TAGS **UNPREDICTABLE** DISPLAY DBGWVR%{N:d}_EL1=%{WVR:h} has bit 2 set, deprecated in ARMv8. Fields:

N unsigned int

index.

WVR unsigned int

DBGWVR.

ArchMsg.Warning.warning_eret_while_software_step_active_pending

Missing ISB between setting MDSCR_EL1.SS and ERET TAGS **UNPREDICTABLE**.

ArchMsg.Warning.warning_exclusive_to_non_normal

TAGS **UNPREDICTABLE** DISPLAY Exclusive Access to Non-Normal Memory %{ADDR}. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_exclusive_to_non_writeback

TAGS **UNPREDICTABLE** DISPLAY Exclusive Access to Non-Writeback-Cacheable Memory %{ADDR}. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_execute_from_device_memory

TAGS **UNPREDICTABLE** DISPLAY An attempt was made to execute from Device memory.

ArchMsg.Warning.warning_illegal_cpsr_mode

TAGS **UNPREDICTABLE** DISPLAY Writing an illegal or unimplement mode (%{NEW_MODE:h}) to CPSR was unpredictable. Fields:

NEW_MODE unsigned int

New value of CPSR.M.

ArchMsg.Warning.warning_illegal_srs_mode

TAGS **UNPREDICTABLE** DISPLAY Illegal or **UNPREDICTABLE** mode (%{MODE:h}) used for SRS instruction. Fields:

MODE unsigned int

mode for Banked SP.

ArchMsg.Warning.warning_implementation_defined_read_debug_register_in_SCS

Read Access to Debug registers from the processor is IMPLEMENTATION DEFINED TAGS IMP_DEF. Fields:

PPB_OFFSET unsigned int
debug register offset.

ArchMsg.Warning.warning_implementation_defined_sequential_security_transitions_supported

The behavior of sequential instruction fetches that cross from non-secure to secure memory and contain an SG instruction is CONSTRAINED UNPREDICTABLE TAGS UNPREDICTABLE.

ArchMsg.Warning.warning_implementation_defined_stack_limit_check_supported

It is IMPLEMENTATION DEFINED whether stack pointer limit checking is performed for this instruction TAGS IMP_DEF. Fields:

ADDRESS unsigned int
Address of instruction.

OPCODE unsigned int
instruction opcode.

ArchMsg.Warning.warning_implementation_defined_taggednonsharedmemorytype

TAGS IMP_DEF DISPLAY Memory attribute has Tagged, non shared memory type; it is IMPLEMENTATION DEFINED whether the memory location is treated as Tagged or non-Tagged.

ArchMsg.Warning.warning_implementation_defined_write_debug_register_in_SCS

Write Access to Debug registers from the processor is IMPLEMENTATION DEFINED TAGS IMP_DEF. Fields:

DATA unsigned int
data attempted to be written.

PPB_OFFSET unsigned int
debug register offset.

ArchMsg.Warning.warning_invalid_tcr_granule

TAGS UNPREDICTABLE DISPLAY TCR.TG%{TG_ID:(0|1)} bits have been set to a granule size %{{REQUEST:d}}K not implemented - using %{{SUBSTITUTE:d}}K. Fields:

REQUEST signed int
page size requested (or 0 for reserved).

SUBSTITUTE signed int
best guess available page size.

TG_ID bool
bits TG0 or TG1.

ArchMsg.Warning.warning_load_multiple_user_registers_from_user_mode

TAGS UNPREDICTABLE DISPLAY An LDM(user registers) instruction executed from user mode.

ArchMsg.Warning.warning_mte_in_mem_attr_not_supported

TAGS UNPREDICTABLE DISPLAY Memory attribute %{ATTR} has Tagged memory type but Armv8.5-MTE is not implemented. Fields:

ATTR unsigned int

Memory attribute.

ArchMsg.Warning.warning_reserved_breakpoint_state_match

TAGS UNPREDICTABLE DISPLAY %{IS_BREAKPOINT:(Watchpoint|Breakpoint)} programmed with a reserved combination of HMC, SSC and %{IS_BREAKPOINT:(PAC|PMC)}. Fields:

IS_BREAKPOINT unsigned int

Is a breakpoint, not a watchpoint.

ArchMsg.Warning.warning_shareability

TAGS UNPREDICTABLE DISPLAY Unpredictable: combination of the 1st and 2nd stages of translation is Normal Inner Non-Cacheable, Outer Non-Cacheable.

ArchMsg.Warning.warning_software_step_set_while_enabled

TAGS UNPREDICTABLE DISPLAY MDSCR_EL1.SS set to 1 while software step debug exceptions are enabled.

ArchMsg.Warning.warning_thumb_instruction_wraps_around_memory

TAGS UNPREDICTABLE DISPLAY Thumb instruction wraps around memory.

ArchMsg.Warning.warning_ttbr_sbz_bits_are_not_zero

TAGS UNPREDICTABLE DISPLAY %{TTBR}[%{MSB:d}:0] should be zero, but are not. Fields:

MSB unsigned int

x-1.

TTBR enum

which TTBR register.

ArchMsg.Warning.warning_unaligned_address_dbgdtrrx_write

TAGS UNPREDICTABLE DISPLAY Unaligned address %{ADDR} in X0/R0 when DBGDTRRX written in memory access mode. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unaligned_address_dbgdtrtx_read

TAGS UNPREDICTABLE DISPLAY Unaligned address %{ADDR} in X0/R0 when DBGDTRTX read in memory access mode. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unaligned_to_device

TAGS UNPREDICTABLE DISPLAY Unaligned Access to Device Memory %{ADDR}. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unaligned_to_strongly_ordered

TAGS **UNPREDICTABLE** DISPLAY Unaligned Access to Strongly Ordered Memory %{ADDR}.

Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unknown_sau_rnr

TAGS **UNPREDICTABLE** DISPLAY SAU_RNR was set to an unsupported value.

ArchMsg.Warning.warning_unpredictable_AIRCR_PRIS_and_BFHFNMINs

TAGS **UNPREDICTABLE** DISPLAY The effect of setting both AIRCR.BFHFNMINs and AIRCR.PRIS to 1 is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_AIRCR_VECTCLRACTIVE_when_not_in_debug

TAGS **UNPREDICTABLE** DISPLAY The effect of writing a 1 to AIRCR.VECTCLRACTIVE if the processor is not halted in Debug state is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_AIRCR_VECTRESET_and_SYSRESETREQ

TAGS **UNPREDICTABLE** DISPLAY When the processor is halted in Debug state, if a write to the register writes a 1 to both VECTRESET and SYSRESETREQ, the behavior is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_AIRCR_VECTRESET_when_not_in_debug

TAGS **UNPREDICTABLE** DISPLAY The effect of writing a 1 to AIRCR.VECTRESET if the processor is not halted in Debug state is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_AIRCR_incorrect_VKEY

TAGS **UNPREDICTABLE** DISPLAY The value 0x05FA must be written to AIRCR.VKEY, otherwise the register write is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_EXC_RETURN_Reserved_Bit

EXC_RETURN[23:7] are reserved with the special condition that all bits should be written as one. Values other than all 1s are **UNPREDICTABLE**. TAGS **UNPREDICTABLE**. Fields:

EXC_RETURN unsigned int

Special PC value.

SBO_MASK unsigned int

Which bits Should Be One to not UNPRED.

ArchMsg.Warning.warning_unpredictable_PMEVTYPEPERn_ELO_VS_reserved_value

TAGS **UNPREDICTABLE** DISPLAY Unpredictable reserved value of PMEVTYPER<n>_ELO.VS.

ArchMsg.Warning.warning_unpredictable_change_to_DEMCR_MON_STEP_at_insufficient_priority

TAGS **UNPREDICTABLE** DISPLAY The effect of changing DEMCR.MON_STEP at an execution priority that is lower than the priority of the DebugMonitor exception is **CONSTRAINED UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_change_to_priority_of_active_exception

TAGS UNPREDICTABLE DISPLAY Changing the priority of an active exception is UNPREDICTABLE.

ArchMsg.Warning.warning_unpredictable_clear_lspact

TAGS UNPREDICTABLE DISPLAY UNPREDICTABLE state on context switch. Fields:

DRVEXC_TAKEN unsigned int

Derived exception taken, not pended.

REACHED_LAST_STORE unsigned int

Only the last faulted.

ArchMsg.Warning.warning_unpredictable_dbg_exit_unaligned_dlr

TAGS UNPREDICTABLE DISPLAY Debug exit to AArch32 with DLR[0] set to 1. PC[0] can be set to 0 or to DLR[0].

ArchMsg.Warning.warning_unpredictable_exception_catch

TAGS UNPREDICTABLE DISPLAY Generation of an exception catch event is unpredictable.

ArchMsg.Warning.warning_unpredictable_exception_return_inconsistent_state

TAGS UNPREDICTABLE DISPLAY state on exception return is unpredictable.

ArchMsg.Warning.warning_unpredictable_exception_return_instruction

TAGS UNPREDICTABLE DISPLAY exception return instruction is unpredictable.

ArchMsg.Warning.warning_unpredictable_in_debug_state

TAGS UNPREDICTABLE DISPLAY Instruction is UNPREDICTABLE when executed in debug state.

ArchMsg.Warning.warning_unpredictable_mair_encoding

TAGS UNPREDICTABLE DISPLAY Memory attribute %{ATTR} has invalid MAIR encoding; this may lead to unpredictable results. Fields:

ATTR unsigned int

Memory attribute.

ArchMsg.Warning.warning_unpredictable_pmu_counter_access

TAGS UNPREDICTABLE DISPLAY Access to PMU counters from non-secure EL0 or EL1 is UNPREDICTABLE with MDCR_EL2.HPMN set to 0.

ArchMsg.Warning.warning_unpredictable_prioritization_breakpoint_match_vector_catch

TAGS UNPREDICTABLE DISPLAY Two events with the same priority occurred at the same instruction: Address Matching Vector Catch debug event, and Breakpoint debug event. It is constrained unpredictable which is taken.

ArchMsg.Warning.warning_unpredictable_stack_selection

TAGS UNPREDICTABLE DISPLAY UNPREDICTABLE state on context switch.

ArchMsg.Warning.warning_unpredictable_tsize_out_of_range

TAGS UNPREDICTABLE DISPLAY %{TCR}.T%{TTBR:d}SZ (%{TSIZE:d} bits) is out of range. Must be between 25 and %{TSIZE_MAX:d}. Fields:

TCR enum

Which TCR.

TSize signed int

Value of TSize.

TSize_MAX signed int

The maximum allowed value.

TTBR unsigned int

Which TTBR region.

ArchMsg.Warning.warning_unpredictable_unaligned_pc_as_base_register

TAGS **UNPREDICTABLE** DISPLAY The PC value must be word-aligned, otherwise the behavior of the instruction is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_unaligned_pop_stack

TAGS **UNPREDICTABLE** DISPLAY **CONSTRAINED UNPREDICTABLE** unaligned stack frame `%{FRAME_ADDR:x}` in exception-return. Fields:

FRAME_ADDR unsigned int

unaligned stack frame address.

ArchMsg.Warning.warning_unpredictable_unaligned_to_non_writeback

TAGS **UNPREDICTABLE** DISPLAY Unaligned Access to Non-Writeback-Cacheable Memory.

ArchMsg.Warning.warning_unpredictable_vmsa_memattrib

TAGS **UNPREDICTABLE** DISPLAY Unpredictable vmsa memory attribute with texcb value programmed to `%{TEXCB}` when tex remap is `%{TEX_REMAP}`. Fields:

TEXCB unsigned int

texcb value.

TEX_REMAP bool

use tex remap.

ArchMsg.Warning.warning_unpredictable_vtcr_t0sz_sl0

TAGS **UNPREDICTABLE** DISPLAY The combination of VTCR.TOSZ=`%{TOSZ:d}` and VTCR.SL0=`%{SL0:d}` is unpredictable. Fields:

SL0 signed int

VTCR.SL0, starting level for stage 2 translation table walks.

TOSZ signed int

VTCR.TOSZ, required input address range.

ArchMsg.Warning.warning_unpredictable_write_DHCSR

TAGS **UNPREDICTABLE** DISPLAY **UNPREDICTABLE** set of bit-changes in DHCSR. Fields:

NEW unsigned int

data attempted to be written.

OLD unsigned int

previous value.

ArchMsg.Warning.warning_unpredictable_write_SHCSR

TAGS UNPREDICTABLE DISPLAY UNPREDICTABLE setting a pending bit to 1 for an exception with priority \geq execution priority.

ArchMsg.Warning.warning_unsupported_access_to_memory_mapped_register

TAGS UNPREDICTABLE DISPLAY Access to memory mapped register is unsupported.

ArchMsg.Warning.warning_user_jmcr_access

TAGS UNPREDICTABLE DISPLAY User %{WRITE:(write|read)} access to JMCR is unpredictable. Fields:

WRITE unsigned int
access is write.

ArchMsg.Warning.warning_wcr_mask_and_bas

TAGS UNPREDICTABLE DISPLAY DBGWCR_EL1.MASK is non-zero and BAS is not 0xff.

ArchMsg.Warning.warning_wcr_mask_reserved

TAGS UNPREDICTABLE DISPLAY DBGWCR_ELn.MASK is set to a reserved value.

ArchMsg.Warning.warning_wcr_non_configuous_bas

TAGS UNPREDICTABLE DISPLAY DBGWCR_EL1.BAS has non-contiguous set of ones.

ArchMsg.Warning.warning_wvr_masked_bit_not_zero

TAGS UNPREDICTABLE DISPLAY DBGWVRn_EL1.VA contains a bit masked by DBGWCRn_EL1.MASK that is not zero.

ArchMsg.Why.why_branch_target_exception

A branch target exception has occurred. DISPLAY branch target exception: target pc=%{PC}, target opcode=%{OPCODE}. PRIMARY KEY PC. Fields:

BTYPE unsigned int
pstate.BTYPE at time branch taken.

OPCODE unsigned int
opcode at branch target.

PC unsigned int
Virtual Address of branch target.

ArchMsg.Why.why_illegal_state

An illegal mode change has occurred. DISPLAY illegal mode change: %{MESSAGE}. Fields:

MESSAGE string
The message.

BRANCH_MISPREDICT

Simulating branch mispredict. Fields:

PC unsigned int
Origin address (or 0 if unavailable).

BRA_DIR

Direct branches, to immediate address. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISET enum

The instruction set of the branch instruction.

IS_COND bool

Indicates if this is a conditional branch.

IS_HINTED bool

Whether the conditional instruction is hinted.

IS_ISB bool

Whether the instruction is ISB.

IS_LINK bool

Indicates if this is a branch with link.

PC unsigned int

The address of the branch instruction.

TARGET_ISET enum

The instruction set after the branch.

TARGET_PC unsigned int

The address the instruction branches to.

BRA_INDIR

Indirect branches, perhaps to a register. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISET enum

The instruction set of the branch instruction.

IS_COND bool

Indicates if this is a conditional branch.

IS_HINTED bool

Whether the conditional instruction is hinted.

IS_ISB bool

Whether the instruction is ISB.

IS_LINK bool

Indicates if this is a branch with link.

PC unsigned int

The address of the branch instruction.

TARGET_ISET enum

The instruction set after the branch.

TARGET_PC unsigned int

The address the instruction branches to.

BRA_INDIR_CCFAIL

Indirect branches fail, perhaps to a register.

BRA_RET_CCFAIL

Branch return fail, perhaps to a register.

CACHE_MAINTENANCE_OP

Cache Maintenance Operation. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

DATA unsigned int

Specified MVA or Set/way.

FUNCTION enum

Clean / Invalidate.

SCOPE enum

Affected region.

SIDE enum

Inst / Data.

CCFAIL

Conditional instruction condition check fail. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CCFAIL_UNC

Unconditional instruction condition check fail. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CCPASS

Conditional instruction condition check pass. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CCPASS_UNC

Unconditional instruction condition check pass. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CODE_CACHE_MAINT

Code cache maintenance. Fields:

END_INCL unsigned int

Inclusive end address for VA or PA requests.

NS bool

Non-secure world for PA.

START unsigned int

Start address for VA or PA requests.

TYPE enum

Request type.

COMPILE_BLOCK_END

Last instruction of basic block translated. Fields:

VADDR unsigned int

Address of next instruction after this basic block.

COMPILE_BLOCK_START

First instruction of basic block translated. Fields:

VADDR unsigned int

Address of first instruction in basic block.

COMPILE_INST

ARM instruction compiled. Fields:

DISASS string

Disassembly of the instruction.

ISSET enum

The instruction set of this instruction.

ITSTATE unsigned int

The ITSTATE current for the instruction.

OPCODE unsigned int

The opcode of the instruction.

PC unsigned int

The address of the instruction.

SIZE unsigned int

The size of the instruction in bytes.

CONTEXTIDR

Context ID Register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

NS bool

Secure or nonsecure banked register is accessed.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The new value written.

CONTEXT_SYNC

Called for every context synchronization event. Fields:

ARCH_CSE bool

Architectural Context Synchronization Event.

PC unsigned int

The address of the synchronization instruction.

SYSREG_SYNC bool

System register synchronization event.

CORE_ENDIAN

Core BE8 Big-Endian state changed. Fields:

BE8 bool

Core BE8 Big-Endian state.

CORE_INFO

Static processor attributes. Only triggered by a call to DumpState(). Fields:

ARCH_PROFILE enum

The architecture profile of the core.

CLUSTER_ID unsigned int

The cluster ID of this processor.

CORE_NUM unsigned int

The number of this core in an MP processor.

DETAILED_CORE_FEATURES enum

Detailed core features as a bitfield. Enum values are bitmasks to extract the info from the field.

FPU_VERSION enum

The VFP version implemented by the core.

MEM_ARCH enum

The memory architecture of the core.

MIDR_EL1 unsigned int

MIDR_EL1 value.

NUM_CORES unsigned int

The number of cores in this MP processor.

QUANTUM_SIZE unsigned int

The default quantum size of the core (in ticks).

SECURITY_FEATURES bool

Does the core have security features?.

CORE_LOADS

Processor load accesses. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ACQREL enum

Is this an acquire/release.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DATA unsigned int

The data read or written.

ELEMENT_SIZE unsigned int

Width of each element.

ENDIANNESS enum

Byte ordering of this access.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

LOCK enum

Normal, exclusive or locked access.

MECID unsigned int

Memory Encryption Context ID, if applicable.

MECID2 unsigned int

Memory Encryption Context ID of the second page, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PADDR_TAG unsigned int

If tag checked, the physical address tag for the access.

PAGE_SIZE unsigned intPage size as $\log_2(\text{size in bytes})$.**PAS enum**

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK, 2=Exclusive Failed, 3=Unsupported.

SIZE unsigned int

Width of the access in bytes. Only required if DATA is not traced.

TAG_CHECKED bool

Is this access tag checked.

TAG_CHECKED2 bool

Is the second page access tag checked.

TRANS bool

Is this a translated access.

VADDR unsigned int

The virtual address of the access.

CORE_REGS

Changes of the core registers R0 to R14. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number, 0 to 14.

MODE enum

Bank of the register accessed.

OLD_VALUE unsigned int

The old value overwritten.

PHYS_ID enum

The physical register accessed.

VALUE unsigned int

The new value written to the register.

CORE_REGS64

Changes of the core registers X0..X30, SP_ELn. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ID enum

The register number, 0-30 for X0-X30, >=32 for SP_ELn.

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

CORE_REGS64_READ

General purpose AArch64 register has been read X0..X30, SP_ELn. Fields:

ID enum

The register number, 0-30 for X0-X30, >=32 for SP_ELn.

VALUE unsigned int

The value read from the register.

CORE_STALLED_TRANSITION

Detect when a core is prevented from progressing PChannel states. Fields:

CURRENT_STATE enum

Current power mode of core.

REASON string

The reason a transition was stalled, and what must be done to unblock it.

TARGET_STATE enum

Desired power mode of core after transition.

CORE_STORES

Processor store accesses. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ACQREL enum

Is this an acquire/release.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DATA unsigned int

The data read or written.

ELEMENT_SIZE unsigned int

Width of each element.

ENDIANNESS enum

Byte ordering of this access.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

LOCK enum

Normal, exclusive or locked access.

MECID unsigned int

Memory Encryption Context ID, if applicable.

MECID2 unsigned int

Memory Encryption Context ID of the second page, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PADDR_TAG unsigned int

If tag checked, the physical address tag for the access.

PAGE_SIZE unsigned int

Page size as $\log_2(\text{size in bytes})$.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK, 2=Exclusive Failed, 3=Unsupported.

SIZE unsigned int

Width of the access in bytes. Only required if DATA is not traced.

TAG_CHECKED bool

Is this access tag checked.

TAG_CHECKED2 bool

Is the second page access tag checked.

TRANS bool

Is this a translated access.

VADDR unsigned int

The virtual address of the access.

CP14_READ

System Coprocessor register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP14 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc1 unsigned int

opcode 1.

opc2 unsigned int

opcode 2.

CP14_WRITE

System Coprocessor register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP14 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc1 unsigned int

opcode 1.

opc2 unsigned int

opcode 2.

CP15_READ

System Coprocessor register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP15 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc1 unsigned int

opcode 1.

opc2 unsigned int

opcode 2.

CP15_READ64

System Coprocessor register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP15 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc unsigned int

opcode 1.

CP15_WRITE

System Control Coprocessor register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP15 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc1 unsigned int

opcode 1.

opc2 unsigned int

opcode 2.

CP15_WRITE64

System Coprocessor register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP15 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc unsigned int

opcode 1.

CPSR

CPSR change. (In AArch64 PSTATE represented in SPSR format). Fields:

CORE_NUM unsigned int

Core number in a multi processor.

OLD_VALUE unsigned int

The old CPSR value.

UNKNOWN unsigned int

Bits within the register that have unknown value.

VALUE unsigned int

The new CPSR value.

CRYPTO_SPEC

Every crypto instruction speculatively executed.

CorePowerStateChange

Core power-up state has changed. DISPLAY Core powered %{PU:(up|down)}. Fields:

PU bool

Whether the core powered up.

DATA_CACHE_SET_ALLOC_TAGS

Set allocation tags from DC GVA instruction. Fields:

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

NSDESC unsigned int

The security state of the access.

PADDR unsigned int

The physical address of the access.

PAS enum

The physical address space of the page.

RESPONSE enum

0=Aborted, 1=OK.

SIZE unsigned int

The size of memory in bytes.

TAG unsigned int

The value of the tag being set.

TAGGING_TYPE enum

Type of Memory tagging (Physical or Virtual).

TAG_VADDR unsigned int

The virtual address of the Virtual Tag if Virtual tagging is used; otherwise 0.

VADDR unsigned int

Virtual address used for setting allocation tags.

DATA_CACHE_ZERO

Zero and invalidate cache line from DC ZVA instruction. Fields:

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

NSDESC unsigned int

The security state of the access.

PADDR unsigned int

The physical address of the access.

PADDR_TAG unsigned int

If tag checked, the physical address tag for the access.

PAS enum

The physical address space of the page.

RESPONSE enum

0=Aborted, 1=OK.

SIZE unsigned int

The size of zero'd memory in bytes.

TAG_CHECKED bool

Is this access tag checked.

VADDR unsigned int

Virtual address of the zero'd cache line.

DEBUG_EVENT

Hardware debug support event. Fields:

EVENT enum

Description of event.

VALUE unsigned int

data value.

DEBUG_TARGET_ISET

Hardware debug target ISA. Fields:

TARGET_ISET enum

The target instruction set of the debug state.

DMI_ALLOCATE

DMI allocated. Fields:

ACCESS_DIRECTION enum

READ / WRITE.

PADDR unsigned int

The physical address of the DMI.

PTR unsigned int

DMI range pointer.

SIDE enum

DSIDE / ISIDE.

SIZE unsigned int

The size of DMI.

VADDR unsigned int

The virtual address of the DMI..

DMI_HIT

DMI hit, access used DMI (only fire for small DMI regions i.e. less than a full page). Fields:

ACCESS_DIRECTION enum

READ / WRITE.

PADDR unsigned int

The physical address of the DMI.

SIDE enum

DSIDE / ISIDE.

SIZE unsigned int

The size of the DMI.

VADDR unsigned int

The virtual address of the DMI..

DMI_REVOKE

DMI revoked. Fields:

ACCESS_DIRECTION enum

READ / WRITE.

PADDR unsigned int

The physical address of the DMI.

SIDE enum

DSIDE / ISIDE.

SIZE unsigned int

The size of DMI.

VADDR unsigned int

The virtual address of the DMI..

END_COMPILE

Compilation end. Fields:

END_OF_PAGE_COUNT unsigned int

Number of basic blocks exited due to page boundary since START_COMPILE.

FETCHFAIL_COUNT unsigned int

Number of basic blocks exited due to ifetch failure START_COMPILE.

INST_COUNT unsigned int

Number of instructions compiled since START_COMPILE.

NONSEQ_COUNT unsigned int

Number of basic blocks exited due to non-sequential instructions since START_COMPILE.

PAGE_STRADDLE_COUNT unsigned int

Number of basic blocks exited due to unaligned instructions crossing page since START_COMPILE.

ESB

Error Synchronization Barrier hit. Fields:

ASYNC bool

ESB is async.

IMPLICIT bool

ESB is implicit.

EXCEPTION

Exception taken. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ESR unsigned int

The value assigned to ESR when exception is taken.

ESR_EC unsigned int

The Value assigned to ESR.EC when exception is taken.

IS_PHYSICAL bool

Physical or Virtual exception.

IS_TRAP bool

Exception coming from a trap.

LR unsigned int

The value assigned to the link register.

PC unsigned int

The location where the exception occurred.

PREFERRED_RETURN unsigned int

The preferred return address for the exception.

TAKEN_LOCALLY bool

Whether exception is taken locally(same exception level).

TARGET_ISET enum

The instruction set of the exception handler code.

TARGET_PC unsigned int

The address the exception branches to.

VECTOR enum

The exception vector.

VECTOR_OFFSET unsigned int

The offset of an exception vector from the base address.

EXCEPTION_END

Every exception completed.

EXCEPTION_INFO

Trace human-readable reason of the current exception for data/instruction aborts, floating point exceptions and unknown exceptions. Fields:

DESCRIPTION string

Reason for current exception.

ESR_EC unsigned int

The value of ESR.EC when exception is taken.

EXCEPTION_RAISE

Every exception raised.

EXCEPTION_RETURN

Branches on leaving exception. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISET enum

The instruction set of the branch instruction.

IS_COND bool

Indicates if this is a conditional branch.

IS_HINTED bool

Whether the conditional instruction is hinted.

IS_ISB bool

Whether the instruction is ISB.

IS_LINK bool

Indicates if this is a branch with link.

PC unsigned int

The address of the branch instruction.

TARGET_ISET enum

The instruction set after the branch.

TARGET_PC unsigned int

The address the instruction branches to.

EXCEPTION_RETURN_PREBRANCH

Exception return event that must occur before PC and processor state have updated.

EXCEPTION_START

Every exception started.

FIQ_TAKEN

FIQ taken exception.

FP_EXCEPTIONS_SIGNALLED

Trace which FP exceptions were signalled as a part of the FP operations performed by an instruction. Fields:

FP_EXCEPTIONS_SIGNALLED_MASK unsigned int

Indicates which FP exceptions were signalled by an instruction. This uses the same encoding as the cumulative floating-point exception bitfields in FPSR.

FP_STATE

Floating point state. Fields:

FPSR unsigned int

FPSR register value.

MASK unsigned int

The mask indicating updated cumulative bits.

FREQ_CHANGED

Core frequency changed. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The instruction count of this core.

NEW_FREQ signed int

The new frequency of this core (expressed in Hz).

OLD_FREQ signed int

The old frequency of this core (expressed in Hz).

HLT

HLT instruction occurred. Fields:

IMM16 unsigned int

The 16-bit immediate value encoded in HLT instruction.

INST

Every instruction executed. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DEBUG_STATE bool

The instruction is executed in debug state.

DISASS string

Disassembly of instruction.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISSET enum

The current instruction set.

ITSTATE unsigned int

The current ITSTATE.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

MODE enum

The mode the core is in.

NS unsigned int

The core's non-secure bit.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

OPCODE unsigned int

The opcode of the instruction.

PADDR unsigned int

The physical address of the instruction.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the instruction.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the page.

PC unsigned int

The address of the instruction.

SECURITY_STATE enum

The core's security state.

SIZE unsigned int

The size of the instruction in bytes.

INST_END

Every instruction completed.

INST_START

Every instruction started. Fields:

DEBUG_STATE bool

The instruction is executed in debug state.

ISSET enum

The current instruction set.

MODE enum

The mode the core is in.

NS enum

The current Secure State.

PC unsigned int

The address of the conditional instruction.

SECURITY_STATE enum

The current Security State.

INST_STRADDLE

Instruction straddles boundary. Fields:

DEBUG_STATE bool

The instruction is executed in debug state.

ISSET enum

The current instruction set.

MODE enum

The mode the core is in.

NS enum

The current Secure State.

PC unsigned int

The address of the conditional instruction.

SECURITY_STATE enum

The current Security State.

IRQ_TAKEN

IRQ taken exception.

ISB

ISB instruction occurred. Fields:

PC unsigned int

The address of the ISB instruction.

LOAD_MULTIPLE_REGS_LIST

List of destination registers (not source) of a multiple registers load instruction. Fields:

REG string

Register to add into the list.

LOCAL_MONITOR

Local monitor activity. Fields:

PADDR unsigned int

Local Monitor Address.

State enum

State of the monitor (Open/Exclusive).

MAPPING_CREATE

A memory mapping has been created. This is an implementation detail of the model and so may change without notice. Fields:

ASID unsigned int

The ASID used during translation, or 0 if no ASID was used.

IS_VALID bool

Whether the mapping is valid or not.

PADDR unsigned int

Base physical address of region, or 0 if not known.

SIDE enum

I-side or D-side.

SIZE unsigned int

Size of the page, or 0 if not known.

VADDR unsigned int

Base virtual address of region.

VMID unsigned int

The VMID used during translation, or 0 if no VMID was used.

MAPPING_DESTROY

A memory mapping has been destroyed. This is an implementation detail of the model and so may change without notice. Fields:

ASID unsigned int

The ASID used during translation.

CAUSE enum

The cause of invalidation.

PADDR unsigned int

Base physical address of page.

SIDE enum

I-side or D-side.

SIZE unsigned int

Size of the page.

VADDR unsigned int

Base virtual address of page.

VMID unsigned int

The VMID used during translation.

MEMMAP_DEBUG_READ

Memory mapped reads to the debug registers. Fields:

ADDR unsigned int

address.

CORE_NUM unsigned int

Core number in a multi processor.

EXT bool

Whether access is from an external device (such as the DAP).

NS enum

Secure state of the access.

REG_NAME string

Name of the debug register accessed.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

MEMMAP_DEBUG_WRITE

Memory mapped writes to the debug registers. Fields:

ADDR **unsigned int**

address.

CORE_NUM **unsigned int**

Core number in a multi processor.

EXT **bool**

Whether access is from an external device (such as the DAP).

NS **enum**

Secure state of the access.

REG_NAME **string**

Name of the debug register accessed.

UNDEF **bool**

The register accessed is undefined.

VALUE **unsigned int**

The new value written.

MEMTAG_LOADS

Allocation tag memory read; when attributes and system configuration allow tag accesses.

Fields:

ACCESS_TYPE **enum**

The type of instruction performing the access.

ATTR **unsigned int**

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 **unsigned int**

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

NSDESC **unsigned int**

The physical address non-secure bit.

NSDESC2 **unsigned int**

The second page physical address non-secure bit.

NUM_GRANULES **unsigned int**

Number of tags to be Loaded/stored.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK.

TAGGING_TYPE enum

Type of Memory tagging (Physical or Virtual).

TAGS unsigned int

The tags read or written. Tags for subsequent granules would be found at subsequent bytes in this field (One byte for each tag).

TAG_VADDR unsigned int

The virtual address of the Virtual Tag if Virtual tagging is used; otherwise 0.

VADDR unsigned int

The virtual address of the access.

MENTAG_LOAD_INST

Allocation tag load instruction; regardless of tags being actually accessed. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

NUM_GRANULES unsigned int

Number of tags to be Loaded/stored.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK.

TAGGING_TYPE enum

Type of Memory tagging (Physical or Virtual).

TAGS unsigned int

The tags read or written. Tags for subsequent granules would be found at subsequent bytes in this field (One byte for each tag).

TAG_VADDR unsigned int

The virtual address of the Virtual Tag if Virtual tagging is used; otherwise 0.

VADDR unsigned int

The virtual address of the access.

MEMTAG_STORES

Allocation tag memory write; when attributes and system configuration allow tag accesses.

Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

NUM_GRANULES unsigned int

Number of tags to be Loaded/stored.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK.

TAGGING_TYPE enum

Type of Memory tagging (Physical or Virtual).

TAGS unsigned int

The tags read or written. Tags for subsequent granules would be found at subsequent bytes in this field (One byte for each tag).

TAG_VADDR unsigned int

The virtual address of the Virtual Tag if Virtual tagging is used; otherwise 0.

VADDR unsigned int

The virtual address of the access.

MEMTAG_STORE_INST

Allocation tag store instruction; regardless of tags being actually accessed. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

NUM_GRANULES unsigned int

Number of tags to be Loaded/stored.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK.

TAGGING_TYPE enum

Type of Memory tagging (Physical or Virtual).

TAGS unsigned int

The tags read or written. Tags for subsequent granules would be found at subsequent bytes in this field (One byte for each tag).

TAG_VADDR unsigned int

The virtual address of the Virtual Tag if Virtual tagging is used; otherwise 0.

VADDR unsigned int

The virtual address of the access.

MMU_D128_TTB_READ

This event is triggered by reads caused by a D128 translation table walk. Fields:

ABORTED bool

Was the walk successful.

DATA_HIGH unsigned int

The data upper bits read.

DATA_LOW unsigned int

The data lower bits read.

IPA unsigned int

For stage 1, the IPA of the read.

LEVEL unsigned int

Translation table level.

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

OUTADDR unsigned int

The output physical address.

PADDR unsigned int

The physical address of the read.

PASmem enum

Physical Address Space of the accessed memory.

PASreq enum

Physical Address Space of the requested lookup.

REGIME_EL enum

Entry matches in this translation regime.

REQUESTER enum

The requester of this table walk.

SIDE enum

Inst / Data.

STAGE unsigned int

Translation stage.

MMU_D128_TTB_WRITE

This event is triggered by writes caused by a D128 translation table walk. Fields:

ABORTED bool

Was the walk successful.

DATA_HIGH unsigned int

The data upper bits written.

DATA_LOW unsigned int

The data lower bits written.

IPA unsigned int

For stage 1, the IPA of the write.

IS_HACDBS_ACCESS bool

If the access was due to HACDBS_CLEAN.

LEVEL unsigned int

Translation table level.

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

OUTADDR unsigned int

The output physical address.

PADDR unsigned int

The physical address of the write.

PASmem enum

Physical Address Space of the accessed memory.

PASreq enum

Physical Address Space of the requested lookup.

REGIME_EL enum

Entry matches in this translation regime.

SIDE enum

Inst / Data.

STAGE unsigned int

Translation stage.

MMU_TRANS

Address translation information. Fields:

ASID unsigned int

Address space identifier.

CORE_NUM unsigned int

Core number in a multi processor.

Hyp bool

Entry matches in Hyp state only.

INNERCACHE_RA bool

Is the inner cache allocate on read.

INNERCACHE_TRANSIENT bool

Is the inner write-through transient.

INNERCACHE_TYPE enum

Inner Caching scheme (NC/MB/WA).

INNERCACHE_WA bool

Is the inner cache allocate on write.

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MEMTYPE enum

Memory type.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC enum

Is secure side supposed to access secure or nonsecure memory.

OUTERCACHE_RA bool

Is the outer cache allocate on read.

OUTERCACHE_TRANSIENT bool

Is the outer write-through transient.

OUTERCACHE_TYPE enum

Outer Caching scheme (NC/MB/WA).

OUTERCACHE_WA bool

Is the outer cache allocate on write.

PADDR unsigned int

Physical address of the access.

PAGESIZE unsigned int

Page size as $\log_2(\text{size})$.

PAS enum

PAS supposed to be accessed is secure or nonsecure memory or Root/Realm if RME is implemented.

SH enum

Shareability.

SIDE enum

Inst / Data.

STAGE1_PERM unsigned int

Stage 1 permissions, [5:3] Privileged access (XWR) [2:0] User access (XWR).

STAGE2_PERM unsigned int

Stage 2 permission mask [5:3] Privileged access (XWR) [2:0] User access (XWR).

VADDR unsigned int

Virtual address of the access.

VMID unsigned int

Virtual machine identifier.

XS bool

XS Attribute.

nG enum

Flag indicating whether ASID will be matched.

MMU_TTB_READ

This event is triggered by reads caused by a translation table walk. Fields:

ABORTED bool

Was the walk successful.

DATA unsigned int

The data read.

DESC_KIND enum

The encoding scheme of the descriptors used for this walk.

IPA unsigned int

For stage 1, the IPA of the read.

LEVEL unsigned int

Translation table level.

LPAE bool

Is this for an LPAE translation (*DEPRECATED*, use DESC_KIND).

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSmem bool

Non secure memory access.

NSreq bool

Non secure request.

OUTADDR unsigned int

The output physical address.

PADDR unsigned int

The physical address of the read.

PASmem enum

Physical Address Space of the accessed memory.

PASreq enum

Physical Address Space of the requested lookup.

REGIME_EL enum

Entry matches in this translation regime.

REQUESTER enum

The requester of this table walk.

SIDE enum

Inst / Data.

STAGE unsigned int

Translation stage.

MMU_TTB_WRITE

This event is triggered by writes caused by a translation table walk. Fields:

ABORTED bool

Was the walk successful.

DATA unsigned int

The data written.

DESC_KIND enum

The encoding scheme of the descriptors used for this walk.

IPA unsigned int

For stage 1, the IPA of the write.

IS_HACDBS_ACCESS bool

If the access was due to HACDBS_CLEAN.

LEVEL unsigned int

Translation table level.

LPAAE boolIs this for an LPAAE translation (*DEPRECATED*, use DESC_KIND).**MECID unsigned int**

Memory Encryption Context(MEC) ID, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSmem bool

Non secure memory access.

NSreq bool

Non secure request.

OUTADDR unsigned int

The output physical address.

PADDR unsigned int

The physical address of the write.

PASmem enum

Physical Address Space of the accessed memory.

PASreq enum

Physical Address Space of the requested lookup.

REGIME_EL enum

Entry matches in this translation regime.

SIDE enum

Inst / Data.

STAGE unsigned int

Translation stage.

MODE_CHANGE

Mode change. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

MODE enum

The new mode.

NON_SECURE enum

The core's new non-secure bit.

OLD_MODE enum

The old mode.

SECURITY_STATE enum

The new security state.

NOT_TAG_CHECKED_INST

MTE Tag not checked for current instruction.

PERIODIC

Called for every quantum. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The instruction count of this CPU.

PC unsigned int

The address of the next instruction to be executed on this CPU.

PMU_COUNTER_OVERFLOW

PMU counter overflow. Fields:

COUNTER_GROUP enum

Counter group.

COUNTER_RANGE enum

Counter range.

EVENT_ID unsigned int

Event Identifier, field is valid only for Event counter group.

INDEX unsigned int

Counter index (as selected by PMSELR), field is valid only for Event/Cycle counter groups.

INTERRUPT bool

Is interrupt enabled on overflow for this counter.

POINTER_AUTH

Called when execution hits a pointer authentication check. Fields:

AUTH_STATE bool

Authentication result.

KEY_USED enum

Pointer authentication key.

POTINDEX_UPDATE

Value of POT index for current pc changed. Fields:

NEW_POTINDEX unsigned int

New POTIndex.

OLD_POTINDEX unsigned int

Old POTIndex.

PREFETCH_MEMORY64

Prefetch from PRFM or PRFUM instructions. Fields:

PRFOP enum

Prefetch hint.

VADDR unsigned int

Virtual address of the location that should be prefetched.

PRELOAD64_REQUEST

Preload requests from PRFM or PRFUM instructions. Fields:

PRFOP enum

Prefetch hint.

VADDR unsigned int

Virtual address of the location that should be prefetched.

PRELOAD_DATA

Data preload from PLD instruction. Fields:

VADDR unsigned int

Virtual address of the data that should be preloaded.

PRELOAD_INST

Instruction preload from PLI instruction. Fields:

VADDR unsigned int

Virtual address of the instruction that should be preloaded.

PRE_CORE_LOAD

Trace just before a core load. Fields:

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

PRE_CORE_STORE

Trace just before a core store. Fields:

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

PRE_TTB_READ

This event is triggered before reads caused by a translation table walk. Fields:

ADDR unsigned int

The physical address of the read.

PSTATE

PSTATE fields update. Fields:

ALLINT unsigned int

All interrupt mask bit.

BTYPE unsigned int

Branch target identification bit.

CORE_NUM unsigned int

Core number in a multi processor.

DAIF unsigned int

Exception mask bits.

DIT unsigned int

Data Independent Timing bit.

EL unsigned int

Current Exception level field.

EXLOCK unsigned int

Exception return state lock bit.

IL unsigned int

Illegal Execution state bit.

NZCV unsigned int

Condition flags.

PACM unsigned int

PACM bit.

PAN unsigned int

Privileged Access Never state bit.

PM unsigned int

PMU exception mask bit.

PPEND unsigned int

PMU exception pending bit.

SM unsigned int

StreamingMode status bit.

SP unsigned int

Stack pointer register selection bit.

SS unsigned int

Software Step bit.

SSBS unsigned int

Speculative Store Bypass Safe bit.

TCO unsigned int

Tag Check Override bit.

TOUCHED unsigned int

Mask of the fields that have been touched since this trace has last been emitted. Note that touched does not necessarily mean the value changed. The index of the field is the bit index in the mask plus the index of this field plus one.

UAO unsigned int

User Access Override bit.

UINJ unsigned int

Undefined Instruction exceptions bit.

ZA unsigned int

ZA storage and ZT0 register enable bit.

nRW unsigned int

Current Execution state bit.

PSTATE_PACTIVE_OPMODE_TRANSITION

Trace pactive and Opmode mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Operating mode after the transition.

OLD_STATE enum

Operating mode before the transition.

PSTATE_PACTIVE_OPMODE_TRANSITION_DENIED

Trace the denied pactive and Opmode mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Operating mode after the transition.

OLD_STATE enum

Operating mode before the transition.

REASON string

Reason for the transition being denied.

PSTATE_PACTIVE_POWER_TRANSITION

Trace pactive and power mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Power mode after the transition.

OLD_STATE enum

Power mode before the transition.

PSTATE_PACTIVE_POWER_TRANSITION_DENIED

Trace the denied pactive and power mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Power mode after the transition.

OLD_STATE enum

Power mode before the transition.

REASON string

Reason for the transition being denied.

PSTATE_PACTIVE_THREAD_OPMODE_TRANSITION

Trace pactive and Opmode mode transitions for threads on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Operating mode after the transition.

OLD_STATE enum

Operating mode before the transition.

QUANTUM_START

Called at the start of every quantum.

RANGE_PRELOAD_REQUEST

Preload requests from RPRFM. Fields:

COUNT unsigned int

Number of blocks of data to access.

LENGTH unsigned int

Number of contiguous bytes to be accessed in each block.

OP unsigned int

Range prefetch operation .

REUSE unsigned int

Maximum number of bytes to be accessed before next RPRFM .

STRIDE signed int

Stride to advance after accessing LENGTH bytes.

VADDR unsigned int

Base virtual address of range prefetch request.

RUN_STATE

Run state transition. Fields:

INST_COUNT **unsigned int**

Ticks count at point of transition.

NEW **enum**

New run state.

OLD **enum**

Old run state.

SEMIHOSTING_CALL

Call of a semihost function occurred. Fields:

PC **unsigned int**

The program counter after the semihosting call.

SEMIHOSTING_PRECALL

About to call semihost. Fields:

EL2 **bool**

Use EL2 translation regime.

NS **bool**

Is Non-Secure.

REG_WIDTH **unsigned int**

The current register width in bytes.

SIGNAL

External signal state change. Fields:

SIGNAL **enum**

Signal that changed.

STATE **bool**

Signal asserted state.

SOFTWARE_STEP

Return the Debug Software Step state. Fields:

STATE **enum**

Software Step state.

SPSR

SPSR change. Fields:

CORE_NUM **unsigned int**

Core number in a multi processor.

MODE **enum**

Which of the banked SPSR registers is written.

OLD_VALUE unsigned int

The old SPSR value.

VALUE unsigned int

The new SPSR value.

START_COMPILE

Compilation started. Fields:

VADDR unsigned int

Instruction where compilation begins.

STG_MULTIPLE_MICRO_OPS_ADDRESS_RANGE

Address range for MTE stores with multiple uops. Fields:

ADDR_END unsigned int

Final address.

ADDR_START unsigned int

Starting address.

STORE_MULTIPLE_REGS_ADDRESS_RANGE

Address range for stores with multiple registers. Fields:

ADDR_END unsigned int

Final address.

ADDR_START unsigned int

Starting address.

SUBOPTIMAL_LDST_RETIRED

suboptimal load/store. Fields:

ACCESS_SIZE unsigned int

Log2 of the access width in bytes used for alignment checking (1: half-word, 2: word ...).

OPTIMAL_ALIGN_SELECTOR unsigned int

Selector to decide the optimal alignment size (1: access width, 2: 4-bytes, 3: 8-bytes, ..., 12: 4-kbytes).

VADDR unsigned int

The virtual address of the access.

SVE_INST_SPEC

SVE operations speculatively executed.

SVE_LD_RETIRED

SVE load instructions architecturally executed.

SVE_LOADS

SVE load accesses. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ACQREL enum

Is this an acquire/release.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DATA unsigned int

The data read or written.

ELEMENT_SIZE unsigned int

Width of each element.

ENDIANNESS enum

Byte ordering of this access.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

LOCK enum

Normal, exclusive or locked access.

MECID unsigned int

Memory Encryption Context ID, if applicable.

MECID2 unsigned int

Memory Encryption Context ID of the second page, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PADDR_TAG unsigned int

If tag checked, the physical address tag for the access.

PAGE_SIZE unsigned int

Page size as $\log_2(\text{size in bytes})$.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK, 2=Exclusive Failed, 3=Unsupported.

SIZE unsigned int

Width of the access in bytes. Only required if DATA is not traced.

TAG_CHECKED bool

Is this access tag checked.

TAG_CHECKED2 bool

Is the second page access tag checked.

TRANS bool

Is this a translated access.

VADDR unsigned int

The virtual address of the access.

SVE_MEM_PREFETCH

Scalable Vector Extension memory prefetch. Fields:

CACHE_LEVEL enum

The preferred cache level.

HINT enum

What the memory is expected to be used for.

SIZE unsigned int

The prefetch size in bytes.

TEMPORAL bool

Whether the data is temporal.

VADDR unsigned int

The virtual address of the prefetch.

SVE_REG_READ

Scalable Vector Extension register read. Fields:

ID enum

The register identifier.

SM bool

Whether the PE is in Streaming Mode.

VALUE unsigned int

The read data.

SVE_REG_UPDATE

Scalable Vector Extension register update. Fields:

ID enum

The register identifier.

SM bool

Whether the PE is in Streaming Mode.

VALUE unsigned int

The written data.

SVE_REG_WRITE

Scalable Vector Extension register write. Fields:

ID enum

The register identifier.

SM bool

Whether the PE is in Streaming Mode.

VALUE unsigned int

The written data.

SVE_STORES

SVE store accesses. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ACQREL enum

Is this an acquire/release.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DATA unsigned int

The data read or written.

ELEMENT_SIZE unsigned int

Width of each element.

ENDIANNESS enum

Byte ordering of this access.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

LOCK enum

Normal, exclusive or locked access.

MECID unsigned int

Memory Encryption Context ID, if applicable.

MECID2 unsigned int

Memory Encryption Context ID of the second page, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PADDR_TAG unsigned int

If tag checked, the physical address tag for the access.

PAGE_SIZE unsigned int

Page size as $\log_2(\text{size in bytes})$.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK, 2=Exclusive Failed, 3=Unsupported.

SIZE unsigned int

Width of the access in bytes. Only required if DATA is not traced.

TAG_CHECKED bool

Is this access tag checked.

TAG_CHECKED2 bool

Is the second page access tag checked.

TRANS bool

Is this a translated access.

VADDR unsigned int

The virtual address of the access.

SVE_ST_RETIRED

SVE store instructions architecturally executed.

SVE_UNALIGNED_LDST_RETIRED

SVE load or store instructions architecturally executed that would cause alignment fault if A bit is set.

SVE_Z_PRED_MEM_ACCESS_INFO

Information about SVE/SME predicated accesses to Z registers (non Gather/Scatter). Fields:

MEM_ELEM_SIZE unsigned int

Width of elements accessed in memory (in bytes).

RnW bool

Whether this is a Read not a Write.

VA unsigned int

Virtual base address computed for accesses.

SYNC

Called for every synchronization, that is every quantum break. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The instruction count of this CPU.

LOCAL_QUANTUM unsigned int

The local quantum of this CPU.

LOCAL_TIME unsigned int

The local time of this CPU.

SYNC_REASON enum

Reason for sync.

SYSCALL

System call instruction executed. Fields:

IMM unsigned int

Immediate value of the system call instruction.

TYPE enum

System call type.

VADDR unsigned int

Instruction that caused the system call.

SYSREG_READ128

System Coprocessor 128 bit register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

REGNUM enum

Internal register number.

REGNUM_PHYS enum

Internal register number physically accessed (e.g. in presense of VHE redirection).

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_READ64

System Coprocessor register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REGNUM enum

Internal register number.

REGNUM_OPERAND enum

Register number corresponding to the instruction operand.

REGNUM_PHYS enum

Internal register number physically accessed (e.g. in presense of VHE redirection).

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_UPDATE128

Triggers when the system updates a register. Fields:

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG enum

Register number.

UNKNOWN unsigned int

Bits of the register which became unknown.

VALUE unsigned int

Value written to the register.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_UPDATE32

Triggers when the system updates a register. Fields:

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG enum

Register number.

UNKNOWN unsigned int

Bits of the register which became unknown.

VALUE unsigned int

Value written to the register.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_UPDATE64

Triggers when the system updates a register. Fields:

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG enum

Register number.

UNKNOWN unsigned int

Bits of the register which became unknown.

VALUE unsigned int

Value written to the register.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_WRITE128

System Coprocessor 128 bit register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

REGNUM enum

Internal register number.

REGNUM_PHYS enum

Internal register number physically accessed (e.g. in presense of VHE redirection).

UNDEF bool

The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int
opcode 2.

SYSREG_WRITE64

System Coprocessor register write. Fields:

CORE_NUM unsigned int
Core number in a multi processor.

CRm unsigned int
CRm.

CRn unsigned int
CRn.

NS bool
Secure or nonsecure banked register is accessed.

REGNUM enum
Internal register number.

REGNUM_OPERAND enum
Register number corresponding to the instruction operand.

REGNUM_PHYS enum
Internal register number physically accessed (e.g. in presense of VHE redirection).

UNDEF bool
The register accessed is undefined.

UPDATED_VALUE unsigned int
Updated value of the register now it has been written.

VALUE unsigned int
The new value written.

opc unsigned int
opcode 1.

opc0 unsigned int
opcode 0.

opc2 unsigned int
opcode 2.

TRBU_BUFFER_WRAP

DISPLAY Trace buffer current write pointer wrapped.

TRBU_TRIGGER_EVENT

DISPLAY Trace buffer Trigger Event.

TRBU_WRITE

DISPLAY PETU trace data has been written to the TRBU memory buffer at
%{BUFFER_ADDRESS} with size %{SIZE} and data %{DATA}. Fields:

BUFFER_ADDRESS unsigned int

Virtual Address of start of buffer write.

DATA unsigned int

Data written to the buffer.

SIZE unsigned int

Size of sample written to buffer, in bytes.

UNALIGNED_LDST_RETIRED

Processor unaligned load/store. Fields:

ACCESS_SIZE unsigned int

Log2 of the access width in bytes used for alignment checking (1=>half-word, 2->word ...).

NUMBER_OF_BEATS unsigned int

Number of accesses (beats) in this burst (Total size in bytes is ACCESS_SIZE * NUMBER_OF_BEATS).

VADDR unsigned int

The virtual address of the access.

VFP_D_REGS

VFP/NEON D 64 bit register write. Fields:

ALIASING enum

Type of register aliasing used.

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a write to the corresponding bit in the register. For example, 0x1 would indicate a write to VALUE[0].

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

VFP_D_REGS_READ

VFP/NEON D 64 bit register read. Fields:

ID unsigned int

The register number.

VALUE unsigned int

The current value of the register.

VFP_Q_REGS

VFP/NEON Q 128 bit register write. Fields:

ALIASING enum

Type of register aliasing used.

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a write to the corresponding byte in the register. For example, 0x1 would indicate a write to VALUE[7:0].

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

VFP_Q_REGS_READ

VFP/NEON Q 128 bit register read. Fields:

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a read to the corresponding byte in the register. For example, 0x1 would indicate a read to VALUE[7:0].

VALUE unsigned int

The current value of the register.

VFP_SYS_REGS

Writes to the VFP/NEON units system registers. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ID enum

Which VFP system register is written.

MASK unsigned int

The mask indicating updated cumulative bits.

OLD_VALUE unsigned int

The register's old value overwritten.

VALUE unsigned int

The new value written to the VFP system register.

VFP_S_REGS

VFP/NEON S 32 bit register write. Fields:

ALIASING enum

Type of register aliasing used.

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a write to the corresponding bit in the register. For example, 0x1 would indicate a write to VALUE[0].

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

VFP_S_REGS_READ

VFP/NEON S 32 bit register read. Fields:

ID unsigned int

The register number.

VALUE unsigned int

The current value of the register.

VIRTUAL_TAG_ALLOCATED

Virtual tag information when allocated. Fields:

TAG_VADDR unsigned int

Virtual tag virtual address.

VADDR unsigned int

Data Virtual address.

VIRTUAL_TAG unsigned int

Virtual tag value.

WAYPOINT

Signals end of basic block execution. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ISSET enum

Origin instruction set.

IS_COND bool

Indicates if this is a conditional waypoint.

PC unsigned int

Origin address (or 0 if unavailable).

TAKEN bool

Indicates if this waypoint was taken.

TARGET unsigned int

Destination address.

TARGET_ISET enum

Destination instruction set.

WFE_END

WFE ended. Fields:

INST_COUNT unsigned int

Ticks count when leaving WFE.

WFE_EVENT_REGISTER

WFE event register status: set/clear, reason. Fields:

INST_COUNT unsigned int

Ticks count.

REASON enum

Reason for set/clear. Only REASON==0 (Cleared by WFE) clears the bit, all other reasons set it.

WFE_IGNORED

WFE ignored. Fields:

EVENT bool

This WFE was ignored because the event register was set.

INST_COUNT unsigned int

Ticks count when ignoring WFE.

TRAPPED bool

This WFE was trapped.

WFE_START

WFE entered. Fields:

INST_COUNT unsigned int

Ticks count when entering WFE.

WFI_END

WFI ended. Fields:

INST_COUNT unsigned int

Ticks count when leaving WFI.

WFI_IGNORED

WFI ignored. Fields:

DISABLED bool

This WFI was ignored because WFI is disabled.

INST_COUNT unsigned int

Ticks count when ignoring WFI.

REASON enum

specifies reason why WFI trace was ignored.

TRAPPED bool

This WFI was trapped.

WFI_START

WFI entered. Fields:

INST_COUNT unsigned int

Ticks count when entering WFI.

WFI_WAKEUP

WFI wakeup. Fields:

INST_COUNT unsigned int

Ticks count when WFI wakeup occurred.

REASON enum

Reason for wakeup.

2.25 ARM_Cortex-A715

This section describes the trace sources.

AA64_ASE_SVE_REGS

Modification of AArch64 Advanced SIMD or SVE Vector Registers (via SIMD or Scalar operations), SVE Predicate Registers and ZT0. Fields:

ID enum

The register identifier.

MASK unsigned int

Bitmask of the register to signal the modified bits in the VALUE field.

SM bool

Whether the PE is in Streaming Mode.

VALUE unsigned int

Value of AArch64 vector register.

ASYNC_MEMORY_FAULT

Context ID Register write. Fields:

FAULT unsigned int

Fault status present in ESR.ISS encoding for Serror in ARCH64 and present in DFSR encoding for external abort format in ARCH32.

PADDR unsigned int

Physical Address (or 0 if unavailable).

VADDR signed int

Virtual Address (or 0 if unavailable).

ATOMIC_END_ACCESS

Bus trace access for atomics. Fields:

ACCESS_FAIL bool

Memory access failed.

ADDR unsigned int

The virtual address of the access.

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

COMPARE_VALUE unsigned int

Compare value for CAS.

LOAD_VALUE unsigned int

Loaded values.

NSDESC unsigned int

The security state of the access.

OPERAND_VALUE unsigned int

Operation's operand.

OPERATION enum

Operation type.

PADDR unsigned int

The physical address of the access.

PRIV bool

Is this a privileged access?.

ATOMIC_START_ACCESS

Bus trace access for atomics. Fields:

ADDR unsigned int

The virtual address of the access.

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

COMPARE_VALUE unsigned int

Compare value for CAS.

NSDESC unsigned int

The security state of the access.

OPERAND_VALUE unsigned int

Operation's operand.

OPERATION enum

Operation type.

PADDR unsigned int

The physical address of the access.

PRIV bool

Is this a privileged access?.

ArchMsg.Error.exit_code

DISPLAY exit code: %{EXITCODE} component kind: %{KIND} reason: %{REASON}. Fields:

EXITCODE signed int

Exit code to be returned.

KIND string

Component kind that invoked the exit code trace.

REASON string

Reason why the exit code trace was invoked (optional).

ArchMsg.Warning.IgnoredWarmResetRequest

Event fires when software attempts to request warm reset through write to deprecated DBGPRCR.CWRR.

ArchMsg.Warning.RAS_ErrselrUnpredictableState

DISPLAY ERRSELR is in unpredictable state due to value greater than number_of_error_records written to it.

ArchMsg.Warning.RAS_UnpredictableStateWrite

DISPLAY %{REG_NAME} write ignored due to unpredictable ERRSELR state. Fields:

REG_NAME string

Name of register being written.

ArchMsg.Warning.ReportedRevisionRegisterRead

DISPLAY Reported revision and patch do not match that of the cpu. Fields:

ORIGINAL_VALUE unsigned int

The value the register would have based on the modelled revision.

REGISTER string

The name of the register that has ben read.

REPORTED_VALUE unsigned int

The value the register has been overridden to.

ArchMsg.Warning.Unpredictably Indexed PM Event Register

TAGS **UNPREDICTABLE** DISPLAY Event %{IsDirect:(Selector value|Register)} %{SEL} is >= %{N}. Fields:

IsDirect unsigned int

Direct Access rather than indirect.

N unsigned int

Number of Accessible Registers.

SEL unsigned int

Selector.

ArchMsg.Warning.cache_contents_unknown

A cache has been activated without being invalidated and may contain **UNKNOWN** content. DISPLAY %{SIDE:(D|I)}-Cache was enabled in %{REGIME:(SECURE|NON_SECURE|HYP)} regime but was not invalidated since power-on: cache lines could contain **UNKNOWN** content %{SIDE:(Note: D-side caches must invalidate SECURE lines even if the cache is only used by NON_SECURE code in order to ensure that unexpected dirty lines tagged SECURE are not naturally evicted)}. Fields:

REGIME unsigned int

Current translation regime.

SIDE unsigned int

0 d-side cache or unified cache, 1 i-side.

ArchMsg.Warning.dcimvac_matches_watchpoint

TAGS **UNPREDICTABLE** DISPLAY Watchpoints matching an AArch32 DCIMVA are implementation defined.

ArchMsg.Warning.decode_fieldmismatch

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with different fields %{FIELD1} and %{FIELD2}. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_frachitsnegative

TAGS **UNPREDICTABLE** DISPLAY VCVT Instruction is unpredictable with a negative number of fraction bits.

ArchMsg.Warning.decode_initblock

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable in an IT block.

ArchMsg.Warning.decode_initblocknotlast

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable in an IT block when not the last.

ArchMsg.Warning.decode_invalid_condition

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable when the condition is not AL.

ArchMsg.Warning.decode_invalidfieldcombination

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with the values in %{FIELD1} and %{FIELD2}. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_invalidvalue

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with the value in %{FIELD}. Fields:

FIELD enum

Field name.

ArchMsg.Warning.decode_mem_hint_unallocated

TAGS **UNPREDICTABLE** DISPLAY There is no memory hint allocated to this bit pattern.

ArchMsg.Warning.decode_registermatch

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with %{FIELD1} and %{FIELD2} as the same register. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_registermismatch

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with %{FIELD1} and %{FIELD2} as different registers. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_registeroutofrange

TAGS **UNPREDICTABLE** DISPLAY Floating-point instruction is unpredictable indexing beyond the end of the register bank.

ArchMsg.Warning.decode_sbzsbo

TAGS **UNPREDICTABLE** DISPLAY Reserved bits in the instruction are not canonical.

ArchMsg.Warning.decode_transactiontoobig

TAGS **UNPREDICTABLE** DISPLAY Attempt to perform a memory transaction of more than 128 bytes.

ArchMsg.Warning.decode_unpred_other

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable.

ArchMsg.Warning.decode_unpreduseofpc

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable when %{FIELD} is PC. Fields:

FIELD enum

Field name.

ArchMsg.Warning.decode_unpreduseofr13

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable when %{FIELD} is R13. Fields:

FIELD enum

Field name.

ArchMsg.Warning.decode_unpreduseofr13inreglist

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with R13 in register-list.

ArchMsg.Warning.decode_useofd16orhigher

TAGS **UNPREDICTABLE** DISPLAY Attempt to use register D16 or higher where it is not allowed.

ArchMsg.Warning.decode_writebackandbaseinlist

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with write back when the base register is in the transfer list.

ArchMsg.Warning.decode_zeroregistersinlist

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable when no registers are to be transferred.

ArchMsg.Warning.misaligned_ptrs

DISPLAY TRBPTR_EL1(%{TRBPTR_EL1}) or TRBTRG_EL1(%{TRBTRG_EL1}) are not aligned as per TRBIDR_EL1.Align(%{TRBIDR_EL1}). Fields:

TRBIDR_EL1 unsigned int

TRBIDR_EL1.Align value.

TRBPTR_EL1 unsigned int

Current write pointer.

TRBTRG_EL1 unsigned int

Trigger pointer.

ArchMsg.Warning.out_of_range_ptrs

DISPLAY TRBPTR_EL1(%{TRBPTR_EL1}) is out of range as per TRBBASER_EL1(%{TRBBASER_EL1}) or TRBLIMITR_EL1.LIMIT(%{TRBLIMITR_EL1}). Fields:

TRBBASER_EL1 unsigned int

Base pointer.

TRBLIMITR_EL1 unsigned int

Limit pointer.

TRBPTR_EL1 unsigned int

Current write pointer.

ArchMsg.Warning.recursive_branch

An instruction is performing a branch that targets the same instruction. If this was intended then a WFI might be more effective. DISPLAY recursive branch detected at PC=%{PC}. PRIMARY KEY PC. Fields:

PC unsigned int

Address of instruction causing the branch.

ArchMsg.Warning.recursive_exception

An instruction has generated an exception that targets the same instruction. DISPLAY recursive %{TYPE} exception detected at PC=%{PC} (CPSR=%{CPSR}). PRIMARY KEY PC. Fields:

CPSR unsigned int

Processor state.

PC unsigned int

Address of instruction causing the exception.

TYPE enum

Type of exception.

ArchMsg.Warning.reentrant_vector_catch

TAGS UNPREDICTABLE DISPLAY Vector catch debug event on Prefetch or Data abort vector %{ADDR}. Fields:

ADDR unsigned int

vector address.

ArchMsg.Warning.reserved_it_state

TAGS UNPREDICTABLE DISPLAY Execution with reserved IT state %{ITSTATE}. Fields:

ITSTATE unsigned int

reserved IT state.

ArchMsg.Warning.tlb_contents_unknown

Translation has been enabled without invalidating the TLB. DISPLAY Translation was enabled in the %{PAS:(Secure|Non-secure|Root|Realm)} EL%{EL:d} translation regime (vmid=

{VMID:x}) without invalidating {INVALIDITY:(|d-side |i-side |)} entries since power-on: {INVALIDITY:(|D||)} TLB could contain **UNKNOWN** content. Fields:

EL unsigned int

Exception level of translation regime.

INVALIDITY unsigned int

Which TLB may be invalid (if TLBs are separate).

NON_SECURE bool

Whether the translation regime is non-secure.

PAS unsigned int

Physical address space of translation regime.

VMID unsigned int

Current VMID, if applicable.

ArchMsg.Warning.unknown_DLR_DSPSR

Exiting debug state with unknown DLR or DSPSR.

ArchMsg.Warning.unknown_ELR_SPSR

Returning from debug exception with unknown ELR or SPSR.

ArchMsg.Warning.unpred_FAR_on_LRCPC3_non_atomic_mem_access

TAGS **UNPREDICTABLE** Unpredictable FAR/FSTATUS on LRCPC3 pair instructions resulting in two separate memory accesses
DISPLAY Unpredictable FAR on LRCPC3 non atomic memory access at {ADDR1} with FSTATUS {FSTATUS1} or {ADDR2} with FSTATUS {FSTATUS2}. Fields:

ADDR1 unsigned int

Possible FAR address 1.

ADDR2 unsigned int

Possible FAR address 2.

FSTATUS1 unsigned int

Possible fault status value 1.

FSTATUS2 unsigned int

Possible fault status value 2.

ArchMsg.Warning.unpred_async_tag_check_fail_on_sve_speculative_load_do_update_ffr

Unpredictable set to FALSE of the FFR element associated to SVE speculative load with an async tag check fail TAGS **UNPREDICTABLE**.

ArchMsg.Warning.unpredictable_a32_breakpoint

TAGS **UNPREDICTABLE** DISPLAY Unpredictable breakpoint on a A32 instruction at {ADDR1} Offset {OFFSET} BAS {BAS} for a {IS_ADDRESS_MISMATCH_BREAKPOINT:(Mismatch|Match)} Breakpoint number {N:d} Breakpoint address {ADDR2}. Fields:

ADDR1 unsigned int

Address.

ADDR2 unsigned int

Breakpoint address.

BAS unsigned int

BAS field.

IS_ADDRESS_MISMATCH_BREAKPOINT bool

Is a Mismatch breakpoint, not a Match breakpoint.

N unsigned int

Breakpoint number.

OFFSET unsigned int

Always has a value of zero.

ArchMsg.Warning.unpredictable_banked_register_access

TAGS **UNPREDICTABLE** DISPLAY Access to banked register (R=%{R:d} SYSm=%{SYSM:2x}) is unpredictable. Fields:

R unsigned int

R bit.

SYSM unsigned int

SYSm.

ArchMsg.Warning.unpredictable_ccfail_undef

TAGS **UNPREDICTABLE** DISPLAY An instruction %{OPCODE} at VA=%{VADDR} failed its condition codes check but would otherwise have caused a non-data-dependent trap or undefined exception. Behaviour is implementation-defined choice of exception or nop. Fields:

OPCODE unsigned int

opcode of the instruction.

VADDR unsigned int

Virtual address of the instruction being executed.

ArchMsg.Warning.unpredictable_ici_bits

TAGS **UNPREDICTABLE** DISPLAY ICI bits do not apply or are architecturally unpredictable for instruction.

ArchMsg.Warning.unpredictable_t32_breakpoint

TAGS **UNPREDICTABLE** DISPLAY Unpredictable breakpoint on a T32 instruction at %{ADDR1} Offset %{OFFSET} BAS %{BAS} for a %{IS_ADDRESS_MISMATCH_BREAKPOINT:(Mismatch|Match)} Breakpoint number %{N:d} Breakpoint address %{ADDR2}. Fields:

ADDR1 unsigned int

Address.

ADDR2 unsigned int

Breakpoint address.

BAS unsigned int

BAS field.

IS_ADDRESS_MISMATCH_BREAKPOINT bool

Is a Mismatch breakpoint, not a Match breakpoint.

N unsigned int

Breakpoint number.

OFFSET unsigned int

Indicates which halfword of a 32-bit Thumb instruction the breakpoint is set on. 0 for the first halfword or if the instruction is 16 bits wide, or 1 for the second halfword.

ArchMsg.Warning.unpredictable_t32_vector_catch

TAGS **UNPREDICTABLE** DISPLAY Unpredictable Vector catch exception on a T32 instruction at %{ADDR} Offset %{OFFSET}. Fields:

ADDR unsigned int

Address.

OFFSET unsigned int

Indicates which halfword of a 32-bit Thumb instruction the vector catch exception is set on. 0 for the first halfword or if the instruction is 16 bits wide, or 1 for the second halfword.

ArchMsg.Warning.unpredictable_watchpoint_far

TAGS **UNPREDICTABLE** DISPLAY The value for FAR/EDWAR for hitting this watchpoint may be anywhere in the range %{LOWER_BOUND} to %{UPPER_BOUND}. Fields:

LOWER_BOUND unsigned int

the lowest address accessed by the instruction that triggered the watchpoint.

UPPER_BOUND unsigned int

the highest watchpointed address accessed by that instruction.

ArchMsg.Warning.unpredictable_watchpoint_far_isscmp

DISPLAY The value for FAR/EDWAR for hitting this watchpoint may be anywhere in the range %{LOWER_BOUND} to %{UPPER_BOUND}. Fields:

LOWER_BOUND unsigned int

the lowest address accessed by the instruction that triggered the watchpoint.

UPPER_BOUND unsigned int

the highest watchpointed address accessed by that instruction.

ArchMsg.Warning.warning_AdvSIMDExpandImmUnexpectedZero

TAGS **UNPREDICTABLE** DISPLAY AdvSIMDExpandImm may treat this immediate value as **UNPREDICTABLE**.

ArchMsg.Warning.warning_ConditionalSMC

TAGS **UNPREDICTABLE** DISPLAY SMC instruction has **UNPREDICTABLE** effects when conditional and when combined with traps.

ArchMsg.Warning.warning_TLBID_out_of_range

TLBID out of range in TLBI according to the configuration of NIS, NVIS, NOS and NVOS.
Fields:

NIS unsigned int

NIS.

NOS unsigned int

NOS.

NVIS unsigned int

NVIS.

NVOS unsigned int

NVOS.

TLBID unsigned int

The TLBID value used in the TLBI instruction.

ArchMsg.Warning.warning_TTL_or_TG_mismatches_current_translation_granule

TTL or TG mismatches current translation granule in TLBI VA/RVA/IPA/RIPA. Fields:

CURRENT_TG unsigned int

Current TG as indicated by TCR_ELx or VTCR_EL2.

TLBI_TG unsigned int

TG (or TTL[3:2]) used in TLBI instruction.

ArchMsg.Warning.warning_access_crosses_page_boundary

TAGS UNPREDICTABLE DISPLAY Access crosses page boundary.

ArchMsg.Warning.warning_access_crosses_page_boundary_has_fault

TAGS UNPREDICTABLE DISPLAY Access that crosses page boundary has fault(s).

ArchMsg.Warning.warning_access_crosses_page_boundary_has_fault_on_both_pages

TAGS UNPREDICTABLE DISPLAY Access that crosses page boundary has a fault on each page.
Fields:

ADDR1 unsigned int

FAR address on first page.

ADDR2 unsigned int

FAR address on second page.

FSTATUS1 unsigned int

Fault status on first page.

FSTATUS2 unsigned int

Fault status on second page.

ArchMsg.Warning.warning_access_crosses_page_boundary_spanning_different_memory

TAGS UNPREDICTABLE DISPLAY Access at address %{ADDR} crosses page from MEM_TYPE %{MEMTYPE_PAGE1} to %{MEMTYPE_PAGE2}. Fields:

ADDR unsigned int

address of access crossing page.

MEMTYPE_PAGE1 signed int

First Page's memory type (0-Normal, 1-Non-Normal).

MEMTYPE_PAGE2 signed int

Second Page's memory type (0-Normal, 1-Non-Normal).

ArchMsg.Warning.warning_access_wraps_around_memory

TAGS **UNPREDICTABLE** DISPLAY Access wraps around memory in %{WIDTH:d}bit mode.
Fields:

WIDTH unsigned int

Address width.

ArchMsg.Warning.warning_bcr_linking_status

TAGS **UNPREDICTABLE** DISPLAY DBGBCR.{BT2,BT} combination gives unpredictable behaviour.

ArchMsg.Warning.warning_bcr_mask_reserved

TAGS **UNPREDICTABLE** DISPLAY DBGBCR_ELn.MASK is set to a reserved value.

ArchMsg.Warning.warning_bvr_masked_bit_not_zero

TAGS **UNPREDICTABLE** DISPLAY DBGBVRn_EL1.VA contains a bit masked by DBGBCRn_EL1.MASK that is not zero.

ArchMsg.Warning.warning_ccsidr_unimplemented_level

TAGS **UNPREDICTABLE** DISPLAY CCSIDR read while CSSELR %{CSSELR:x} points to an unimplemented cache level. Fields:

CSSELR unsigned int

current effective value of CSSELR.

ArchMsg.Warning.warning_change_to_ns_when_tge_set

TAGS **UNPREDICTABLE** DISPLAY Attempting to change NS to 1 when HCR.TGE = 1.

ArchMsg.Warning.warning_contiguous_bit_check_abort

TAGS **UNPREDICTABLE** DISPLAY An abort occurred whilst attempting to check TLB entry is contiguous at %{ENTRY_ADDR}. Fields:

ENTRY_ADDR unsigned int

address of conflicting TLB entry.

ArchMsg.Warning.warning_contiguous_bit_error

TAGS **UNPREDICTABLE** DISPLAY TLB entry at %{ENTRY_ADDR} was not contiguous with entry at %{CONTIG_ADDR} Contents are %{ENTRY_DATA} but expected %{CONTIG_DATA}.
Fields:

CONTIG_ADDR unsigned int

address of the TLB first read, with which this entry is expected to be contiguous.

CONTIG_DATA unsigned int

expected contents based on the entry at CONTIG_ADDR.

ENTRY_ADDR unsigned int

address of conflicting TLB entry.

ENTRY_DATA unsigned int

contents of conflicting TLB entry.

ArchMsg.Warning.warning_contiguous_bit_error_D128

TAGS **UNPREDICTABLE** DISPLAY TLB entry at %{ENTRY_ADDR} was not contiguous with entry at %{CONTIG_ADDR} Contents are %{ENTRY_DATA_HI} %{ENTRY_DATA_LO} but expected %{CONTIG_DATA}. Fields:

CONTIG_ADDR unsigned int

address of the TLB first read, with which this entry is expected to be contiguous.

CONTIG_DATA unsigned int

expected contents based on the entry at CONTIG_ADDR.

ENTRY_ADDR unsigned int

address of conflicting TLB entry.

ENTRY_DATA_HI unsigned int

Higher bit contents of conflicting TLB entry.

ENTRY_DATA_LO unsigned int

Lower bit contents of conflicting TLB entry.

ArchMsg.Warning.warning_cp10_cp11_mismatch

TAGS **UNPREDICTABLE** DISPLAY Unpredictable configuration of CP10 and CP11 controls at EL %{EL:d}. Fields:

EL unsigned int

EL owning the control.

ArchMsg.Warning.warning_cp10_cp11_reserved_value

TAGS **UNPREDICTABLE** DISPLAY Unpredictable configuration of CPACR CP10 (%{CP10}) or CP11 (%{CP11}). Fields:

CP10 unsigned int

CP10 access.

CP11 unsigned int

CP11 access.

ArchMsg.Warning.warning_csselr_level_out_of_range

TAGS **UNPREDICTABLE** DISPLAY CSSELR.Level out of range, written value %{LEVEL} with only %{IMPLEMENTED:d} levels implemented. Fields:

IMPLEMENTED unsigned int

number of cache levels implemented.

LEVEL unsigned int

written value of CSSELR.Level.

TYPE enum

The type of cache selected by CSSELR.InD and TnD.

ArchMsg.Warning.warning_debug_flow_control_bits_not_obeyed

TAGS UNPREDICTABLE DISPLAY Flow control bits for %{REGISTER} were not obeyed, causing %{ERROR}. Fields:

ERROR enum

type of error resulting.

REGISTER enum

register under consideration.

ArchMsg.Warning.warning_debug_register_access_during_reset

TAGS UNPREDICTABLE DISPLAY Unpredictable %{IS_WRITE:(read|write)} access to debug register offset %{OFFSET} during reset. Fields:

IS_WRITE unsigned int

Write Not Read.

OFFSET unsigned int

Register Offset.

ArchMsg.Warning.warning_decode_cps_inconsistent_fields

TAGS UNPREDICTABLE DISPLAY The mode setting fields in the CPS instruction are inconsistent.

ArchMsg.Warning.warning_decode_invalid_state

TAGS UNPREDICTABLE DISPLAY Instruction is unpredictable in current exception-level/security state.

ArchMsg.Warning.warning_default_cacheable_mmu_on

TAGS UNPREDICTABLE DISPLAY Default cacheable is enabled (HCR.DC = 1) while MMU is enabled.

ArchMsg.Warning.warning_default_cacheable_vmmu_off

TAGS UNPREDICTABLE DISPLAY Default cacheable is enabled (HCR.DC = 1) while HCR.VM = 0.

ArchMsg.Warning.warning_deprecated_wvr_bit_2_set

TAGS UNPREDICTABLE DISPLAY DBGWVR%{N:d}_EL1=%{WVR:h} has bit 2 set, deprecated in ARMv8. Fields:

N unsigned int

index.

WVR unsigned int

DBGWVR.

ArchMsg.Warning.warning_eret_while_software_step_active_pending

Missing ISB between setting MDSCR_EL1.SS and ERET TAGS **UNPREDICTABLE**.

ArchMsg.Warning.warning_exclusive_to_non_normal

TAGS **UNPREDICTABLE** DISPLAY Exclusive Access to Non-Normal Memory %{ADDR}. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_exclusive_to_non_writeback

TAGS **UNPREDICTABLE** DISPLAY Exclusive Access to Non-Writeback-Cacheable Memory %{ADDR}. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_execute_from_device_memory

TAGS **UNPREDICTABLE** DISPLAY An attempt was made to execute from Device memory.

ArchMsg.Warning.warning_illegal_cpsr_mode

TAGS **UNPREDICTABLE** DISPLAY Writing an illegal or unimplement mode (%{NEW_MODE:h}) to CPSR was unpredictable. Fields:

NEW_MODE unsigned int

New value of CPSR.M.

ArchMsg.Warning.warning_illegal_srs_mode

TAGS **UNPREDICTABLE** DISPLAY Illegal or **UNPREDICTABLE** mode (%{MODE:h}) used for SRS instruction. Fields:

MODE unsigned int

mode for Banked SP.

ArchMsg.Warning.warning_implementation_defined_read_debug_register_in_SCS

Read Access to Debug registers from the processor is IMPLEMENTATION DEFINED TAGS IMP_DEF. Fields:

PPB_OFFSET unsigned int

debug register offset.

ArchMsg.Warning.warning_implementation_defined_sequential_security_transitions_supported

The behavior of sequential instruction fetches that cross from non-secure to secure memory and contain an SG instruction is CONSTRAINED **UNPREDICTABLE** TAGS **UNPREDICTABLE**.

ArchMsg.Warning.warning_implementation_defined_stack_limit_check_supported

It is **IMPLEMENTATION DEFINED** whether stack pointer limit checking is performed for this instruction TAGS IMP_DEF. Fields:

ADDRESS unsigned int

Address of instruction.

OPCODE unsigned int

instruction opcode.

ArchMsg.Warning.warning_implementation_defined_taggednonsharedmemorytype

TAGS IMP_DEF DISPLAY Memory attribute has Tagged, non shared memory type; it is IMPLEMENTATION DEFINED whether the memory location is treated as Tagged or non-Tagged.

ArchMsg.Warning.warning_implementation_defined_write_debug_register_in_SCS

Write Access to Debug registers from the processor is IMPLEMENTATION DEFINED TAGS IMP_DEF. Fields:

DATA unsigned int

data attempted to be written.

PPB_OFFSET unsigned int

debug register offset.

ArchMsg.Warning.warning_invalid_tcr_granule

TAGS UNPREDICTABLE DISPLAY TCR.TG%{TG_ID:(0|1)} bits have been set to a granule size %{{REQUEST:d}K not implemented - using %{{SUBSTITUTE:d}K. Fields:

REQUEST signed int

page size requested (or 0 for reserved).

SUBSTITUTE signed int

best guess available page size.

TG_ID bool

bits TG0 or TG1.

ArchMsg.Warning.warning_load_multiple_user_registers_from_user_mode

TAGS UNPREDICTABLE DISPLAY An LDM(user registers) instruction executed from user mode.

ArchMsg.Warning.warning_mte_in_mem_attr_not_supported

TAGS UNPREDICTABLE DISPLAY Memory attribute %{{ATTR}} has Tagged memory type but Armv8.5-MTE is not implemented. Fields:

ATTR unsigned int

Memory attribute.

ArchMsg.Warning.warning_reserved_breakpoint_state_match

TAGS UNPREDICTABLE DISPLAY %{{IS_BREAKPOINT:(Watchpoint|Breakpoint)}} programmed with a reserved combination of HMC, SSC and %{{IS_BREAKPOINT:(PAC|PMC)}}. Fields:

IS_BREAKPOINT unsigned int

Is a breakpoint, not a watchpoint.

ArchMsg.Warning.warning_shareability

TAGS UNPREDICTABLE DISPLAY Unpredictable: combination of the 1st and 2nd stages of translation is Normal Inner Non-Cacheable, Outer Non-Cacheable.

ArchMsg.Warning.warning_software_step_set_while_enabled

TAGS UNPREDICTABLE DISPLAY MDSCR_EL1.SS set to 1 while software step debug exceptions are enabled.

ArchMsg.Warning.warning_thumb_instruction_wraps_around_memory

TAGS UNPREDICTABLE DISPLAY Thumb instruction wraps around memory.

ArchMsg.Warning.warning_ttbr_sbz_bits_are_not_zero

TAGS UNPREDICTABLE DISPLAY %{TTBR}[%{MSB:d}:0] should be zero, but are not. Fields:

MSB unsigned int

x-1.

TTBR enum

which TTBR register.

ArchMsg.Warning.warning_unaligned_address_dbgdtrrx_write

TAGS UNPREDICTABLE DISPLAY Unaligned address %{ADDR} in X0/R0 when DBGDTRRX written in memory access mode. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unaligned_address_dbgdtrtx_read

TAGS UNPREDICTABLE DISPLAY Unaligned address %{ADDR} in X0/R0 when DBGDTRTX read in memory access mode. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unaligned_to_device

TAGS UNPREDICTABLE DISPLAY Unaligned Access to Device Memory %{ADDR}. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unaligned_to_strongly_ordered

TAGS UNPREDICTABLE DISPLAY Unaligned Access to Strongly Ordered Memory %{ADDR}. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unknown_sau_rnr

TAGS UNPREDICTABLE DISPLAY SAU_RNR was set to an unsupported value.

ArchMsg.Warning.warning_unpredictable_AIRCR_PRIS_and_BFHFNMINs

TAGS UNPREDICTABLE DISPLAY The effect of setting both AIRCR.BFHFNMINs and AIRCR.PRIS to 1 is UNPREDICTABLE.

ArchMsg.Warning.warning_unpredictable_AIRCR_VECTCLRACTIVE_when_not_in_debug

TAGS **UNPREDICTABLE** DISPLAY The effect of writing a 1 to AIRCR.VECTCLRACTIVE if the processor is not halted in Debug state is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_AIRCR_VECTRESET_and_SYSRESETREQ

TAGS **UNPREDICTABLE** DISPLAY When the processor is halted in Debug state, if a write to the register writes a 1 to both VECTRESET and SYSRESETREQ, the behavior is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_AIRCR_VECTRESET_when_not_in_debug

TAGS **UNPREDICTABLE** DISPLAY The effect of writing a 1 to AIRCR.VECTRESET if the processor is not halted in Debug state is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_AIRCR_incorrect_VKEY

TAGS **UNPREDICTABLE** DISPLAY The value 0x05FA must be written to AIRCR.VKEY, otherwise the register write is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_EXC_RETURN_Reserved_Bit

EXC_RETURN[23:7] are reserved with the special condition that all bits should be written as one. Values other than all 1s are **UNPREDICTABLE**. TAGS **UNPREDICTABLE**. Fields:

EXC_RETURN unsigned int

Special PC value.

SBO_MASK unsigned int

Which bits Should Be One to not UNPRED.

ArchMsg.Warning.warning_unpredictable_PMEVTYPEPn_ELO_VS_reserved_value

TAGS **UNPREDICTABLE** DISPLAY Unpredictable reserved value of PMEVTYPEN<n>_ELO.VS.

ArchMsg.Warning.warning_unpredictable_change_to_DEMCR_MON_STEP_at_insufficient_priority

TAGS **UNPREDICTABLE** DISPLAY The effect of changing DEMCR.MON_STEP at an execution priority that is lower than the priority of the DebugMonitor exception is **CONSTRAINED UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_change_to_priority_of_active_exception

TAGS **UNPREDICTABLE** DISPLAY Changing the priority of an active exception is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_clear_lspact

TAGS **UNPREDICTABLE** DISPLAY **UNPREDICTABLE** state on context switch. Fields:

DRVEXC_TAKEN unsigned int

Derived exception taken, not pended.

REACHED_LAST_STORE unsigned int

Only the last faulted.

ArchMsg.Warning.warning_unpredictable_dbg_exit_unaligned_dlr

TAGS **UNPREDICTABLE** DISPLAY Debug exit to AArch32 with DLR[0] set to 1. PC[0] can be set to 0 or to DLR[0].

ArchMsg.Warning.warning_unpredictable_exception_catch

TAGS **UNPREDICTABLE** DISPLAY Generation of an exception catch event is unpredictable.

ArchMsg.Warning.warning_unpredictable_exception_return_inconsistent_state

TAGS **UNPREDICTABLE** DISPLAY state on exception return is unpredictable.

ArchMsg.Warning.warning_unpredictable_exception_return_instruction

TAGS **UNPREDICTABLE** DISPLAY exception return instruction is unpredictable.

ArchMsg.Warning.warning_unpredictable_in_debug_state

TAGS **UNPREDICTABLE** DISPLAY Instruction is **UNPREDICTABLE** when executed in debug state.

ArchMsg.Warning.warning_unpredictable_mair_encoding

TAGS **UNPREDICTABLE** DISPLAY Memory attribute %{ATTR} has invalid MAIR encoding; this may lead to unpredictable results. Fields:

ATTR unsigned int

Memory attribute.

ArchMsg.Warning.warning_unpredictable_pmu_counter_access

TAGS **UNPREDICTABLE** DISPLAY Access to PMU counters from non-secure EL0 or EL1 is **UNPREDICTABLE** with MDCR_EL2.HPMN set to 0.

ArchMsg.Warning.warning_unpredictable_prioritization_breakpoint_match_vector_catch

TAGS **UNPREDICTABLE** DISPLAY Two events with the same priority occurred at the same instruction: Address Matching Vector Catch debug event, and Breakpoint debug event. It is constrained unpredictable which is taken.

ArchMsg.Warning.warning_unpredictable_stack_selection

TAGS **UNPREDICTABLE** DISPLAY **UNPREDICTABLE** state on context switch.

ArchMsg.Warning.warning_unpredictable_tsize_out_of_range

TAGS **UNPREDICTABLE** DISPLAY %{TCR}.T%{TTBR:d}SZ (%{TSIZE:d} bits) is out of range. Must be between 25 and %{TSIZE_MAX:d}. Fields:

TCR enum

Which TCR.

TSIZE signed int

Value of TSize.

TSIZE_MAX signed int

The maximum allowed value.

TTBR unsigned int

Which TTBR region.

ArchMsg.Warning.warning_unpredictable_unaligned_pc_as_base_register

TAGS **UNPREDICTABLE** DISPLAY The PC value must be word-aligned, otherwise the behavior of the instruction is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_unaligned_pop_stack

TAGS **UNPREDICTABLE** DISPLAY **CONSTRAINED UNPREDICTABLE** unaligned stack frame %{FRAME_ADDR:x} in exception-return. Fields:

FRAME_ADDR unsigned int
unaligned stack frame address.

ArchMsg.Warning.warning_unpredictable_unaligned_to_non_writeback
TAGS **UNPREDICTABLE** DISPLAY Unaligned Access to Non-Writeback-Cacheable Memory.

ArchMsg.Warning.warning_unpredictable_vmsa_memattrib
TAGS **UNPREDICTABLE** DISPLAY Unpredictable vmsa memory attribute with texcb value programmed to %{TEXCB} when tex remap is %{TEX_REMAP}. Fields:

TEXCB unsigned int
texcb value.

TEX_REMAP bool
use tex remap.

ArchMsg.Warning.warning_unpredictable_vtcr_t0sz_sl0
TAGS **UNPREDICTABLE** DISPLAY The combination of VTCR.TOSZ=%{TOSZ:d} and VTCR.SLO=%{SLO:d} is unpredictable. Fields:

SLO signed int
VTCR.SLO, starting level for stage 2 translation table walks.

TOSZ signed int
VTCR.TOSZ, required input address range.

ArchMsg.Warning.warning_unpredictable_write_DHCSR
TAGS **UNPREDICTABLE** DISPLAY **UNPREDICTABLE** set of bit-changes in DHCSR. Fields:

NEW unsigned int
data attempted to be written.

OLD unsigned int
previous value.

ArchMsg.Warning.warning_unpredictable_write_SHCSR
TAGS **UNPREDICTABLE** DISPLAY **UNPREDICTABLE** setting a pending bit to 1 for an exception with priority >= execution priority.

ArchMsg.Warning.warning_unsupported_access_to_memory_mapped_register
TAGS **UNPREDICTABLE** DISPLAY Access to memory mapped register is unsupported.

ArchMsg.Warning.warning_user_jmcr_access
TAGS **UNPREDICTABLE** DISPLAY User %{WRITE:(write|read)} access to JMCR is unpredictable. Fields:

WRITE unsigned int
access is write.

ArchMsg.Warning.warning_wcr_mask_and_bas
TAGS **UNPREDICTABLE** DISPLAY DBGWCR_EL1.MASK is non-zero and BAS is not 0xff.

ArchMsg.Warning.warning_wcr_mask_reserved

TAGS **UNPREDICTABLE** DISPLAY DBGWCR_ELn.MASK is set to a reserved value.

ArchMsg.Warning.warning_wcr_non_configuous_bas

TAGS **UNPREDICTABLE** DISPLAY DBGWCR_EL1.BAS has non-contiguous set of ones.

ArchMsg.Warning.warning_wvr_masked_bit_not_zero

TAGS **UNPREDICTABLE** DISPLAY DBGWVRn_EL1.VA contains a bit masked by DBGWCRn_EL1.MASK that is not zero.

ArchMsg.Why.why_branch_target_exception

A branch target exception has occurred. DISPLAY branch target exception: target pc=%{PC}, target opcode=%{OPCODE}. PRIMARY KEY PC. Fields:

BTYPE unsigned int

pstate.BTYPE at time branch taken.

OPCODE unsigned int

opcode at branch target.

PC unsigned int

Virtual Address of branch target.

ArchMsg.Why.why_illegal_state

An illegal mode change has occurred. DISPLAY illegal mode change: %{MESSAGE}. Fields:

MESSAGE string

The message.

BRANCH_MISPREDICT

Simulating branch mispredict. Fields:

PC unsigned int

Origin address (or 0 if unavailable).

BRA_DIR

Direct branches, to immediate address. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISSET enum

The instruction set of the branch instruction.

IS_COND bool

Indicates if this is a conditional branch.

IS_HINTED bool

Whether the conditional instruction is hinted.

IS_ISB bool

Whether the instruction is ISB.

IS_LINK bool

Indicates if this is a branch with link.

PC unsigned int

The address of the branch instruction.

TARGET_ISET enum

The instruction set after the branch.

TARGET_PC unsigned int

The address the instruction branches to.

BRA_INDIR

Indirect branches, perhaps to a register. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISET enum

The instruction set of the branch instruction.

IS_COND bool

Indicates if this is a conditional branch.

IS_HINTED bool

Whether the conditional instruction is hinted.

IS_ISB bool

Whether the instruction is ISB.

IS_LINK bool

Indicates if this is a branch with link.

PC unsigned int

The address of the branch instruction.

TARGET_ISET enum

The instruction set after the branch.

TARGET_PC unsigned int

The address the instruction branches to.

BRA_INDIR_CCFAIL

Indirect branches fail, perhaps to a register.

BRA_RET_CCFAIL

Branch return fail, perhaps to a register.

CACHE_MAINTENANCE_OP

Cache Maintenance Operation. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

DATA unsigned int

Specified MVA or Set/way.

FUNCTION enum

Clean / Invalidate.

SCOPE enum

Affected region.

SIDE enum

Inst / Data.

CCFAIL

Conditional instruction condition check fail. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CCFAIL_UNC

Unconditional instruction condition check fail. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CCPASS

Conditional instruction condition check pass. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CCPASS_UNC

Unconditional instruction condition check pass. Fields:

COND **enum**

The condition of the conditional instruction.

CORE_NUM **unsigned int**

Core number in a multi processor.

PC **unsigned int**

The address of the conditional instruction.

CODE_CACHE_MAINT

Code cache maintenance. Fields:

END_INCL **unsigned int**

Inclusive end address for VA or PA requests.

NS **bool**

Non-secure world for PA.

START **unsigned int**

Start address for VA or PA requests.

TYPE **enum**

Request type.

COMPILE_BLOCK_END

Last instruction of basic block translated. Fields:

VADDR **unsigned int**

Address of next instruction after this basic block.

COMPILE_BLOCK_START

First instruction of basic block translated. Fields:

VADDR **unsigned int**

Address of first instruction in basic block.

COMPILE_INST

ARM instruction compiled. Fields:

DISASS **string**

Disassembly of the instruction.

ISSET **enum**

The instruction set of this instruction.

ITSTATE **unsigned int**

The ITSTATE current for the instruction.

OPCODE **unsigned int**

The opcode of the instruction.

PC unsigned int

The address of the instruction.

SIZE unsigned int

The size of the instruction in bytes.

CONTEXTIDR

Context ID Register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

NS bool

Secure or nonsecure banked register is accessed.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The new value written.

CONTEXT_SYNC

Called for every context synchronization event. Fields:

ARCH_CSE bool

Architectural Context Synchronization Event.

PC unsigned int

The address of the synchronization instruction.

SYSREG_SYNC bool

System register synchronization event.

CORE_ENDIAN

Core BE8 Big-Endian state changed. Fields:

BE8 bool

Core BE8 Big-Endian state.

CORE_INFO

Static processor attributes. Only triggered by a call to DumpState(). Fields:

ARCH_PROFILE enum

The architecture profile of the core.

CLUSTER_ID unsigned int

The cluster ID of this processor.

CORE_NUM unsigned int

The number of this core in an MP processor.

DETAILED_CORE_FEATURES enum

Detailed core features as a bitfield. Enum values are bitmasks to extract the info from the field.

FPU_VERSION enum

The VFP version implemented by the core.

MEM_ARCH enum

The memory architecture of the core.

MIDR_EL1 unsigned int

MIDR_EL1 value.

NUM_CORES unsigned int

The number of cores in this MP processor.

QUANTUM_SIZE unsigned int

The default quantum size of the core (in ticks).

SECURITY_FEATURES bool

Does the core have security features?.

CORE_LOADS

Processor load accesses. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ACQREL enum

Is this an acquire/release.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DATA unsigned int

The data read or written.

ELEMENT_SIZE unsigned int

Width of each element.

ENDIANNESS enum

Byte ordering of this access.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

LOCK enum

Normal, exclusive or locked access.

MECID unsigned int

Memory Encryption Context ID, if applicable.

MECID2 unsigned int

Memory Encryption Context ID of the second page, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PADDR_TAG unsigned int

If tag checked, the physical address tag for the access.

PAGE_SIZE unsigned int

Page size as $\log_2(\text{size in bytes})$.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK, 2=Exclusive Failed, 3=Unsupported.

SIZE unsigned int

Width of the access in bytes. Only required if DATA is not traced.

TAG_CHECKED bool

Is this access tag checked.

TAG_CHECKED2 bool

Is the second page access tag checked.

TRANS bool

Is this a translated access.

VADDR unsigned int

The virtual address of the access.

CORE_REGS

Changes of the core registers R0 to R14. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number, 0 to 14.

MODE enum

Bank of the register accessed.

OLD_VALUE unsigned int

The old value overwritten.

PHYS_ID enum

The physical register accessed.

VALUE unsigned int

The new value written to the register.

CORE_REGS64

Changes of the core registers X0..X30, SP_ELn. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ID enum

The register number, 0-30 for X0-X30, >=32 for SP_ELn.

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

CORE_REGS64_READ

General purpose AArch64 register has been read X0..X30, SP_ELn. Fields:

ID enum

The register number, 0-30 for X0-X30, >=32 for SP_ELn.

VALUE unsigned int

The value read from the register.

CORE_STALLED_TRANSITION

Detect when a core is prevented from progressing PChannel states. Fields:

CURRENT_STATE enum

Current power mode of core.

REASON string

The reason a transition was stalled, and what must be done to unblock it.

TARGET_STATE enum

Desired power mode of core after transition.

CORE_STORES

Processor store accesses. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ACQREL enum

Is this an acquire/release.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DATA unsigned int

The data read or written.

ELEMENT_SIZE unsigned int

Width of each element.

ENDIANNESS enum

Byte ordering of this access.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

LOCK enum

Normal, exclusive or locked access.

MECID unsigned int

Memory Encryption Context ID, if applicable.

MECID2 unsigned int

Memory Encryption Context ID of the second page, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PADDR_TAG unsigned int

If tag checked, the physical address tag for the access.

PAGE_SIZE unsigned int

Page size as $\log_2(\text{size in bytes})$.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK, 2=Exclusive Failed, 3=Unsupported.

SIZE unsigned int

Width of the access in bytes. Only required if DATA is not traced.

TAG_CHECKED bool

Is this access tag checked.

TAG_CHECKED2 bool

Is the second page access tag checked.

TRANS bool

Is this a translated access.

VADDR unsigned int

The virtual address of the access.

CP14_READ

System Coprocessor register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP14 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc1 unsigned int

opcode 1.

opc2 unsigned int

opcode 2.

CP14_WRITE

System Coprocessor register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP14 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc1 unsigned int

opcode 1.

opc2 unsigned int

opcode 2.

CP15_READ

System Coprocessor register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP15 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc1 unsigned int

opcode 1.

opc2 unsigned int

opcode 2.

CP15_READ64

System Coprocessor register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP15 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc unsigned int

opcode 1.

CP15_WRITE

System Control Coprocessor register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP15 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc1 unsigned int

opcode 1.

opc2 unsigned int
opcode 2.

CP15_WRITE64

System Coprocessor register write. Fields:

CORE_NUM unsigned int
Core number in a multi processor.

CRm unsigned int
CRm.

NS bool
Secure or nonsecure banked register is accessed.

REG_NAME string
Name of the CP15 register accessed.

SECURITY_STATE enum
The current security state.

UNDEF bool
The register accessed is undefined.

UPDATED_VALUE unsigned int
Updated value of the register now it has been written.

VALUE unsigned int
The new value written.

opc unsigned int
opcode 1.

CPSR

CPSR change. (In AArch64 PSTATE represented in SPSR format). Fields:

CORE_NUM unsigned int
Core number in a multi processor.

OLD_VALUE unsigned int
The old CPSR value.

UNKNOWN unsigned int
Bits within the register that have unknown value.

VALUE unsigned int
The new CPSR value.

CRYPTO_SPEC

Every crypto instruction speculatively executed.

CorePowerStateChange

Core power-up state has changed. DISPLAY Core powered %{PU:{up|down}}. Fields:

PU bool

Whether the core powered up.

DATA_CACHE_SET_ALLOC_TAGS

Set allocation tags from DC GVA instruction. Fields:

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

NSDESC unsigned int

The security state of the access.

PADDR unsigned int

The physical address of the access.

PAS enum

The physical address space of the page.

RESPONSE enum

0=Aborted, 1=OK.

SIZE unsigned int

The size of memory in bytes.

TAG unsigned int

The value of the tag being set.

TAGGING_TYPE enum

Type of Memory tagging (Physical or Virtual).

TAG_VADDR unsigned int

The virtual address of the Virtual Tag if Virtual tagging is used; otherwise 0.

VADDR unsigned int

Virtual address used for setting allocation tags.

DATA_CACHE_ZERO

Zero and invalidate cache line from DC ZVA instruction. Fields:

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

NSDESC unsigned int

The security state of the access.

PADDR unsigned int

The physical address of the access.

PADDR_TAG unsigned int

If tag checked, the physical address tag for the access.

PAS enum

The physical address space of the page.

RESPONSE enum

0=Aborted, 1=OK.

SIZE unsigned int

The size of zero'd memory in bytes.

TAG_CHECKED bool

Is this access tag checked.

VADDR unsigned int

Virtual address of the zero'd cache line.

DEBUG_EVENT

Hardware debug support event. Fields:

EVENT enum

Description of event.

VALUE unsigned int

data value.

DEBUG_TARGET_ISET

Hardware debug target ISA. Fields:

TARGET_ISET enum

The target instruction set of the debug state.

DMI_ALLOCATE

DMI allocated. Fields:

ACCESS_DIRECTION enum

READ / WRITE.

PADDR unsigned int

The physical address of the DMI.

PTR unsigned int

DMI range pointer.

SIDE enum

DSIDE / ISIDE.

SIZE unsigned int

The size of DMI.

VADDR unsigned int

The virtual address of the DMI..

DMI_HIT

DMI hit, access used DMI (only fire for small DMI regions i.e. less than a full page). Fields:

ACCESS_DIRECTION enum

READ / WRITE.

PADDR unsigned int

The physical address of the DMI.

SIDE enum

DSIDE / ISIDE.

SIZE unsigned int

The size of the DMI.

VADDR unsigned int

The virtual address of the DMI..

DMI_REVOKE

DMI revoked. Fields:

ACCESS_DIRECTION enum

READ / WRITE.

PADDR unsigned int

The physical address of the DMI.

SIDE enum

DSIDE / ISIDE.

SIZE unsigned int

The size of DMI.

VADDR unsigned int

The virtual address of the DMI..

END_COMPILE

Compilation end. Fields:

END_OF_PAGE_COUNT unsigned int

Number of basic blocks exited due to page boundary since START_COMPILE.

FETCHFAIL_COUNT unsigned int

Number of basic blocks exited due to ifetch failure START_COMPILE.

INST_COUNT unsigned int

Number of instructions compiled since START_COMPILE.

NONSEQ_COUNT unsigned int

Number of basic blocks exited due to non-sequential instructions since START_COMPILE.

PAGE_STRADDLE_COUNT unsigned int

Number of basic blocks exited due to unaligned instructions crossing page since START_COMPILE.

ESB

Error Synchronization Barrier hit. Fields:

ASYNC bool

ESB is async.

IMPLICIT bool

ESB is implicit.

EXCEPTION

Exception taken. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ESR unsigned int

The value assigned to ESR when exception is taken.

ESR_EC unsigned int

The Value assigned to ESR.EC when exception is taken.

IS_PHYSICAL bool

Physical or Virtual exception.

IS_TRAP bool

Exception coming from a trap.

LR unsigned int

The value assigned to the link register.

PC unsigned int

The location where the exception occurred.

PREFERRED_RETURN unsigned int

The preferred return address for the exception.

TAKEN_LOCALLY bool

Whether exception is taken locally(same exception level).

TARGET_ISET enum

The instruction set of the exception handler code.

TARGET_PC unsigned int

The address the exception branches to.

VECTOR enum

The exception vector.

VECTOR_OFFSET unsigned int

The offset of an exception vector from the base address.

EXCEPTION_END

Every exception completed.

EXCEPTION_INFO

Trace human-readable reason of the current exception for data/instruction aborts, floating point exceptions and unknown exceptions. Fields:

DESCRIPTION string

Reason for current exception.

ESR_EC unsigned int

The value of ESR.EC when exception is taken.

EXCEPTION_RAISE

Every exception raised.

EXCEPTION_RETURN

Branches on leaving exception. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISSET enum

The instruction set of the branch instruction.

IS_COND bool

Indicates if this is a conditional branch.

IS_HINTED bool

Whether the conditional instruction is hinted.

IS_ISB bool

Whether the instruction is ISB.

IS_LINK bool

Indicates if this is a branch with link.

PC unsigned int

The address of the branch instruction.

TARGET_ISET enum

The instruction set after the branch.

TARGET_PC unsigned int

The address the instruction branches to.

EXCEPTION_RETURN_PREBRANCH

Exception return event that must occur before PC and processor state have updated.

EXCEPTION_START

Every exception started.

FIQ_TAKEN

FIQ taken exception.

FP_EXCEPTIONS_SIGNALLED

Trace which FP exceptions were signalled as a part of the FP operations performed by an instruction. Fields:

FP_EXCEPTIONS_SIGNALLED_MASK unsigned int

Indicates which FP exceptions were signalled by an instruction. This uses the same encoding as the cumulative floating-point exception bitfields in FPSR.

FP_STATE

Floating point state. Fields:

FPSR unsigned int

FPSR register value.

MASK unsigned int

The mask indicating updated cumulative bits.

FREQ_CHANGED

Core frequency changed. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The instruction count of this core.

NEW_FREQ signed int

The new frequency of this core (expressed in Hz).

OLD_FREQ signed int

The old frequency of this core (expressed in Hz).

HLT

HLT instruction occurred. Fields:

IMM16 unsigned int

The 16-bit immediate value encoded in HLT instruction.

INST

Every instruction executed. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DEBUG_STATE bool

The instruction is executed in debug state.

DISASS string

Disassembly of instruction.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISET enum

The current instruction set.

ITSTATE unsigned int

The current ITSTATE.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

MODE enum

The mode the core is in.

NS unsigned int

The core's non-secure bit.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

OPCODE unsigned int

The opcode of the instruction.

PADDR unsigned int

The physical address of the instruction.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the instruction.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the page.

PC unsigned int

The address of the instruction.

SECURITY_STATE enum

The core's security state.

SIZE unsigned int

The size of the instruction in bytes.

INST_END

Every instruction completed.

INST_START

Every instruction started. Fields:

DEBUG_STATE **bool**

The instruction is executed in debug state.

ISET **enum**

The current instruction set.

MODE **enum**

The mode the core is in.

NS **enum**

The current Secure State.

PC **unsigned int**

The address of the conditional instruction.

SECURITY_STATE **enum**

The current Security State.

INST_STRADDLE

Instruction straddles boundary. Fields:

DEBUG_STATE **bool**

The instruction is executed in debug state.

ISET **enum**

The current instruction set.

MODE **enum**

The mode the core is in.

NS **enum**

The current Secure State.

PC **unsigned int**

The address of the conditional instruction.

SECURITY_STATE **enum**

The current Security State.

IRQ_TAKEN

IRQ taken exception.

ISB

ISB instruction occurred. Fields:

PC **unsigned int**

The address of the ISB instruction.

LOAD_MULTIPLE_REGS_LIST

List of destination registers (not source) of a multiple registers load instruction. Fields:

REG string

Register to add into the list.

LOCAL_MONITOR

Local monitor activity. Fields:

PADDR unsigned int

Local Monitor Address.

State enum

State of the monitor (Open/Exclusive).

MAPPING_CREATE

A memory mapping has been created. This is an implementation detail of the model and so may change without notice. Fields:

ASID unsigned int

The ASID used during translation, or 0 if no ASID was used.

IS_VALID bool

Whether the mapping is valid or not.

PADDR unsigned int

Base physical address of region, or 0 if not known.

SIDE enum

I-side or D-side.

SIZE unsigned int

Size of the page, or 0 if not known.

VADDR unsigned int

Base virtual address of region.

VMID unsigned int

The VMID used during translation, or 0 if no VMID was used.

MAPPING_DESTROY

A memory mapping has been destroyed. This is an implementation detail of the model and so may change without notice. Fields:

ASID unsigned int

The ASID used during translation.

CAUSE enum

The cause of invalidation.

PADDR unsigned int

Base physical address of page.

SIDE enum

I-side or D-side.

SIZE unsigned int

Size of the page.

VADDR unsigned int

Base virtual address of page.

VMID unsigned int

The VMID used during translation.

MEMMAP_DEBUG_READ

Memory mapped reads to the debug registers. Fields:

ADDR unsigned int

address.

CORE_NUM unsigned int

Core number in a multi processor.

EXT bool

Whether access is from an external device (such as the DAP).

NS enum

Secure state of the access.

REG_NAME string

Name of the debug register accessed.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

MEMMAP_DEBUG_WRITE

Memory mapped writes to the debug registers. Fields:

ADDR unsigned int

address.

CORE_NUM unsigned int

Core number in a multi processor.

EXT bool

Whether access is from an external device (such as the DAP).

NS enum

Secure state of the access.

REG_NAME string

Name of the debug register accessed.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The new value written.

MEMTAG_LOADS

Allocation tag memory read; when attributes and system configuration allow tag accesses.

Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

NUM_GRANULES unsigned int

Number of tags to be Loaded/stored.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK.

TAGGING_TYPE enum

Type of Memory tagging (Physical or Virtual).

TAGS unsigned int

The tags read or written. Tags for subsequent granules would be found at subsequent bytes in this field (One byte for each tag).

TAG_VADDR unsigned int

The virtual address of the Virtual Tag if Virtual tagging is used; otherwise 0.

VADDR unsigned int

The virtual address of the access.

MEMTAG_LOAD_INST

Allocation tag load instruction; regardless of tags being actually accessed. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

NUM_GRANULES unsigned int

Number of tags to be Loaded/stored.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK.

TAGGING_TYPE enum

Type of Memory tagging (Physical or Virtual).

TAGS unsigned int

The tags read or written. Tags for subsequent granules would be found at subsequent bytes in this field (One byte for each tag).

TAG_VADDR unsigned int

The virtual address of the Virtual Tag if Virtual tagging is used; otherwise 0.

VADDR unsigned int

The virtual address of the access.

MENTAG_STORES

Allocation tag memory write; when attributes and system configuration allow tag accesses.

Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

NUM_GRANULES unsigned int

Number of tags to be Loaded/stored.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK.

TAGGING_TYPE enum

Type of Memory tagging (Physical or Virtual).

TAGS unsigned int

The tags read or written. Tags for subsequent granules would be found at subsequent bytes in this field (One byte for each tag).

TAG_VADDR unsigned int

The virtual address of the Virtual Tag if Virtual tagging is used; otherwise 0.

VADDR unsigned int

The virtual address of the access.

MENTAG_STORE_INST

Allocation tag store instruction; regardless of tags being actually accessed. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

NUM_GRANULES unsigned int

Number of tags to be Loaded/stored.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK.

TAGGING_TYPE enum

Type of Memory tagging (Physical or Virtual).

TAGS unsigned int

The tags read or written. Tags for subsequent granules would be found at subsequent bytes in this field (One byte for each tag).

TAG_VADDR unsigned int

The virtual address of the Virtual Tag if Virtual tagging is used; otherwise 0.

VADDR unsigned int

The virtual address of the access.

MMU_D128_TTB_READ

This event is triggered by reads caused by a D128 translation table walk. Fields:

ABORTED bool

Was the walk successful.

DATA_HIGH unsigned int

The data upper bits read.

DATA_LOW unsigned int

The data lower bits read.

IPA unsigned int

For stage 1, the IPA of the read.

LEVEL unsigned int

Translation table level.

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

OUTADDR unsigned int

The output physical address.

PADDR unsigned int

The physical address of the read.

PASmem enum

Physical Address Space of the accessed memory.

PASreq enum

Physical Address Space of the requested lookup.

REGIME_EL enum

Entry matches in this translation regime.

REQUESTER enum

The requester of this table walk.

SIDE enum

Inst / Data.

STAGE unsigned int

Translation stage.

MMU_D128_TTB_WRITE

This event is triggered by writes caused by a D128 translation table walk. Fields:

ABORTED bool

Was the walk successful.

DATA_HIGH unsigned int

The data upper bits written.

DATA_LOW unsigned int

The data lower bits written.

IPA unsigned int

For stage 1, the IPA of the write.

IS_HACDBS_ACCESS bool

If the access was due to HACDBS_CLEAN.

LEVEL unsigned int

Translation table level.

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

OUTADDR unsigned int

The output physical address.

PADDR unsigned int

The physical address of the write.

PASmem enum

Physical Address Space of the accessed memory.

PASreq enum

Physical Address Space of the requested lookup.

REGIME_EL enum

Entry matches in this translation regime.

SIDE enum

Inst / Data.

STAGE unsigned int

Translation stage.

MMU_TRANS

Address translation information. Fields:

ASID unsigned int

Address space identifier.

CORE_NUM unsigned int

Core number in a multi processor.

Hyp bool

Entry matches in Hyp state only.

INNERCACHE_RA bool

Is the inner cache allocate on read.

INNERCACHE_TRANSIENT bool

Is the inner write-through transient.

INNERCACHE_TYPE enum

Inner Caching scheme (NC/MB/WA).

INNERCACHE_WA bool

Is the inner cache allocate on write.

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MEMTYPE enum

Memory type.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC enum

Is secure side supposed to access secure or nonsecure memory.

OUTERCACHE_RA bool

Is the outer cache allocate on read.

OUTERCACHE_TRANSIENT bool

Is the outer write-through transient.

OUTERCACHE_TYPE enum

Outer Caching scheme (NC/MB/WA).

OUTERCACHE_WA bool

Is the outer cache allocate on write.

PADDR unsigned int

Physical address of the access.

PAGESIZE unsigned int

Page size as $\log_2(\text{size})$.

PAS enum

PAS supposed to be accessed is secure or nonsecure memory or Root/Realm if RME is implemented.

SH enum

Shareability.

SIDE enum

Inst / Data.

STAGE1_PERM unsigned int

Stage 1 permissions, [5:3] Privileged access (XWR) [2:0] User access (XWR).

STAGE2_PERM unsigned int

Stage 2 permission mask [5:3] Privileged access (XWR) [2:0] User access (XWR).

VADDR unsigned int

Virtual address of the access.

VMID unsigned int

Virtual machine identifier.

XS bool

XS Attribute.

nG enum

Flag indicating whether ASID will be matched.

MMU_TTB_READ

This event is triggered by reads caused by a translation table walk. Fields:

ABORTED bool

Was the walk successful.

DATA unsigned int

The data read.

DESC_KIND enum

The encoding scheme of the descriptors used for this walk.

IPA unsigned int

For stage 1, the IPA of the read.

LEVEL unsigned int

Translation table level.

LPAE bool

Is this for an LPAE translation (*DEPRECATED*, use DESC_KIND).

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSmem bool

Non secure memory access.

NSreq bool

Non secure request.

OUTADDR unsigned int

The output physical address.

PADDR unsigned int

The physical address of the read.

PASmem enum

Physical Address Space of the accessed memory.

PASreq enum

Physical Address Space of the requested lookup.

REGIME_EL enum

Entry matches in this translation regime.

REQUESTER enum

The requester of this table walk.

SIDE enum

Inst / Data.

STAGE unsigned int

Translation stage.

MMU_TTB_WRITE

This event is triggered by writes caused by a translation table walk. Fields:

ABORTED bool

Was the walk successful.

DATA unsigned int

The data written.

DESC_KIND enum

The encoding scheme of the descriptors used for this walk.

IPA unsigned int

For stage 1, the IPA of the write.

IS_HACDBS_ACCESS bool

If the access was due to HACDBS_CLEAN.

LEVEL unsigned int

Translation table level.

LPAE bool

Is this for an LPAE translation (*DEPRECATED*, use DESC_KIND).

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSmem bool

Non secure memory access.

NSreq bool

Non secure request.

OUTADDR unsigned int

The output physical address.

PADDR unsigned int

The physical address of the write.

PASmem enum

Physical Address Space of the accessed memory.

PASreq enum

Physical Address Space of the requested lookup.

REGIME_EL enum

Entry matches in this translation regime.

SIDE enum

Inst / Data.

STAGE unsigned int

Translation stage.

MODE_CHANGE

Mode change. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

MODE enum

The new mode.

NON_SECURE enum

The core's new non-secure bit.

OLD_MODE enum

The old mode.

SECURITY_STATE enum

The new security state.

NOT_TAG_CHECKED_INST

MTE Tag not checked for current instruction.

PERIODIC

Called for every quantum. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The instruction count of this CPU.

PC unsigned int

The address of the next instruction to be executed on this CPU.

PMU_COUNTER_OVERFLOW

PMU counter overflow. Fields:

COUNTER_GROUP enum

Counter group.

COUNTER_RANGE enum

Counter range.

EVENT_ID unsigned int

Event Identifier, field is valid only for Event counter group.

INDEX unsigned int

Counter index (as selected by PMSELR), field is valid only for Event/Cycle counter groups.

INTERRUPT bool

Is interrupt enabled on overflow for this counter.

POINTER_AUTH

Called when execution hits a pointer authentication check. Fields:

AUTH_STATE bool

Authentication result.

KEY_USED enum

Pointer authentication key.

POTINDEX_UPDATE

Value of POT index for current pc changed. Fields:

NEW_POTINDEX unsigned int

New POTIndex.

OLD_POTINDEX unsigned int

Old POTIndex.

PREFETCH_MEMORY64

Prefetch from PRFM or PRFUM instructions. Fields:

PRFOP enum

Prefetch hint.

VADDR unsigned int

Virtual address of the location that should be prefetched.

PRELOAD64_REQUEST

Preload requests from PRFM or PRFUM instructions. Fields:

PRFOP enum

Prefetch hint.

VADDR unsigned int

Virtual address of the location that should be prefetched.

PRELOAD_DATA

Data preload from PLD instruction. Fields:

VADDR unsigned int

Virtual address of the data that should be preloaded.

PRELOAD_INST

Instruction preload from PLI instruction. Fields:

VADDR unsigned int

Virtual address of the instruction that should be preloaded.

PRE_CORE_LOAD

Trace just before a core load. Fields:

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

PRE_CORE_STORE

Trace just before a core store. Fields:

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

PRE_TTB_READ

This event is triggered before reads caused by a translation table walk. Fields:

ADDR unsigned int

The physical address of the read.

PSTATE

PSTATE fields update. Fields:

ALLINT unsigned int

All interrupt mask bit.

BTYPE unsigned int

Branch target identification bit.

CORE_NUM unsigned int

Core number in a multi processor.

DAIF unsigned int

Exception mask bits.

DIT unsigned int

Data Independent Timing bit.

EL unsigned int

Current Exception level field.

EXLOCK unsigned int

Exception return state lock bit.

IL unsigned int

Illegal Execution state bit.

NZCV unsigned int

Condition flags.

PACM unsigned int

PACM bit.

PAN unsigned int

Privileged Access Never state bit.

PM unsigned int

PMU exception mask bit.

PPEND unsigned int

PMU exception pending bit.

SM unsigned int

StreamingMode status bit.

SP unsigned int

Stack pointer register selection bit.

SS unsigned int

Software Step bit.

SSBS unsigned int

Speculative Store Bypass Safe bit.

TCO unsigned int

Tag Check Override bit.

TOUCHED unsigned int

Mask of the fields that have been touched since this trace has last been emitted. Note that touched does not necessarily mean the value changed. The index of the field is the bit index in the mask plus the index of this field plus one.

UAO unsigned int

User Access Override bit.

UINJ unsigned int

Undefined Instruction exceptions bit.

ZA unsigned int

ZA storage and ZT0 register enable bit.

nRW unsigned int

Current Execution state bit.

PSTATE_PACTIVE_OPMODE_TRANSITION

Trace pactive and Opmode mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Operating mode after the transition.

OLD_STATE enum

Operating mode before the transition.

PSTATE_PACTIVE_OPMODE_TRANSITION_DENIED

Trace the denied pactive and Opmode mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Operating mode after the transition.

OLD_STATE enum

Operating mode before the transition.

REASON string

Reason for the transition being denied.

PSTATE_PACTIVE_POWER_TRANSITION

Trace pactive and power mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Power mode after the transition.

OLD_STATE enum

Power mode before the transition.

PSTATE_PACTIVE_POWER_TRANSITION_DENIED

Trace the denied pactive and power mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Power mode after the transition.

OLD_STATE enum

Power mode before the transition.

REASON string

Reason for the transition being denied.

PSTATE_PACTIVE_THREAD_OPMODE_TRANSITION

Trace pactive and Opmode mode transitions for threads on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Operating mode after the transition.

OLD_STATE enum

Operating mode before the transition.

QUANTUM_START

Called at the start of every quantum.

RANGE_PRELOAD_REQUEST

Preload requests from RPRFM. Fields:

COUNT unsigned int

Number of blocks of data to access.

LENGTH unsigned int

Number of contiguous bytes to be accessed in each block.

OP unsigned int

Range prefetch operation .

REUSE unsigned int

Maximum number of bytes to be accessed before next RPRFM .

STRIDE signed int

Stride to advance after accessing LENGTH bytes.

VADDR unsigned int

Base virtual address of range prefetch request.

RUN_STATE

Run state transition. Fields:

INST_COUNT unsigned int

Ticks count at point of transition.

NEW enum

New run state.

OLD enum

Old run state.

SEMIHOSTING_CALL

Call of a semihost function occurred. Fields:

PC unsigned int

The program counter after the semihosting call.

SEMIHOSTING_PRECALL

About to call semihost. Fields:

EL2 bool

Use EL2 translation regime.

NS bool

Is Non-Secure.

REG_WIDTH unsigned int

The current register width in bytes.

SIGNAL

External signal state change. Fields:

SIGNAL enum

Signal that changed.

STATE bool

Signal asserted state.

SOFTWARE_STEP

Return the Debug Software Step state. Fields:

STATE enum

Software Step state.

SPSR

SPSR change. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

MODE enum

Which of the banked SPSR registers is written.

OLD_VALUE unsigned int

The old SPSR value.

VALUE unsigned int

The new SPSR value.

START_COMPILE

Compilation started. Fields:

VADDR unsigned int

Instruction where compilation begins.

STG_MULTIPLE_MICRO_OPS_ADDRESS_RANGE

Address range for MTE stores with multiple uops. Fields:

ADDR_END unsigned int

Final address.

ADDR_START unsigned int

Starting address.

STORE_MULTIPLE_REGS_ADDRESS_RANGE

Address range for stores with multiple registers. Fields:

ADDR_END unsigned int

Final address.

ADDR_START unsigned int

Starting address.

SUBOPTIMAL_LDST_RETIRED

suboptimal load/store. Fields:

ACCESS_SIZE unsigned int

Log2 of the access width in bytes used for alignment checking (1: half-word, 2: word ...).

OPTIMAL_ALIGN_SELECTOR unsigned int

Selector to decide the optimal alignment size (1: access width, 2: 4-bytes, 3: 8-bytes, ..., 12: 4-kbytes).

VADDR unsigned int

The virtual address of the access.

SVE_INST_SPEC

SVE operations speculatively executed.

SVE_LD_RETIRED

SVE load instructions architecturally executed.

SVE_LOADS

SVE load accesses. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ACQREL enum

Is this an acquire/release.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DATA unsigned int

The data read or written.

ELEMENT_SIZE unsigned int

Width of each element.

ENDIANNESS enum

Byte ordering of this access.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

LOCK enum

Normal, exclusive or locked access.

MECID unsigned int

Memory Encryption Context ID, if applicable.

MECID2 unsigned int

Memory Encryption Context ID of the second page, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PADDR_TAG unsigned int

If tag checked, the physical address tag for the access.

PAGE_SIZE unsigned int

Page size as $\log_2(\text{size in bytes})$.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK, 2=Exclusive Failed, 3=Unsupported.

SIZE unsigned int

Width of the access in bytes. Only required if DATA is not traced.

TAG_CHECKED bool

Is this access tag checked.

TAG_CHECKED2 bool

Is the second page access tag checked.

TRANS bool

Is this a translated access.

VADDR unsigned int

The virtual address of the access.

SVE_MEM_PREFETCH

Scalable Vector Extension memory prefetch. Fields:

CACHE_LEVEL enum

The preferred cache level.

HINT enum

What the memory is expected to be used for.

SIZE unsigned int

The prefetch size in bytes.

TEMPORAL bool

Whether the data is temporal.

VADDR unsigned int

The virtual address of the prefetch.

SVE_REG_READ

Scalable Vector Extension register read. Fields:

ID enum

The register identifier.

SM bool

Whether the PE is in Streaming Mode.

VALUE unsigned int

The read data.

SVE_REG_UPDATE

Scalable Vector Extension register update. Fields:

ID enum

The register identifier.

SM bool

Whether the PE is in Streaming Mode.

VALUE unsigned int

The written data.

SVE_REG_WRITE

Scalable Vector Extension register write. Fields:

ID enum

The register identifier.

SM bool

Whether the PE is in Streaming Mode.

VALUE unsigned int

The written data.

SVE_STORES

SVE store accesses. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ACQREL enum

Is this an acquire/release.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DATA unsigned int

The data read or written.

ELEMENT_SIZE unsigned int

Width of each element.

ENDIANNESS enum

Byte ordering of this access.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

LOCK enum

Normal, exclusive or locked access.

MECID unsigned int

Memory Encryption Context ID, if applicable.

MECID2 unsigned int

Memory Encryption Context ID of the second page, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PADDR_TAG unsigned int

If tag checked, the physical address tag for the access.

PAGE_SIZE unsigned int

Page size as $\log_2(\text{size in bytes})$.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK, 2=Exclusive Failed, 3=Unsupported.

SIZE unsigned int

Width of the access in bytes. Only required if DATA is not traced.

TAG_CHECKED bool

Is this access tag checked.

TAG_CHECKED2 bool

Is the second page access tag checked.

TRANS bool

Is this a translated access.

VADDR unsigned int

The virtual address of the access.

SVE_ST_RETIRED

SVE store instructions architecturally executed.

SVE_UNALIGNED_LDST_RETIRED

SVE load or store instructions architecturally executed that would cause alignment fault if A bit is set.

SVE_Z_PRED_MEM_ACCESS_INFO

Information about SVE/SME predicated accesses to Z registers (non Gather/Scatter). Fields:

MEM_ELEM_SIZE unsigned int

Width of elements accessed in memory (in bytes).

RnW bool

Whether this is a Read not a Write.

VA unsigned int

Virtual base address computed for accesses.

SYNC

Called for every synchronization, that is every quantum break. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The instruction count of this CPU.

LOCAL_QUANTUM unsigned int

The local quantum of this CPU.

LOCAL_TIME unsigned int

The local time of this CPU.

SYNC_REASON enum

Reason for sync.

SYSCALL

System call instruction executed. Fields:

IMM unsigned int

Immediate value of the system call instruction.

TYPE enum

System call type.

VADDR unsigned int

Instruction that caused the system call.

SYSREG_READ128

System Coprocessor 128 bit register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

REGNUM enum

Internal register number.

REGNUM_PHYS enum

Internal register number physically accessed (e.g. in presense of VHE redirection).

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_READ64

System Coprocessor register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REGNUM enum

Internal register number.

REGNUM_OPERAND enum

Register number corresponding to the instruction operand.

REGNUM_PHYS enum

Internal register number physically accessed (e.g. in presense of VHE redirection).

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_UPDATE128

Triggers when the system updates a register. Fields:

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG enum

Register number.

UNKNOWN unsigned int

Bits of the register which became unknown.

VALUE unsigned int

Value written to the register.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_UPDATE32

Triggers when the system updates a register. Fields:

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG enum

Register number.

UNKNOWN unsigned int

Bits of the register which became unknown.

VALUE unsigned int

Value written to the register.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_UPDATE64

Triggers when the system updates a register. Fields:

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG enum

Register number.

UNKNOWN unsigned int

Bits of the register which became unknown.

VALUE unsigned int

Value written to the register.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_WRITE128

System Coprocessor 128 bit register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

REGNUM enum

Internal register number.

REGNUM_PHYS enum

Internal register number physically accessed (e.g. in presense of VHE redirection).

UNDEF bool

The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_WRITE64

System Coprocessor register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REGNUM enum

Internal register number.

REGNUM_OPERAND enum

Register number corresponding to the instruction operand.

REGNUM_PHYS enum

Internal register number physically accessed (e.g. in presense of VHE redirection).

UNDEF bool

The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

TRBU_BUFFER_WRAP

DISPLAY Trace buffer current write pointer wrapped.

TRBU_TRIGGER_EVENT

DISPLAY Trace buffer Trigger Event.

TRBU_WRITE

DISPLAY PETU trace data has been written to the TRBU memory buffer at `%{BUFFER_ADDRESS}` with size `%{SIZE}` and data `%{DATA}`. Fields:

BUFFER_ADDRESS unsigned int

Virtual Address of start of buffer write.

DATA unsigned int

Data written to the buffer.

SIZE unsigned int

Size of sample written to buffer, in bytes.

UNALIGNED_LDST_RETIRED

Processor unaligned load/store. Fields:

ACCESS_SIZE unsigned int

Log2 of the access width in bytes used for alignment checking (1=>half-word, 2->word ...).

NUMBER_OF_BEATS unsigned int

Number of accesses (beats) in this burst (Total size in bytes is `ACCESS_SIZE * NUMBER_OF_BEATS`).

VADDR unsigned int

The virtual address of the access.

VFP_D_REGS

VFP/NEON D 64 bit register write. Fields:

ALIASING enum

Type of register aliasing used.

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a write to the corresponding bit in the register. For example, 0x1 would indicate a write to VALUE[0].

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

VFP_D_REGS_READ

VFP/NEON D 64 bit register read. Fields:

ID unsigned int

The register number.

VALUE unsigned int

The current value of the register.

VFP_Q_REGS

VFP/NEON Q 128 bit register write. Fields:

ALIASING enum

Type of register aliasing used.

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a write to the corresponding byte in the register. For example, 0x1 would indicate a write to VALUE[7:0].

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

VFP_Q_REGS_READ

VFP/NEON Q 128 bit register read. Fields:

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a read to the corresponding byte in the register. For example, 0x1 would indicate a read to VALUE[7:0].

VALUE unsigned int

The current value of the register.

VFP_SYS_REGS

Writes to the VFP/NEON units system registers. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ID enum

Which VFP system register is written.

MASK unsigned int

The mask indicating updated cumulative bits.

OLD_VALUE unsigned int

The register's old value overwritten.

VALUE unsigned int

The new value written to the VFP system register.

VFP_S_REGS

VFP/NEON S 32 bit register write. Fields:

ALIASING enum

Type of register aliasing used.

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a write to the corresponding bit in the register. For example, 0x1 would indicate a write to VALUE[0].

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

VFP_S_REGS_READ

VFP/NEON S 32 bit register read. Fields:

ID unsigned int

The register number.

VALUE unsigned int

The current value of the register.

VIRTUAL_TAG_ALLOCATED

Virtual tag information when allocated. Fields:

TAG_VADDR **unsigned int**

Virtual tag virtual address.

VADDR **unsigned int**

Data Virtual address.

VIRTUAL_TAG **unsigned int**

Virtual tag value.

WAYPOINT

Signals end of basic block execution. Fields:

CORE_NUM **unsigned int**

Core number in a multi processor.

ISET **enum**

Origin instruction set.

IS_COND **bool**

Indicates if this is a conditional waypoint.

PC **unsigned int**

Origin address (or 0 if unavailable).

TAKEN **bool**

Indicates if this waypoint was taken.

TARGET **unsigned int**

Destination address.

TARGET_ISET **enum**

Destination instruction set.

WFE_END

WFE ended. Fields:

INST_COUNT **unsigned int**

Ticks count when leaving WFE.

WFE_EVENT_REGISTER

WFE event register status: set/clear, reason. Fields:

INST_COUNT **unsigned int**

Ticks count.

REASON **enum**

Reason for set/clear. Only REASON==0 (Cleared by WFE) clears the bit, all other reasons set it.

WFE_IGNORED

WFE ignored. Fields:

EVENT bool

This WFE was ignored because the event register was set.

INST_COUNT unsigned int

Ticks count when ignoring WFE.

TRAPPED bool

This WFE was trapped.

WFE_START

WFE entered. Fields:

INST_COUNT unsigned int

Ticks count when entering WFE.

WFI_END

WFI ended. Fields:

INST_COUNT unsigned int

Ticks count when leaving WFI.

WFI_IGNORED

WFI ignored. Fields:

DISABLED bool

This WFI was ignored because WFI is disabled.

INST_COUNT unsigned int

Ticks count when ignoring WFI.

REASON enum

specifies reason why WFI trace was ignored.

TRAPPED bool

This WFI was trapped.

WFI_START

WFI entered. Fields:

INST_COUNT unsigned int

Ticks count when entering WFI.

WFI_WAKEUP

WFI wakeup. Fields:

INST_COUNT unsigned int

Ticks count when WFI wakeup occurred.

REASON enum

Reason for wakeup.

2.26 ARM_Cortex-A72

This section describes the trace sources.

AA64_ASE_SVE_REGS

Modification of AArch64 Advanced SIMD or SVE Vector Registers (via SIMD or Scalar operations), SVE Predicate Registers and ZT0. Fields:

ID **enum**

The register identifier.

MASK **unsigned int**

Bitmask of the register to signal the modified bits in the VALUE field.

SM **bool**

Whether the PE is in Streaming Mode.

VALUE **unsigned int**

Value of AArch64 vector register.

ASYNC_MEMORY_FAULT

Context ID Register write. Fields:

FAULT **unsigned int**

Fault status present in ESR.ISS encoding for Serror in ARCH64 and present in DFSR encoding for external abort format in ARCH32.

PADDR **unsigned int**

Physical Address (or 0 if unavailable).

VADDR **signed int**

Virtual Address (or 0 if unavailable).

ATOMIC_END_ACCESS

Bus trace access for atomics. Fields:

ACCESS_FAIL **bool**

Memory access failed.

ADDR **unsigned int**

The virtual address of the access.

ATTR **unsigned int**

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

COMPARE_VALUE **unsigned int**

Compare value for CAS.

LOAD_VALUE unsigned int

Loaded values.

NSDESC unsigned int

The security state of the access.

OPERAND_VALUE unsigned int

Operation's operand.

OPERATION enum

Operation type.

PADDR unsigned int

The physical address of the access.

PRIV bool

Is this a privileged access?.

ATOMIC_START_ACCESS

Bus trace access for atomics. Fields:

ADDR unsigned int

The virtual address of the access.

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

COMPARE_VALUE unsigned int

Compare value for CAS.

NSDESC unsigned int

The security state of the access.

OPERAND_VALUE unsigned int

Operation's operand.

OPERATION enum

Operation type.

PADDR unsigned int

The physical address of the access.

PRIV bool

Is this a privileged access?.

ArchMsg.Error.exit_code

DISPLAY exit code: %{EXITCODE} component kind: %{KIND} reason: %{REASON}. Fields:

EXITCODE signed int

Exit code to be returned.

KIND string

Component kind that invoked the exit code trace.

REASON string

Reason why the exit code trace was invoked (optional).

ArchMsg.Warning.IgnoredWarmResetRequest

Event fires when software attempts to request warm reset through write to deprecated DBGPRCR.CWRR.

ArchMsg.Warning.ReportedRevisionRegisterRead

DISPLAY Reported revision and patch do not match that of the cpu. Fields:

ORIGINAL_VALUE unsigned int

The value the register would have based on the modelled revision.

REGISTER string

The name of the register that has ben read.

REPORTED_VALUE unsigned int

The value the register has been overridden to.

ArchMsg.Warning.Unpredictably Indexed PM Event Register

TAGS **UNPREDICTABLE** DISPLAY Event %{IsDirect:(Selector value|Register)} %{SEL} is >= %{N}. Fields:

IsDirect unsigned int

Direct Access rather than indirect.

N unsigned int

Number of Accessible Registers.

SEL unsigned int

Selector.

ArchMsg.Warning.cache_contents_unknown

A cache has been activated without being invalidated and may contain **UNKNOWN** content. DISPLAY %{SIDE:(D|I)}-Cache was enabled in %{REGIME:(SECURE|NON_SECURE|HYP)} regime but was not invalidated since power-on: cache lines could contain **UNKNOWN** content %{SIDE:(Note: D-side caches must invalidate SECURE lines even if the cache is only used by NON_SECURE code in order to ensure that unexpected dirty lines tagged SECURE are not naturally evicted)}. Fields:

REGIME unsigned int

Current translation regime.

SIDE unsigned int

0 d-side cache or unified cache, 1 i-side.

ArchMsg.Warning.dcimvac_matches_watchpoint

TAGS **UNPREDICTABLE** DISPLAY Watchpoints matching an AArch32 DCIMVA are implementation defined.

ArchMsg.Warning.decode_fieldmismatch

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with different fields %{FIELD1} and %{FIELD2}. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_fracbitsnegative

TAGS **UNPREDICTABLE** DISPLAY VCVT Instruction is unpredictable with a negative number of fraction bits.

ArchMsg.Warning.decode_initblock

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable in an IT block.

ArchMsg.Warning.decode_initblocknotlast

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable in an IT block when not the last.

ArchMsg.Warning.decode_invalid_condition

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable when the condition is not AL.

ArchMsg.Warning.decode_invalidfieldcombination

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with the values in %{FIELD1} and %{FIELD2}. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_invalidvalue

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with the value in %{FIELD}. Fields:

FIELD enum

Field name.

ArchMsg.Warning.decode_mem_hint_unallocated

TAGS **UNPREDICTABLE** DISPLAY There is no memory hint allocated to this bit pattern.

ArchMsg.Warning.decode_registermatch

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with %{FIELD1} and %{FIELD2} as the same register. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_registermismatch

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with %{FIELD1} and %{FIELD2} as different registers. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_registeroutofrange

TAGS **UNPREDICTABLE** DISPLAY Floating-point instruction is unpredictable indexing beyond the end of the register bank.

ArchMsg.Warning.decode_sbzsbo

TAGS **UNPREDICTABLE** DISPLAY Reserved bits in the instruction are not canonical.

ArchMsg.Warning.decode_transactiontoobig

TAGS **UNPREDICTABLE** DISPLAY Attempt to perform a memory transaction of more than 128 bytes.

ArchMsg.Warning.decode_unpred_other

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable.

ArchMsg.Warning.decode_unpreduseofpc

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable when %{FIELD} is PC. Fields:

FIELD enum

Field name.

ArchMsg.Warning.decode_unpreduseofr13

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable when %{FIELD} is R13. Fields:

FIELD enum

Field name.

ArchMsg.Warning.decode_unpreduseofr13inreglist

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with R13 in register-list.

ArchMsg.Warning.decode_useofd16orhigher

TAGS **UNPREDICTABLE** DISPLAY Attempt to use register D16 or higher where it is not allowed.

ArchMsg.Warning.decode_writebackandbaseinlist

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with write back when the base register is in the transfer list.

ArchMsg.Warning.decode_zeroregistersinlist

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable when no registers are to be transferred.

ArchMsg.Warning.recursive_branch

An instruction is performing a branch that targets the same instruction. If this was intended then a WFI might be more effective. DISPLAY recursive branch detected at PC=%{PC}.

PRIMARY KEY PC. Fields:

PC unsigned int

Address of instruction causing the branch.

ArchMsg.Warning.recursive_exception

An instruction has generated an exception that targets the same instruction. DISPLAY recursive %{TYPE} exception detected at PC=%{PC} (CPSR=%{CPSR}). PRIMARY KEY PC. Fields:

CPSR unsigned int

Processor state.

PC unsigned int

Address of instruction causing the exception.

TYPE enum

Type of exception.

ArchMsg.Warning.reentrant_vector_catch

TAGS **UNPREDICTABLE** DISPLAY Vector catch debug event on Prefetch or Data abort vector %{ADDR}. Fields:

ADDR unsigned int

vector address.

ArchMsg.Warning.reserved_it_state

TAGS **UNPREDICTABLE** DISPLAY Execution with reserved IT state %{ITSTATE}. Fields:

ITSTATE unsigned int

reserved IT state.

ArchMsg.Warning.tlb_contents_unknown

Translation has been enabled without invalidating the TLB. DISPLAY Translation was enabled in the %{PAS:(Secure|Non-secure|Root|Realm)} EL%{EL:d} translation regime (vmid=%{VMID:x}) without invalidating %{INVALIDITY:(|d-side |i-side |)}entries since power-on: %{INVALIDITY:(|D||I)}TLB could contain **UNKNOWN** content. Fields:

EL unsigned int

Exception level of translation regime.

INVALIDITY unsigned int

Which TLB may be invalid (if TLBs are separate).

NON_SECURE bool

Whether the translation regime is non-secure.

PAS unsigned int

Physical address space of translation regime.

VMID unsigned int

Current VMID, if applicable.

ArchMsg.Warning.unknown_DLR_DSPSR

Exiting debug state with unknown DLR or DSPSR.

ArchMsg.Warning.unknown_ELR_SPSR

Returning from debug exception with unknown ELR or SPSR.

ArchMsg.Warning.unpred_FAR_on_LRCPC3_non_atomic_mem_access

TAGS **UNPREDICTABLE** Unpredictable FAR/FSTATUS on LRCPC3 pair instructions resulting in two separate memory accesses
DISPLAY Unpredictable FAR on LRCPC3 non atomic memory access at %{ADDR1} with FSTATUS %{FSTATUS1} or %{ADDR2} with FSTATUS %{FSTATUS2}. Fields:

ADDR1 unsigned int

Possible FAR address 1.

ADDR2 unsigned int

Possible FAR address 2.

FSTATUS1 unsigned int

Possible fault status value 1.

FSTATUS2 unsigned int

Possible fault status value 2.

ArchMsg.Warning.unpred_async_tag_check_fail_on_sve_speculative_load_do_update_ffr

Unpredictable set to FALSE of the FFR element associated to SVE speculative load with an async tag check fail TAGS **UNPREDICTABLE**.

ArchMsg.Warning.unpredictable_a32_breakpoint

TAGS **UNPREDICTABLE** DISPLAY Unpredictable breakpoint on a A32 instruction at %{ADDR1} Offset %{OFFSET} BAS %{BAS} for a %{IS_ADDRESS_MISMATCH_BREAKPOINT:(Mismatch|Match)} Breakpoint number %{N:d} Breakpoint address %{ADDR2}. Fields:

ADDR1 unsigned int

Address.

ADDR2 unsigned int

Breakpoint address.

BAS unsigned int

BAS field.

IS_ADDRESS_MISMATCH_BREAKPOINT bool

Is a Mismatch breakpoint, not a Match breakpoint.

N unsigned int

Breakpoint number.

OFFSET unsigned int

Always has a value of zero.

ArchMsg.Warning.unpredictable_banked_register_access

TAGS **UNPREDICTABLE** DISPLAY Access to banked register (R=%{R:d} SYSm=%{SYSM:2x}) is unpredictable. Fields:

R unsigned int

R bit.

SYSM unsigned int

SYSm.

ArchMsg.Warning.unpredictable_ccfail_undef

TAGS **UNPREDICTABLE** DISPLAY An instruction %{OPCODE} at VA=%{VADDR} failed its condition codes check but would otherwise have caused a non-data-dependent trap or undefined exception. Behaviour is implementation-defined choice of exception or nop. Fields:

OPCODE unsigned int

opcode of the instruction.

VADDR unsigned int

Virtual address of the instruction being executed.

ArchMsg.Warning.unpredictable_ici_bits

TAGS **UNPREDICTABLE** DISPLAY ICI bits do not apply or are architecturally unpredictable for instruction.

ArchMsg.Warning.unpredictable_t32_breakpoint

TAGS **UNPREDICTABLE** DISPLAY Unpredictable breakpoint on a T32 instruction at %{ADDR1} Offset %{OFFSET} BAS %{BAS} for a %{IS_ADDRESS_MISMATCH_BREAKPOINT:(Mismatch|Match)} Breakpoint number %{N:d} Breakpoint address %{ADDR2}. Fields:

ADDR1 unsigned int

Address.

ADDR2 unsigned int

Breakpoint address.

BAS unsigned int

BAS field.

IS_ADDRESS_MISMATCH_BREAKPOINT bool

Is a Mismatch breakpoint, not a Match breakpoint.

N unsigned int

Breakpoint number.

OFFSET unsigned int

Indicates which halfword of a 32-bit Thumb instruction the breakpoint is set on. 0 for the first halfword or if the instruction is 16 bits wide, or 1 for the second halfword.

ArchMsg.Warning.unpredictable_t32_vector_catch

TAGS **UNPREDICTABLE** DISPLAY Unpredictable Vector catch exception on a T32 instruction at %{ADDR} Offset %{OFFSET}. Fields:

ADDR unsigned int

Address.

OFFSET unsigned int

Indicates which halfword of a 32-bit Thumb instruction the vector catch exception is set on. 0 for the first halfword or if the instruction is 16 bits wide, or 1 for the second halfword.

ArchMsg.Warning.unpredictable_watchpoint_far

TAGS **UNPREDICTABLE** DISPLAY The value for FAR/EDWAR for hitting this watchpoint may be anywhere in the range %{LOWER_BOUND} to %{UPPER_BOUND}. Fields:

LOWER_BOUND unsigned int

the lowest address accessed by the instruction that triggered the watchpoint.

UPPER_BOUND unsigned int

the highest watchpointed address accessed by that instruction.

ArchMsg.Warning.unpredictable_watchpoint_far_isscmp

DISPLAY The value for FAR/EDWAR for hitting this watchpoint may be anywhere in the range %{LOWER_BOUND} to %{UPPER_BOUND}. Fields:

LOWER_BOUND unsigned int

the lowest address accessed by the instruction that triggered the watchpoint.

UPPER_BOUND unsigned int

the highest watchpointed address accessed by that instruction.

ArchMsg.Warning.warning_AdvSIMDExpandImmUnexpectedZero

TAGS **UNPREDICTABLE** DISPLAY AdvSIMDExpandImm may treat this immediate value as **UNPREDICTABLE**.

ArchMsg.Warning.warning_ConditionalSMC

TAGS **UNPREDICTABLE** DISPLAY SMC instruction has **UNPREDICTABLE** effects when conditional and when combined with traps.

ArchMsg.Warning.warning_TLBID_out_of_range

TLBID out of range in TLBI according to the configuration of NIS, NVIS, NOS and NVOS. Fields:

NIS unsigned int

NIS.

NOS unsigned int

NOS.

NVIS unsigned int

NVIS.

NVOS unsigned int

NVOS.

TLBID unsigned int

The TLBID value used in the TLBI instruction.

ArchMsg.Warning.warning_TTL_or_TG_mismatches_current_translation_granule

TTL or TG mismatches current translation granule in TLBI VA/RVA/IPA/RIPA. Fields:

CURRENT_TG unsigned int

Current TG as indicated by TCR_ELx or VTCR_EL2.

TLBI_TG unsigned int

TG (or TTL[3:2]) used in TLBI instruction.

ArchMsg.Warning.warning_access_crosses_page_boundary

TAGS **UNPREDICTABLE** DISPLAY Access crosses page boundary.

ArchMsg.Warning.warning_access_crosses_page_boundary_has_fault

TAGS **UNPREDICTABLE** DISPLAY Access that crosses page boundary has fault(s).

ArchMsg.Warning.warning_access_crosses_page_boundary_has_fault_on_both_pages

TAGS **UNPREDICTABLE** DISPLAY Access that crosses page boundary has a fault on each page.
Fields:

ADDR1 unsigned int

FAR address on first page.

ADDR2 unsigned int

FAR address on second page.

FSTATUS1 unsigned int

Fault status on first page.

FSTATUS2 unsigned int

Fault status on second page.

ArchMsg.Warning.warning_access_crosses_page_boundary_spanning_different_memory

TAGS **UNPREDICTABLE** DISPLAY Access at address %{ADDR} crosses page from MEM_TYPE %{MEMTYPE_PAGE1} to %{MEMTYPE_PAGE2}. Fields:

ADDR unsigned int

address of access crossing page.

MEMTYPE_PAGE1 signed int

First Page's memory type (0-Normal, 1-Non-Normal).

MEMTYPE_PAGE2 signed int

Second Page's memory type (0-Normal, 1-Non-Normal).

ArchMsg.Warning.warning_access_wraps_around_memory

TAGS **UNPREDICTABLE** DISPLAY Access wraps around memory in %{WIDTH:d}bit mode.
Fields:

WIDTH unsigned int

Address width.

ArchMsg.Warning.warning_bcr_linking_status

TAGS **UNPREDICTABLE** DISPLAY DBGBCR.{BT2,BT} combination gives unpredictable behaviour.

ArchMsg.Warning.warning_bcr_mask_reserved

TAGS **UNPREDICTABLE** DISPLAY DBGBCR_ELn.MASK is set to a reserved value.

ArchMsg.Warning.warning_bvr_masked_bit_not_zero

TAGS **UNPREDICTABLE** DISPLAY DBGBVRn_EL1.VA contains a bit masked by DBGBCRn_EL1.MASK that is not zero.

ArchMsg.Warning.warning_ccsidr_unimplemented_level

TAGS **UNPREDICTABLE** DISPLAY CCSIDR read while CSSELR %{CSSELR:x} points to an unimplemented cache level. Fields:

CSSELR unsigned int

current effective value of CSSELR.

ArchMsg.Warning.warning_change_to_ns_when_tge_set

TAGS **UNPREDICTABLE** DISPLAY Attempting to change NS to 1 when HCR.TGE = 1.

ArchMsg.Warning.warning_contiguous_bit_check_abort

TAGS **UNPREDICTABLE** DISPLAY An abort occurred whilst attempting to check TLB entry is contiguous at %{ENTRY_ADDR}. Fields:

ENTRY_ADDR unsigned int

address of conflicting TLB entry.

ArchMsg.Warning.warning_contiguous_bit_error

TAGS **UNPREDICTABLE** DISPLAY TLB entry at %{ENTRY_ADDR} was not contiguous with entry at %{CONTIG_ADDR} Contents are %{ENTRY_DATA} but expected %{CONTIG_DATA}. Fields:

CONTIG_ADDR unsigned int

address of the TLB first read, with which this entry is expected to be contiguous.

CONTIG_DATA unsigned int

expected contents based on the entry at CONTIG_ADDR.

ENTRY_ADDR unsigned int

address of conflicting TLB entry.

ENTRY_DATA unsigned int

contents of conflicting TLB entry.

ArchMsg.Warning.warning_contiguous_bit_error_D128

TAGS **UNPREDICTABLE** DISPLAY TLB entry at %{ENTRY_ADDR} was not contiguous with entry at %{CONTIG_ADDR} Contents are %{ENTRY_DATA_HI} %{ENTRY_DATA_LO} but expected %{CONTIG_DATA}. Fields:

CONTIG_ADDR unsigned int

address of the TLB first read, with which this entry is expected to be contiguous.

CONTIG_DATA unsigned int

expected contents based on the entry at CONTIG_ADDR.

ENTRY_ADDR unsigned int

address of conflicting TLB entry.

ENTRY_DATA_HI unsigned int

Higher bit contents of conflicting TLB entry.

ENTRY_DATA_LO unsigned int

Lower bit contents of conflicting TLB entry.

ArchMsg.Warning.warning_cp10_cp11_mismatch

TAGS **UNPREDICTABLE** DISPLAY Unpredictable configuration of CP10 and CP11 controls at EL
%{EL:d}. Fields:

EL unsigned int

EL owning the control.

ArchMsg.Warning.warning_cp10_cp11_reserved_value

TAGS **UNPREDICTABLE** DISPLAY Unpredictable configuration of CPACR CP10 (%{CP10}) or
CP11 (%{CP11}). Fields:

CP10 unsigned int

CP10 access.

CP11 unsigned int

CP11 access.

ArchMsg.Warning.warning_csselr_level_out_of_range

TAGS **UNPREDICTABLE** DISPLAY CSSELR.Level out of range, written value %{LEVEL} with only
%{IMPLEMENTED:d} levels implemented. Fields:

IMPLEMENTED unsigned int

number of cache levels implemented.

LEVEL unsigned int

written value of CSSELR.Level.

TYPE enum

The type of cache selected by CSSELR.InD and TnD.

ArchMsg.Warning.warning_debug_flow_control_bits_not_obeyed

TAGS **UNPREDICTABLE** DISPLAY Flow control bits for %{REGISTER} were not obeyed, causing
%{ERROR}. Fields:

ERROR enum

type of error resulting.

REGISTER enum

register under consideration.

ArchMsg.Warning.warning_debug_register_access_during_reset

TAGS **UNPREDICTABLE** DISPLAY Unpredictable %{IS_WRITE:(read|write)} access to debug register offset %{OFFSET} during reset. Fields:

IS_WRITE unsigned int

Write Not Read.

OFFSET unsigned int

Register Offset.

ArchMsg.Warning.warning_decode_cps_inconsistent_fields

TAGS **UNPREDICTABLE** DISPLAY The mode setting fields in the CPS instruction are inconsistent.

ArchMsg.Warning.warning_decode_invalid_state

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable in current exception-level/security state.

ArchMsg.Warning.warning_default_cacheable_mmu_on

TAGS **UNPREDICTABLE** DISPLAY Default cacheable is enabled (HCR.DC = 1) while MMU is enabled.

ArchMsg.Warning.warning_default_cacheable_vmmu_off

TAGS **UNPREDICTABLE** DISPLAY Default cacheable is enabled (HCR.DC = 1) while HCR.VM = 0.

ArchMsg.Warning.warning_deprecated_wvr_bit_2_set

TAGS **UNPREDICTABLE** DISPLAY DBGWVR%{N:d}_EL1=%{WVR:h} has bit 2 set, deprecated in ARMv8. Fields:

N unsigned int

index.

WVR unsigned int

DBGWVR.

ArchMsg.Warning.warning_eret_while_software_step_active_pending

Missing ISB between setting MDSCR_EL1.SS and ERET TAGS **UNPREDICTABLE**.

ArchMsg.Warning.warning_exclusive_to_non_normal

TAGS **UNPREDICTABLE** DISPLAY Exclusive Access to Non-Normal Memory %{ADDR}. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_exclusive_to_non_writeback

TAGS **UNPREDICTABLE** DISPLAY Exclusive Access to Non-Writeback-Cacheable Memory %{ADDR}. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_execute_from_device_memory

TAGS **UNPREDICTABLE** DISPLAY An attempt was made to execute from Device memory.

ArchMsg.Warning.warning_illegal_cpsr_mode

TAGS **UNPREDICTABLE** DISPLAY Writing an illegal or unimplement mode ({NEW_MODE:h}) to CPSR was unpredictable. Fields:

NEW_MODE unsigned int

New value of CPSR.M.

ArchMsg.Warning.warning_illegal_srs_mode

TAGS **UNPREDICTABLE** DISPLAY Illegal or **UNPREDICTABLE** mode ({MODE:h}) used for SRS instruction. Fields:

MODE unsigned int

mode for Banked SP.

ArchMsg.Warning.warning_implementation_defined_read_debug_register_in_SCS

Read Access to Debug registers from the processor is **IMPLEMENTATION DEFINED** TAGS **IMP_DEF**. Fields:

PPB_OFFSET unsigned int

debug register offset.

ArchMsg.Warning.warning_implementation_defined_sequential_security_transitions_supported

The behavior of sequential instruction fetches that cross from non-secure to secure memory and contain an SG instruction is **CONSTRAINED UNPREDICTABLE** TAGS **UNPREDICTABLE**.

ArchMsg.Warning.warning_implementation_defined_stack_limit_check_supported

It is **IMPLEMENTATION DEFINED** whether stack pointer limit checking is performed for this instruction TAGS **IMP_DEF**. Fields:

ADDRESS unsigned int

Address of instruction.

OPCODE unsigned int

instruction opcode.

ArchMsg.Warning.warning_implementation_defined_taggednonsharedmemorytype

TAGS **IMP_DEF** DISPLAY Memory attribute has Tagged, non shared memory type; it is **IMPLEMENTATION DEFINED** whether the memory location is treated as Tagged or non-Tagged.

ArchMsg.Warning.warning_implementation_defined_write_debug_register_in_SCS

Write Access to Debug registers from the processor is **IMPLEMENTATION DEFINED** TAGS **IMP_DEF**. Fields:

DATA unsigned int

data attempted to be written.

PPB_OFFSET unsigned int

debug register offset.

ArchMsg.Warning.warning_invalid_tcr_granule

TAGS UNPREDICTABLE DISPLAY TCR.TG{%TG_ID:(0|1)} bits have been set to a granule size {%REQUEST:d}K not implemented - using {%SUBSTITUTE:d}K. Fields:

REQUEST signed int

page size requested (or 0 for reserved).

SUBSTITUTE signed int

best guess available page size.

TG_ID bool

bits TG0 or TG1.

ArchMsg.Warning.warning_load_multiple_user_registers_from_user_mode

TAGS UNPREDICTABLE DISPLAY An LDM(user registers) instruction executed from user mode.

ArchMsg.Warning.warning_mte_in_mem_attr_not_supported

TAGS UNPREDICTABLE DISPLAY Memory attribute {%ATTR} has Tagged memory type but Armv8.5-MTE is not implemented. Fields:

ATTR unsigned int

Memory attribute.

ArchMsg.Warning.warning_reserved_breakpoint_state_match

TAGS UNPREDICTABLE DISPLAY {%IS_BREAKPOINT:(Watchpoint|Breakpoint)} programmed with a reserved combination of HMC, SSC and {%IS_BREAKPOINT:(PAC|PMC)}. Fields:

IS_BREAKPOINT unsigned int

Is a breakpoint, not a watchpoint.

ArchMsg.Warning.warning_shareability

TAGS UNPREDICTABLE DISPLAY Unpredictable: combination of the 1st and 2nd stages of translation is Normal Inner Non-Cacheable, Outer Non-Cacheable.

ArchMsg.Warning.warning_software_step_set_while_enabled

TAGS UNPREDICTABLE DISPLAY MDSCR_EL1.SS set to 1 while software step debug exceptions are enabled.

ArchMsg.Warning.warning_thumb_instruction_wraps_around_memory

TAGS UNPREDICTABLE DISPLAY Thumb instruction wraps around memory.

ArchMsg.Warning.warning_ttbr_sbz_bits_are_not_zero

TAGS UNPREDICTABLE DISPLAY {%TTBR}[{%MSB:d}:0] should be zero, but are not. Fields:

MSB unsigned int

x-1.

TTBR enum

which TTBR register.

ArchMsg.Warning.warning_unaligned_address_dbgdtrrx_write

TAGS **UNPREDICTABLE** DISPLAY Unaligned address %{ADDR} in X0/R0 when DBGDTRRX written in memory access mode. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unaligned_address_dbgdtrtx_read

TAGS **UNPREDICTABLE** DISPLAY Unaligned address %{ADDR} in X0/R0 when DBGDTRTX read in memory access mode. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unaligned_to_device

TAGS **UNPREDICTABLE** DISPLAY Unaligned Access to Device Memory %{ADDR}. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unaligned_to_strongly_ordered

TAGS **UNPREDICTABLE** DISPLAY Unaligned Access to Strongly Ordered Memory %{ADDR}. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unknown_sau_rnr

TAGS **UNPREDICTABLE** DISPLAY SAU_RNR was set to an unsupported value.

ArchMsg.Warning.warning_unpredictable_AIRCR_PRIS_and_BFHFNMINs

TAGS **UNPREDICTABLE** DISPLAY The effect of setting both AIRCR.BFHFNMINs and AIRCR.PRIS to 1 is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_AIRCR_VECTCLRACTIVE_when_not_in_debug

TAGS **UNPREDICTABLE** DISPLAY The effect of writing a 1 to AIRCR.VECTCLRACTIVE if the processor is not halted in Debug state is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_AIRCR_VECTRESET_and_SYSRESETREQ

TAGS **UNPREDICTABLE** DISPLAY When the processor is halted in Debug state, if a write to the register writes a 1 to both VECTRESET and SYSRESETREQ, the behavior is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_AIRCR_VECTRESET_when_not_in_debug

TAGS **UNPREDICTABLE** DISPLAY The effect of writing a 1 to AIRCR.VECTRESET if the processor is not halted in Debug state is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_AIRCR_incorrect_VKEY

TAGS **UNPREDICTABLE** DISPLAY The value 0x05FA must be written to AIRCR.VKEY, otherwise the register write is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_EXC_RETURN_Reserved_Bit

EXC_RETURN[23:7] are reserved with the special condition that all bits should be written as one. Values other than all 1s are **UNPREDICTABLE**. TAGS **UNPREDICTABLE**. Fields:

EXC_RETURN unsigned int

Special PC value.

SBO_MASK unsigned int

Which bits Should Be One to not UNPRED.

ArchMsg.Warning.warning_unpredictable_PMEVTYPEPn_ELO_VS_reserved_value

TAGS **UNPREDICTABLE** DISPLAY Unpredictable reserved value of PMEVTYPEn<n>_ELO.VS.

ArchMsg.Warning.warning_unpredictable_change_to_DEMCR_MON_STEP_at_insufficient_priority

TAGS **UNPREDICTABLE** DISPLAY The effect of changing DEMCR.MON_STEP at an execution priority that is lower than the priority of the DebugMonitor exception is **CONSTRAINED UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_change_to_priority_of_active_exception

TAGS **UNPREDICTABLE** DISPLAY Changing the priority of an active exception is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_clear_lspact

TAGS **UNPREDICTABLE** DISPLAY **UNPREDICTABLE** state on context switch. Fields:

DRVEXC_TAKEN unsigned int

Derived exception taken, not pended.

REACHED_LAST_STORE unsigned int

Only the last faulted.

ArchMsg.Warning.warning_unpredictable_dbg_exit_unaligned_dlr

TAGS **UNPREDICTABLE** DISPLAY Debug exit to AArch32 with DLR[0] set to 1. PC[0] can be set to 0 or to DLR[0].

ArchMsg.Warning.warning_unpredictable_exception_catch

TAGS **UNPREDICTABLE** DISPLAY Generation of an exception catch event is unpredictable.

ArchMsg.Warning.warning_unpredictable_exception_return_inconsistent_state

TAGS **UNPREDICTABLE** DISPLAY state on exception return is unpredictable.

ArchMsg.Warning.warning_unpredictable_exception_return_instruction

TAGS **UNPREDICTABLE** DISPLAY exception return instruction is unpredictable.

ArchMsg.Warning.warning_unpredictable_in_debug_state

TAGS **UNPREDICTABLE** DISPLAY Instruction is **UNPREDICTABLE** when executed in debug state.

ArchMsg.Warning.warning_unpredictable_mair_encoding

TAGS **UNPREDICTABLE** DISPLAY Memory attribute %{ATTR} has invalid MAIR encoding; this may lead to unpredictable results. Fields:

ATTR unsigned int

Memory attribute.

ArchMsg.Warning.warning_unpredictable_pmu_counter_access

TAGS **UNPREDICTABLE** DISPLAY Access to PMU counters from non-secure EL0 or EL1 is **UNPREDICTABLE** with MDCR_EL2.HPMN set to 0.

ArchMsg.Warning.warning_unpredictable_prioritization_breakpoint_match_vector_catch

TAGS **UNPREDICTABLE** DISPLAY Two events with the same priority occurred at the same instruction: Address Matching Vector Catch debug event, and Breakpoint debug event. It is constrained unpredictable which is taken.

ArchMsg.Warning.warning_unpredictable_stack_selection

TAGS **UNPREDICTABLE** DISPLAY **UNPREDICTABLE** state on context switch.

ArchMsg.Warning.warning_unpredictable_tsize_out_of_range

TAGS **UNPREDICTABLE** DISPLAY %{TCR}.T%{TTBR:d}SZ (%{TSIZE:d} bits) is out of range. Must be between 25 and %{TSIZE_MAX:d}. Fields:

TCR enum

Which TCR.

TSIZE signed int

Value of TSize.

TSIZE_MAX signed int

The maximum allowed value.

TTBR unsigned int

Which TTBR region.

ArchMsg.Warning.warning_unpredictable_unaligned_pc_as_base_register

TAGS **UNPREDICTABLE** DISPLAY The PC value must be word-aligned, otherwise the behavior of the instruction is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_unaligned_pop_stack

TAGS **UNPREDICTABLE** DISPLAY **CONSTRAINED UNPREDICTABLE** unaligned stack frame %{FRAME_ADDR:x} in exception-return. Fields:

FRAME_ADDR unsigned int

unaligned stack frame address.

ArchMsg.Warning.warning_unpredictable_unaligned_to_non_writeback

TAGS **UNPREDICTABLE** DISPLAY Unaligned Access to Non-Writeback-Cacheable Memory.

ArchMsg.Warning.warning_unpredictable_vmsa_memattrib

TAGS **UNPREDICTABLE** DISPLAY Unpredictable vmsa memory attribute with texcb value programmed to %{TEXCB} when tex remap is %{TEX_REMAP}. Fields:

TEXCB unsigned int

texcb value.

TEX_REMAP bool

use tex remap.

ArchMsg.Warning.warning_unpredictable_vtcr_t0sz_sl0

TAGS **UNPREDICTABLE** DISPLAY The combination of VTCR.TOSZ=%{TOSZ:d} and VTCR.SLO=%{SLO:d} is unpredictable. Fields:

SLO signed int

VTCR.SLO, starting level for stage 2 translation table walks.

TOSZ signed int

VTCR.TOSZ, required input address range.

ArchMsg.Warning.warning_unpredictable_write_DHCSR

TAGS **UNPREDICTABLE** DISPLAY **UNPREDICTABLE** set of bit-changes in DHCSR. Fields:

NEW unsigned int

data attempted to be written.

OLD unsigned int

previous value.

ArchMsg.Warning.warning_unpredictable_write_SHCSR

TAGS **UNPREDICTABLE** DISPLAY **UNPREDICTABLE** setting a pending bit to 1 for an exception with priority >= execution priority.

ArchMsg.Warning.warning_unsupported_access_to_memory_mapped_register

TAGS **UNPREDICTABLE** DISPLAY Access to memory mapped register is unsupported.

ArchMsg.Warning.warning_user_jmcr_access

TAGS **UNPREDICTABLE** DISPLAY User %{WRITE:(write|read)} access to JMCR is unpredictable. Fields:

WRITE unsigned int

access is write.

ArchMsg.Warning.warning_wcr_mask_and_bas

TAGS **UNPREDICTABLE** DISPLAY DBGWCR_EL1.MASK is non-zero and BAS is not 0xff.

ArchMsg.Warning.warning_wcr_mask_reserved

TAGS **UNPREDICTABLE** DISPLAY DBGWCR_ELn.MASK is set to a reserved value.

ArchMsg.Warning.warning_wcr_non_configuous_bas

TAGS **UNPREDICTABLE** DISPLAY DBGWCR_EL1.BAS has non-contiguous set of ones.

ArchMsg.Warning.warning_wvr_masked_bit_not_zero

TAGS **UNPREDICTABLE** DISPLAY DBGWVRn_EL1.VA contains a bit masked by DBGWCRn_EL1.MASK that is not zero.

ArchMsg.Why.why_branch_target_exception

A branch target exception has occurred. DISPLAY branch target exception: target pc=%{PC}, target opcode=%{OPCODE}. PRIMARY KEY PC. Fields:

BTYPE unsigned int

pstate.BTYPE at time branch taken.

OPCODE unsigned int

opcode at branch target.

PC unsigned int

Virtual Address of branch target.

ArchMsg.Why.why_illegal_state

An illegal mode change has occurred. DISPLAY illegal mode change: %{MESSAGE}. Fields:

MESSAGE string

The message.

BRANCH_MISPREDICT

Simulating branch mispredict. Fields:

PC unsigned int

Origin address (or 0 if unavailable).

BRA_DIR

Direct branches, to immediate address. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISSET enum

The instruction set of the branch instruction.

IS_COND bool

Indicates if this is a conditional branch.

IS_HINTED bool

Whether the conditional instruction is hinted.

IS_ISB bool

Whether the instruction is ISB.

IS_LINK bool

Indicates if this is a branch with link.

PC unsigned int

The address of the branch instruction.

TARGET_ISET enum

The instruction set after the branch.

TARGET_PC unsigned int

The address the instruction branches to.

BRA_INDIR

Indirect branches, perhaps to a register. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISET enum

The instruction set of the branch instruction.

IS_COND bool

Indicates if this is a conditional branch.

IS_HINTED bool

Whether the conditional instruction is hinted.

IS_ISB bool

Whether the instruction is ISB.

IS_LINK bool

Indicates if this is a branch with link.

PC unsigned int

The address of the branch instruction.

TARGET_ISET enum

The instruction set after the branch.

TARGET_PC unsigned int

The address the instruction branches to.

BRA_INDIR_CCFAIL

Indirect branches fail, perhaps to a register.

BRA_RET_CCFAIL

Branch return fail, perhaps to a register.

CACHE_MAINTENANCE_OP

Cache Maintenance Operation. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

DATA unsigned int

Specified MVA or Set/way.

FUNCTION enum

Clean / Invalidate.

SCOPE enum

Affected region.

SIDE enum

Inst / Data.

CCFAIL

Conditional instruction condition check fail. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CCFAIL_UNC

Unconditional instruction condition check fail. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CCPASS

Conditional instruction condition check pass. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CCPASS_UNC

Unconditional instruction condition check pass. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CODE_CACHE_MAINT

Code cache maintenance. Fields:

END_INCL unsigned int

Inclusive end address for VA or PA requests.

NS bool

Non-secure world for PA.

START unsigned int

Start address for VA or PA requests.

TYPE enum

Request type.

COMPILE_BLOCK_END

Last instruction of basic block translated. Fields:

VADDR unsigned int

Address of next instruction after this basic block.

COMPILE_BLOCK_START

First instruction of basic block translated. Fields:

VADDR unsigned int

Address of first instruction in basic block.

COMPILE_INST

ARM instruction compiled. Fields:

DISASS string

Disassembly of the instruction.

ISSET enum

The instruction set of this instruction.

ITSTATE unsigned int

The ITSTATE current for the instruction.

OPCODE unsigned int

The opcode of the instruction.

PC unsigned int

The address of the instruction.

SIZE unsigned int

The size of the instruction in bytes.

CONTEXTIDR

Context ID Register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

NS bool

Secure or nonsecure banked register is accessed.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The new value written.

CONTEXT_SYNC

Called for every context synchronization event. Fields:

ARCH_CSE bool

Architectural Context Synchronization Event.

PC unsigned int

The address of the synchronization instruction.

SYSREG_SYNC bool

System register synchronization event.

CORE_ENDIAN

Core BE8 Big-Endian state changed. Fields:

BE8 bool

Core BE8 Big-Endian state.

CORE_INFO

Static processor attributes. Only triggered by a call to DumpState(). Fields:

ARCH_PROFILE enum

The architecture profile of the core.

CLUSTER_ID unsigned int

The cluster ID of this processor.

CORE_NUM unsigned int

The number of this core in an MP processor.

DETAILED_CORE_FEATURES enum

Detailed core features as a bitfield. Enum values are bitmasks to extract the info from the field.

FPU_VERSION enum

The VFP version implemented by the core.

MEM_ARCH enum

The memory architecture of the core.

MIDR_EL1 unsigned int

MIDR_EL1 value.

NUM_CORES unsigned int

The number of cores in this MP processor.

QUANTUM_SIZE unsigned int

The default quantum size of the core (in ticks).

SECURITY_FEATURES bool

Does the core have security features?.

CORE_LOADS

Processor load accesses. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ACQREL enum

Is this an acquire/release.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DATA unsigned int

The data read or written.

ELEMENT_SIZE unsigned int

Width of each element.

ENDIANNESS enum

Byte ordering of this access.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

LOCK enum

Normal, exclusive or locked access.

MECID unsigned int

Memory Encryption Context ID, if applicable.

MECID2 unsigned int

Memory Encryption Context ID of the second page, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PAGE_SIZE unsigned int

Page size as $\log_2(\text{size in bytes})$.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK, 2=Exclusive Failed, 3=Unsupported.

SIZE unsigned int

Width of the access in bytes. Only required if DATA is not traced.

TRANS bool

Is this a translated access.

VADDR unsigned int

The virtual address of the access.

CORE_REGS

Changes of the core registers R0 to R14. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number, 0 to 14.

MODE enum

Bank of the register accessed.

OLD_VALUE unsigned int

The old value overwritten.

PHYS_ID enum

The physical register accessed.

VALUE unsigned int

The new value written to the register.

CORE_REGS64

Changes of the core registers X0..X30, SP_ELn. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ID enum

The register number, 0-30 for X0-X30, >=32 for SP_ELn.

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

CORE_REGS64_READ

General purpose AArch64 register has been read X0..X30, SP_ELn. Fields:

ID enum

The register number, 0-30 for X0-X30, >=32 for SP_ELn.

VALUE unsigned int

The value read from the register.

CORE_STALLED_TRANSITION

Detect when a core is prevented from progressing PChannel states. Fields:

CURRENT_STATE enum

Current power mode of core.

REASON string

The reason a transition was stalled, and what must be done to unblock it.

TARGET_STATE enum

Desired power mode of core after transition.

CORE_STORES

Processor store accesses. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ACQREL enum

Is this an acquire/release.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DATA unsigned int

The data read or written.

ELEMENT_SIZE unsigned int

Width of each element.

ENDIANNESS enum

Byte ordering of this access.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

LOCK enum

Normal, exclusive or locked access.

MECID unsigned int

Memory Encryption Context ID, if applicable.

MECID2 unsigned int

Memory Encryption Context ID of the second page, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PAGE_SIZE unsigned int

Page size as $\log_2(\text{size in bytes})$.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK, 2=Exclusive Failed, 3=Unsupported.

SIZE unsigned int

Width of the access in bytes. Only required if DATA is not traced.

TRANS bool

Is this a translated access.

VADDR unsigned int

The virtual address of the access.

CP14_READ

System Coprocessor register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP14 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc1 unsigned int

opcode 1.

opc2 unsigned int

opcode 2.

CP14_WRITE

System Coprocessor register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP14 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc1 unsigned int

opcode 1.

opc2 unsigned int

opcode 2.

CP15_READ

System Coprocessor register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP15 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc1 unsigned int

opcode 1.

opc2 unsigned int

opcode 2.

CP15_READ64

System Coprocessor register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP15 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc unsigned int

opcode 1.

CP15_WRITE

System Control Coprocessor register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP15 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc1 unsigned int

opcode 1.

opc2 unsigned int

opcode 2.

CP15_WRITE64

System Coprocessor register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP15 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc unsigned int

opcode 1.

CPSR

CPSR change. (In AArch64 PSTATE represented in SPSR format). Fields:

CORE_NUM unsigned int

Core number in a multi processor.

OLD_VALUE unsigned int

The old CPSR value.

UNKNOWN unsigned int

Bits within the register that have unknown value.

VALUE unsigned int

The new CPSR value.

CRYPTO_SPEC

Every crypto instruction speculatively executed.

CorePowerStateChange

Core power-up state has changed. DISPLAY Core powered %{PU:(up|down)}. Fields:

PU bool

Whether the core powered up.

DATA_CACHE_SET_ALLOC_TAGS

Set allocation tags from DC GVA instruction. Fields:

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

NSDESC unsigned int

The security state of the access.

PADDR unsigned int

The physical address of the access.

PAS enum

The physical address space of the page.

RESPONSE enum

0=Aborted, 1=OK.

SIZE unsigned int

The size of memory in bytes.

TAG unsigned int

The value of the tag being set.

TAGGING_TYPE enum

Type of Memory tagging (Physical or Virtual).

TAG_VADDR unsigned int

The virtual address of the Virtual Tag if Virtual tagging is used; otherwise 0.

VADDR unsigned int

Virtual address used for setting allocation tags.

DATA_CACHE_ZERO

Zero and invalidate cache line from DC ZVA instruction. Fields:

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

NSDESC unsigned int

The security state of the access.

PADDR unsigned int

The physical address of the access.

PADDR_TAG unsigned int

If tag checked, the physical address tag for the access.

PAS enum

The physical address space of the page.

RESPONSE enum

0=Aborted, 1=OK.

SIZE unsigned int

The size of zero'd memory in bytes.

TAG_CHECKED bool

Is this access tag checked.

VADDR unsigned int

Virtual address of the zero'd cache line.

DEBUG_EVENT

Hardware debug support event. Fields:

EVENT enum

Description of event.

VALUE unsigned int

data value.

DEBUG_TARGET_ISET

Hardware debug target ISA. Fields:

TARGET_ISET enum

The target instruction set of the debug state.

DMI_ALLOCATE

DMI allocated. Fields:

ACCESS_DIRECTION enum

READ / WRITE.

PADDR unsigned int

The physical address of the DMI.

PTR unsigned int

DMI range pointer.

SIDE enum

DSIDE / ISIDE.

SIZE unsigned int

The size of DMI.

VADDR unsigned int

The virtual address of the DMI..

DMI_HIT

DMI hit, access used DMI (only fire for small DMI regions i.e. less than a full page). Fields:

ACCESS_DIRECTION enum

READ / WRITE.

PADDR unsigned int

The physical address of the DMI.

SIDE enum

DSIDE / ISIDE.

SIZE unsigned int

The size of the DMI.

VADDR unsigned int

The virtual address of the DMI..

DMI_REVOKE

DMI revoked. Fields:

ACCESS_DIRECTION enum

READ / WRITE.

PADDR unsigned int

The physical address of the DMI.

SIDE enum

DSIDE / ISIDE.

SIZE unsigned int

The size of DMI.

VADDR unsigned int

The virtual address of the DMI..

END_COMPILE

Compilation end. Fields:

END_OF_PAGE_COUNT unsigned int

Number of basic blocks exited due to page boundary since START_COMPILE.

FETCHFAIL_COUNT unsigned int

Number of basic blocks exited due to ifetch failure START_COMPILE.

INST_COUNT unsigned int

Number of instructions compiled since START_COMPILE.

NONSEQ_COUNT unsigned int

Number of basic blocks exited due to non-sequential instructions since START_COMPILE.

PAGE_STRADDLE_COUNT unsigned int

Number of basic blocks exited due to unaligned instructions crossing page since START_COMPILE.

ESB

Error Synchronization Barrier hit. Fields:

ASYNC bool

ESB is async.

IMPLICIT bool

ESB is implicit.

EXCEPTION

Exception taken. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ESR unsigned int

The value assigned to ESR when exception is taken.

ESR_EC unsigned int

The Value assigned to ESR.EC when exception is taken.

IS_PHYSICAL bool

Physical or Virtual exception.

IS_TRAP bool

Exception coming from a trap.

LR unsigned int

The value assigned to the link register.

PC unsigned int

The location where the exception occurred.

PREFERRED_RETURN unsigned int

The preferred return address for the exception.

TAKEN_LOCALLY bool

Whether exception is taken locally(same exception level).

TARGET_ISET enum

The instruction set of the exception handler code.

TARGET_PC unsigned int

The address the exception branches to.

VECTOR enum

The exception vector.

VECTOR_OFFSET unsigned int

The offset of an exception vector from the base address.

EXCEPTION_END

Every exception completed.

EXCEPTION_INFO

Trace human-readable reason of the current exception for data/instruction aborts, floating point exceptions and unknown exceptions. Fields:

DESCRIPTION string

Reason for current exception.

ESR_EC unsigned int

The value of ESR.EC when exception is taken.

EXCEPTION_RAISE

Every exception raised.

EXCEPTION_RETURN

Branches on leaving exception. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISET enum

The instruction set of the branch instruction.

IS_COND bool

Indicates if this is a conditional branch.

IS_HINTED bool

Whether the conditional instruction is hinted.

IS_ISB bool

Whether the instruction is ISB.

IS_LINK bool

Indicates if this is a branch with link.

PC unsigned int

The address of the branch instruction.

TARGET_ISET enum

The instruction set after the branch.

TARGET_PC unsigned int

The address the instruction branches to.

EXCEPTION_RETURN_PREBRANCH

Exception return event that must occur before PC and processor state have updated.

EXCEPTION_START

Every exception started.

FIQ_TAKEN

FIQ taken exception.

FP_EXCEPTIONS_SIGNALLED

Trace which FP exceptions were signalled as a part of the FP operations performed by an instruction. Fields:

FP_EXCEPTIONS_SIGNALLED_MASK unsigned int

Indicates which FP exceptions were signalled by an instruction. This uses the same encoding as the cumulative floating-point exception bitfields in FPSR.

FP_STATE

Floating point state. Fields:

FPSR unsigned int

FPSR register value.

MASK unsigned int

The mask indicating updated cumulative bits.

FREQ_CHANGED

Core frequency changed. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The instruction count of this core.

NEW_FREQ signed int

The new frequency of this core (expressed in Hz).

OLD_FREQ signed int

The old frequency of this core (expressed in Hz).

HLT

HLT instruction occurred. Fields:

IMM16 unsigned int

The 16-bit immediate value encoded in HLT instruction.

INST

Every instruction executed. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DEBUG_STATE bool

The instruction is executed in debug state.

DISASS string

Disassembly of instruction.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISSET enum

The current instruction set.

ITSTATE unsigned int

The current ITSTATE.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

MODE enum

The mode the core is in.

NS unsigned int

The core's non-secure bit.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

OPCODE unsigned int

The opcode of the instruction.

PADDR unsigned int

The physical address of the instruction.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the instruction.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the page.

PC unsigned int

The address of the instruction.

SECURITY_STATE enum

The core's security state.

SIZE unsigned int

The size of the instruction in bytes.

INST_END

Every instruction completed.

INST_START

Every instruction started. Fields:

DEBUG_STATE bool

The instruction is executed in debug state.

ISET enum

The current instruction set.

MODE enum

The mode the core is in.

NS enum

The current Secure State.

PC unsigned int

The address of the conditional instruction.

SECURITY_STATE enum

The current Security State.

INST_STRADDLE

Instruction straddles boundary. Fields:

DEBUG_STATE bool

The instruction is executed in debug state.

ISET enum

The current instruction set.

MODE enum

The mode the core is in.

NS enum

The current Secure State.

PC unsigned int

The address of the conditional instruction.

SECURITY_STATE enum

The current Security State.

IRQ_TAKEN

IRQ taken exception.

ISB

ISB instruction occurred. Fields:

PC unsigned int

The address of the ISB instruction.

LOAD_MULTIPLE_REGS_LIST

List of destination registers (not source) of a multiple registers load instruction. Fields:

REG string

Register to add into the list.

LOCAL_MONITOR

Local monitor activity. Fields:

PADDR unsigned int

Local Monitor Address.

State enum

State of the monitor (Open/Exclusive).

MAPPING_CREATE

A memory mapping has been created. This is an implementation detail of the model and so may change without notice. Fields:

ASID unsigned int

The ASID used during translation, or 0 if no ASID was used.

IS_VALID bool

Whether the mapping is valid or not.

PADDR unsigned int

Base physical address of region, or 0 if not known.

SIDE enum

I-side or D-side.

SIZE unsigned int

Size of the page, or 0 if not known.

VADDR unsigned int

Base virtual address of region.

VMID unsigned int

The VMID used during translation, or 0 if no VMID was used.

MAPPING_DESTROY

A memory mapping has been destroyed. This is an implementation detail of the model and so may change without notice. Fields:

ASID unsigned int

The ASID used during translation.

CAUSE enum

The cause of invalidation.

PADDR unsigned int

Base physical address of page.

SIDE enum

I-side or D-side.

SIZE unsigned int

Size of the page.

VADDR unsigned int

Base virtual address of page.

VMID unsigned int

The VMID used during translation.

MEMMAP_DEBUG_READ

Memory mapped reads to the debug registers. Fields:

ADDR unsigned int

address.

CORE_NUM unsigned int

Core number in a multi processor.

EXT bool

Whether access is from an external device (such as the DAP).

NS enum

Secure state of the access.

REG_NAME string

Name of the debug register accessed.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

MEMMAP_DEBUG_WRITE

Memory mapped writes to the debug registers. Fields:

ADDR **unsigned int**

address.

CORE_NUM **unsigned int**

Core number in a multi processor.

EXT **bool**

Whether access is from an external device (such as the DAP).

NS **enum**

Secure state of the access.

REG_NAME **string**

Name of the debug register accessed.

UNDEF **bool**

The register accessed is undefined.

VALUE **unsigned int**

The new value written.

MMU_D128_TTB_READ

This event is triggered by reads caused by a D128 translation table walk. Fields:

ABORTED **bool**

Was the walk successful.

DATA_HIGH **unsigned int**

The data upper bits read.

DATA_LOW **unsigned int**

The data lower bits read.

IPA **unsigned int**

For stage 1, the IPA of the read.

LEVEL **unsigned int**

Translation table level.

MECID **unsigned int**

Memory Encryption Context(MEC) ID, if applicable.

MPAM_PARTID **unsigned int**

MPAM Partition ID.

MPAM_PMG **unsigned int**

MPAM Performance Monitoring Group.

MPAM_SP **unsigned int**

MPAM Partition ID Space.

OUTADDR unsigned int

The output physical address.

PADDR unsigned int

The physical address of the read.

PASmem enum

Physical Address Space of the accessed memory.

PASreq enum

Physical Address Space of the requested lookup.

REGIME_EL enum

Entry matches in this translation regime.

REQUESTER enum

The requester of this table walk.

SIDE enum

Inst / Data.

STAGE unsigned int

Translation stage.

MMU_D128_TTB_WRITE

This event is triggered by writes caused by a D128 translation table walk. Fields:

ABORTED bool

Was the walk successful.

DATA_HIGH unsigned int

The data upper bits written.

DATA_LOW unsigned int

The data lower bits written.

IPA unsigned int

For stage 1, the IPA of the write.

IS_HACDBS_ACCESS bool

If the access was due to HACDBS_CLEAN.

LEVEL unsigned int

Translation table level.

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

OUTADDR unsigned int

The output physical address.

PADDR unsigned int

The physical address of the write.

PASmem enum

Physical Address Space of the accessed memory.

PASreq enum

Physical Address Space of the requested lookup.

REGIME_EL enum

Entry matches in this translation regime.

SIDE enum

Inst / Data.

STAGE unsigned int

Translation stage.

MMU_TRANS

Address translation information. Fields:

ASID unsigned int

Address space identifier.

CORE_NUM unsigned int

Core number in a multi processor.

Hyp bool

Entry matches in Hyp state only.

INNERCACHE_RA bool

Is the inner cache allocate on read.

INNERCACHE_TRANSIENT bool

Is the inner write-through transient.

INNERCACHE_TYPE enum

Inner Caching scheme (NC/MB/WA).

INNERCACHE_WA bool

Is the inner cache allocate on write.

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MEMTYPE enum

Memory type.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC enum

Is secure side supposed to access secure or nonsecure memory.

OUTERCACHE_RA bool

Is the outer cache allocate on read.

OUTERCACHE_TRANSIENT bool

Is the outer write-through transient.

OUTERCACHE_TYPE enum

Outer Caching scheme (NC/MB/WA).

OUTERCACHE_WA bool

Is the outer cache allocate on write.

PADDR unsigned int

Physical address of the access.

PAGESIZE unsigned int

Page size as log2(size).

PAS enum

PAS supposed to be accessed is secure or nonsecure memory or Root/Realm if RME is implemented.

SH enum

Shareability.

SIDE enum

Inst / Data.

STAGE1_PERM unsigned int

Stage 1 permissions, [5:3] Privileged access (XWR) [2:0] User access (XWR).

STAGE2_PERM unsigned int

Stage 2 permission mask [5:3] Privileged access (XWR) [2:0] User access (XWR).

VADDR unsigned int

Virtual address of the access.

VMID unsigned int

Virtual machine identifier.

XS bool

XS Attribute.

nG enum

Flag indicating whether ASID will be matched.

MMU_TTB_READ

This event is triggered by reads caused by a translation table walk. Fields:

ABORTED bool

Was the walk successful.

DATA unsigned int

The data read.

DESC_KIND enum

The encoding scheme of the descriptors used for this walk.

IPA unsigned int

For stage 1, the IPA of the read.

LEVEL unsigned int

Translation table level.

LPAE bool

Is this for an LPAE translation (*DEPRECATED*, use DESC_KIND).

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSmem bool

Non secure memory access.

NSreq bool

Non secure request.

OUTADDR unsigned int

The output physical address.

PADDR unsigned int

The physical address of the read.

PASmem enum

Physical Address Space of the accessed memory.

PASreq enum

Physical Address Space of the requested lookup.

REGIME_EL enum

Entry matches in this translation regime.

REQUESTER enum

The requester of this table walk.

SIDE enum

Inst / Data.

STAGE unsigned int

Translation stage.

MMU_TTB_WRITE

This event is triggered by writes caused by a translation table walk. Fields:

ABORTED bool

Was the walk successful.

DATA unsigned int

The data written.

DESC_KIND enum

The encoding scheme of the descriptors used for this walk.

IPA unsigned int

For stage 1, the IPA of the write.

IS_HACDBS_ACCESS bool

If the access was due to HACDBS_CLEAN.

LEVEL unsigned int

Translation table level.

LPAE bool

Is this for an LPAE translation (*DEPRECATED*, use DESC_KIND).

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSmem bool

Non secure memory access.

NSreq bool

Non secure request.

OUTADDR unsigned int

The output physical address.

PADDR unsigned int

The physical address of the write.

PASmem enum

Physical Address Space of the accessed memory.

PASreq enum

Physical Address Space of the requested lookup.

REGIME_EL enum

Entry matches in this translation regime.

SIDE enum

Inst / Data.

STAGE unsigned int

Translation stage.

MODE_CHANGE

Mode change. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

MODE enum

The new mode.

NON_SECURE enum

The core's new non-secure bit.

OLD_MODE enum

The old mode.

SECURITY_STATE enum

The new security state.

NOT_TAG_CHECKED_INST

MTE Tag not checked for current instruction.

PERIODIC

Called for every quantum. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The instruction count of this CPU.

PC unsigned int

The address of the next instruction to be executed on this CPU.

PMU_COUNTER_OVERFLOW

PMU counter overflow. Fields:

COUNTER_GROUP enum

Counter group.

COUNTER_RANGE enum

Counter range.

EVENT_ID unsigned int

Event Identifier, field is valid only for Event counter group.

INDEX unsigned int

Counter index (as selected by PMSELR), field is valid only for Event/Cycle counter groups.

INTERRUPT bool

Is interrupt enabled on overflow for this counter.

POTINDEX_UPDATE

Value of POT index for current pc changed. Fields:

NEW_POTINDEX unsigned int

New POTIndex.

OLD_POTINDEX unsigned int

Old POTIndex.

PREFETCH_MEMORY64

Prefetch from PRFM or PRFUM instructions. Fields:

PRFOP enum

Prefetch hint.

VADDR unsigned int

Virtual address of the location that should be prefetched.

PRELOAD64_REQUEST

Preload requests from PRFM or PRFUM instructions. Fields:

PRFOP enum

Prefetch hint.

VADDR unsigned int

Virtual address of the location that should be prefetched.

PRELOAD_DATA

Data preload from PLD instruction. Fields:

VADDR unsigned int

Virtual address of the data that should be preloaded.

PRELOAD_INST

Instruction preload from PLI instruction. Fields:

VADDR unsigned int

Virtual address of the instruction that should be preloaded.

PRE_CORE_LOAD

Trace just before a core load. Fields:

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

PRE_CORE_STORE

Trace just before a core store. Fields:

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

PRE_TTB_READ

This event is triggered before reads caused by a translation table walk. Fields:

ADDR unsigned int

The physical address of the read.

PSTATE

PSTATE fields update. Fields:

ALLINT unsigned int

All interrupt mask bit.

BTYPE unsigned int

Branch target identification bit.

CORE_NUM unsigned int

Core number in a multi processor.

DAIF unsigned int

Exception mask bits.

DIT unsigned int

Data Independent Timing bit.

EL unsigned int

Current Exception level field.

EXLOCK unsigned int

Exception return state lock bit.

IL unsigned int

Illegal Execution state bit.

NZCV unsigned int

Condition flags.

PACM unsigned int

PACM bit.

PAN unsigned int

Privileged Access Never state bit.

PM unsigned int

PMU exception mask bit.

PPEND unsigned int

PMU exception pending bit.

SM unsigned int

StreamingMode status bit.

SP unsigned int

Stack pointer register selection bit.

SS unsigned int

Software Step bit.

SSBS unsigned int

Speculative Store Bypass Safe bit.

TCO unsigned int

Tag Check Override bit.

TOUCHED unsigned int

Mask of the fields that have been touched since this trace has last been emitted. Note that touched does not necessarily mean the value changed. The index of the field is the bit index in the mask plus the index of this field plus one.

UAO unsigned int

User Access Override bit.

UINJ unsigned int

Undefined Instruction exceptions bit.

ZA unsigned int

ZA storage and ZT0 register enable bit.

nRW unsigned int

Current Execution state bit.

PSTATE_PACTIVE_OPMODE_TRANSITION

Trace pactive and Opmode mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Operating mode after the transition.

OLD_STATE enum

Operating mode before the transition.

PSTATE_PACTIVE_OPMODE_TRANSITION_DENIED

Trace the denied pactive and Opmode mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Operating mode after the transition.

OLD_STATE enum

Operating mode before the transition.

REASON string

Reason for the transition being denied.

PSTATE_PACTIVE_POWER_TRANSITION

Trace pactive and power mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Power mode after the transition.

OLD_STATE enum

Power mode before the transition.

PSTATE_PACTIVE_POWER_TRANSITION_DENIED

Trace the denied pactive and power mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Power mode after the transition.

OLD_STATE enum

Power mode before the transition.

REASON string

Reason for the transition being denied.

PSTATE_PACTIVE_THREAD_OPMODE_TRANSITION

Trace pactive and Opmode mode transitions for threads on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Operating mode after the transition.

OLD_STATE enum

Operating mode before the transition.

QUANTUM_START

Called at the start of every quantum.

RANGE_PRELOAD_REQUEST

Preload requests from RPRFM. Fields:

COUNT unsigned int

Number of blocks of data to access.

LENGTH unsigned int

Number of contiguous bytes to be accessed in each block.

OP unsigned int

Range prefetch operation .

REUSE unsigned int

Maximum number of bytes to be accessed before next RPRFM .

STRIDE signed int

Stride to advance after accessing LENGTH bytes.

VADDR unsigned int

Base virtual address of range prefetch request.

RUN_STATE

Run state transition. Fields:

INST_COUNT unsigned int

Ticks count at point of transition.

NEW enum

New run state.

OLD enum

Old run state.

SEMIHOSTING_CALL

Call of a semihost function occurred. Fields:

PC unsigned int

The program counter after the semihosting call.

SEMIHOSTING_PRECALL

About to call semihost. Fields:

EL2 bool

Use EL2 translation regime.

NS bool

Is Non-Secure.

REG_WIDTH unsigned int

The current register width in bytes.

SIGNAL

External signal state change. Fields:

SIGNAL enum

Signal that changed.

STATE bool

Signal asserted state.

SOFTWARE_STEP

Return the Debug Software Step state. Fields:

STATE enum

Software Step state.

SPSR

SPSR change. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

MODE enum

Which of the banked SPSR registers is written.

OLD_VALUE unsigned int

The old SPSR value.

VALUE unsigned int

The new SPSR value.

START_COMPILE

Compilation started. Fields:

VADDR unsigned int

Instruction where compilation begins.

STG_MULTIPLE_MICRO_OPS_ADDRESS_RANGE

Address range for MTE stores with multiple uops. Fields:

ADDR_END unsigned int

Final address.

ADDR_START unsigned int

Starting address.

STORE_MULTIPLE_REGS_ADDRESS_RANGE

Address range for stores with multiple registers. Fields:

ADDR_END unsigned int

Final address.

ADDR_START unsigned int

Starting address.

SUBOPTIMAL_LDST_RETIRED

suboptimal load/store. Fields:

ACCESS_SIZE unsigned int

Log2 of the access width in bytes used for alignment checking (1: half-word, 2: word ...).

OPTIMAL_ALIGN_SELECTOR unsigned int

Selector to decide the optimal alignment size (1: access width, 2: 4-bytes, 3: 8-bytes, ..., 12: 4-kbytes).

VADDR unsigned int

The virtual address of the access.

SYNC

Called for every synchronization, that is every quantum break. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The instruction count of this CPU.

LOCAL_QUANTUM unsigned int

The local quantum of this CPU.

LOCAL_TIME unsigned int

The local time of this CPU.

SYNC_REASON enum

Reason for sync.

SYSCALL

System call instruction executed. Fields:

IMM unsigned int

Immediate value of the system call instruction.

TYPE enum

System call type.

VADDR unsigned int

Instruction that caused the system call.

SYSREG_READ128

System Coprocessor 128 bit register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

REGNUM enum

Internal register number.

REGNUM_PHYS enum

Internal register number physically accessed (e.g. in presense of VHE redirection).

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_READ64

System Coprocessor register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REGNUM enum

Internal register number.

REGNUM_OPERAND enum

Register number corresponding to the instruction operand.

REGNUM_PHYS enum

Internal register number physically accessed (e.g. in presense of VHE redirection).

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_UPDATE128

Triggers when the system updates a register. Fields:

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG enum

Register number.

UNKNOWN unsigned int

Bits of the register which became unknown.

VALUE unsigned int

Value written to the register.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_UPDATE32

Triggers when the system updates a register. Fields:

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG enum

Register number.

UNKNOWN unsigned int

Bits of the register which became unknown.

VALUE unsigned int

Value written to the register.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_UPDATE64

Triggers when the system updates a register. Fields:

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG enum

Register number.

UNKNOWN unsigned int

Bits of the register which became unknown.

VALUE unsigned int

Value written to the register.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_WRITE128

System Coprocessor 128 bit register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

REGNUM enum

Internal register number.

REGNUM_PHYS enum

Internal register number physically accessed (e.g. in presense of VHE redirection).

UNDEF bool

The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_WRITE64

System Coprocessor register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REGNUM enum

Internal register number.

REGNUM_OPERAND enum

Register number corresponding to the instruction operand.

REGNUM_PHYS enum

Internal register number physically accessed (e.g. in presense of VHE redirection).

UNDEF bool

The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

UNALIGNED_LDST_RETIRED

Processor unaligned load/store. Fields:

ACCESS_SIZE unsigned int

Log2 of the access width in bytes used for alignment checking (1=>half-word, 2->word ...).

NUMBER_OF_BEATS unsigned int

Number of accesses (beats) in this burst (Total size in bytes is ACCESS_SIZE * NUMBER_OF_BEATS).

VADDR unsigned int

The virtual address of the access.

VFP_D_REGS

VFP/NEON D 64 bit register write. Fields:

ALIASING enum

Type of register aliasing used.

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a write to the corresponding bit in the register. For example, 0x1 would indicate a write to VALUE[0].

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

VFP_D_REGS_READ

VFP/NEON D 64 bit register read. Fields:

ID unsigned int

The register number.

VALUE unsigned int

The current value of the register.

VFP_Q_REGS

VFP/NEON Q 128 bit register write. Fields:

ALIASING enum

Type of register aliasing used.

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a write to the corresponding byte in the register. For example, 0x1 would indicate a write to VALUE[7:0].

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

VFP_Q_REGS_READ

VFP/NEON Q 128 bit register read. Fields:

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a read to the corresponding byte in the register. For example, 0x1 would indicate a read to VALUE[7:0].

VALUE unsigned int

The current value of the register.

VFP_SYS_REGS

Writes to the VFP/NEON units system registers. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ID enum

Which VFP system register is written.

MASK unsigned int

The mask indicating updated cumulative bits.

OLD_VALUE unsigned int

The register's old value overwritten.

VALUE unsigned int

The new value written to the VFP system register.

VFP_S_REGS

VFP/NEON S 32 bit register write. Fields:

ALIASING enum

Type of register aliasing used.

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a write to the corresponding bit in the register. For example, 0x1 would indicate a write to VALUE[0].

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

VFP_S_REGS_READ

VFP/NEON S 32 bit register read. Fields:

ID unsigned int

The register number.

VALUE unsigned int

The current value of the register.

VIRTUAL_TAG_ALLOCATED

Virtual tag information when allocated. Fields:

TAG_VADDR unsigned int

Virtual tag virtual address.

VADDR unsigned int

Data Virtual address.

VIRTUAL_TAG unsigned int

Virtual tag value.

WAYPOINT

Signals end of basic block execution. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ISET enum

Origin instruction set.

IS_COND bool

Indicates if this is a conditional waypoint.

PC unsigned int

Origin address (or 0 if unavailable).

TAKEN bool

Indicates if this waypoint was taken.

TARGET unsigned int

Destination address.

TARGET_ISET enum

Destination instruction set.

WFE_END

WFE ended. Fields:

INST_COUNT unsigned int

Ticks count when leaving WFE.

WFE_EVENT_REGISTER

WFE event register status: set/clear, reason. Fields:

INST_COUNT unsigned int

Ticks count.

REASON enum

Reason for set/clear. Only REASON==0 (Cleared by WFE) clears the bit, all other reasons set it.

WFE_IGNORED

WFE ignored. Fields:

EVENT bool

This WFE was ignored because the event register was set.

INST_COUNT unsigned int

Ticks count when ignoring WFE.

TRAPPED bool

This WFE was trapped.

WFE_START

WFE entered. Fields:

INST_COUNT unsigned int

Ticks count when entering WFE.

WFI_END

WFI ended. Fields:

INST_COUNT unsigned int

Ticks count when leaving WFI.

WFI_IGNORED

WFI ignored. Fields:

DISABLED bool

This WFI was ignored because WFI is disabled.

INST_COUNT unsigned int

Ticks count when ignoring WFI.

REASON enum

specifies reason why WFI trace was ignored.

TRAPPED bool

This WFI was trapped.

WFI_START

WFI entered. Fields:

INST_COUNT unsigned int

Ticks count when entering WFI.

WFI_WAKEUP

WFI wakeup. Fields:

INST_COUNT unsigned int

Ticks count when WFI wakeup occurred.

REASON enum

Reason for wakeup.

2.27 ARM_Cortex-A720

This section describes the trace sources.

AA64_ASE_SVE_REGS

Modification of AArch64 Advanced SIMD or SVE Vector Registers (via SIMD or Scalar operations), SVE Predicate Registers and ZT0. Fields:

ID **enum**

The register identifier.

MASK **unsigned int**

Bitmask of the register to signal the modified bits in the VALUE field.

SM **bool**

Whether the PE is in Streaming Mode.

VALUE **unsigned int**

Value of AArch64 vector register.

ASYNC_MEMORY_FAULT

Context ID Register write. Fields:

FAULT **unsigned int**

Fault status present in ESR.ISS encoding for Serror in ARCH64 and present in DFSR encoding for external abort format in ARCH32.

PADDR **unsigned int**

Physical Address (or 0 if unavailable).

VADDR **signed int**

Virtual Address (or 0 if unavailable).

ATOMIC_END_ACCESS

Bus trace access for atomics. Fields:

ACCESS_FAIL **bool**

Memory access failed.

ADDR **unsigned int**

The virtual address of the access.

ATTR **unsigned int**

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

COMPARE_VALUE **unsigned int**

Compare value for CAS.

LOAD_VALUE unsigned int

Loaded values.

NSDESC unsigned int

The security state of the access.

OPERAND_VALUE unsigned int

Operation's operand.

OPERATION enum

Operation type.

PADDR unsigned int

The physical address of the access.

PRIV bool

Is this a privileged access?.

ATOMIC_START_ACCESS

Bus trace access for atomics. Fields:

ADDR unsigned int

The virtual address of the access.

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

COMPARE_VALUE unsigned int

Compare value for CAS.

NSDESC unsigned int

The security state of the access.

OPERAND_VALUE unsigned int

Operation's operand.

OPERATION enum

Operation type.

PADDR unsigned int

The physical address of the access.

PRIV bool

Is this a privileged access?.

ArchMsg.Error.exit_code

DISPLAY exit code: %{EXITCODE} component kind: %{KIND} reason: %{REASON}. Fields:

EXITCODE signed int

Exit code to be returned.

KIND string

Component kind that invoked the exit code trace.

REASON string

Reason why the exit code trace was invoked (optional).

ArchMsg.Warning.IgnoredWarmResetRequest

Event fires when software attempts to request warm reset through write to deprecated DBGPRCR.CWRR.

ArchMsg.Warning.RAS_ErrselrUnpredictableState

DISPLAY ERRSELR is in unpredictable state due to value greater than number_of_error_records written to it.

ArchMsg.Warning.RAS_UnpredictableStateWrite

DISPLAY %{REG_NAME} write ignored due to unpredictable ERRSELR state. Fields:

REG_NAME string

Name of register being written.

ArchMsg.Warning.ReportedRevisionRegisterRead

DISPLAY Reported revision and patch do not match that of the cpu. Fields:

ORIGINAL_VALUE unsigned int

The value the register would have based on the modelled revision.

REGISTER string

The name of the register that has ben read.

REPORTED_VALUE unsigned int

The value the register has been overridden to.

ArchMsg.Warning.Unpredictably Indexed PM Event Register

TAGS UNPREDICTABLE DISPLAY Event %{IsDirect:(Selector value|Register)} %{SEL} is >= %{N}. Fields:

IsDirect unsigned int

Direct Access rather than indirect.

N unsigned int

Number of Accessible Registers.

SEL unsigned int

Selector.

ArchMsg.Warning.cache_contents_unknown

A cache has been activated without being invalidated and may contain **UNKNOWN** content. DISPLAY %{SIDE:(D|I)}-Cache was enabled in %{REGIME:(SECURE|NON_SECURE|HYP)} regime but was not invalidated since power-on: cache lines could contain **UNKNOWN** content %{SIDE:(Note: D-side caches must invalidate SECURE lines even if the cache is only used by NON_SECURE code in order to ensure that unexpected dirty lines tagged SECURE are not naturally evicted)}. Fields:

REGIME unsigned int

Current translation regime.

SIDE unsigned int

0 d-side cache or unified cache, 1 i-side.

ArchMsg.Warning.dcimvac_matches_watchpoint

TAGS **UNPREDICTABLE** DISPLAY Watchpoints matching an AArch32 DCIMVA are implementation defined.

ArchMsg.Warning.decode_fieldmismatch

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with different fields %{FIELD1} and %{FIELD2}. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_fracbitsnegative

TAGS **UNPREDICTABLE** DISPLAY VCVT Instruction is unpredictable with a negative number of fraction bits.

ArchMsg.Warning.decode_initblock

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable in an IT block.

ArchMsg.Warning.decode_initblocknotlast

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable in an IT block when not the last.

ArchMsg.Warning.decode_invalid_condition

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable when the condition is not AL.

ArchMsg.Warning.decode_invalidfieldcombination

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with the values in %{FIELD1} and %{FIELD2}. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_invalidvalue

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with the value in %{FIELD}. Fields:

FIELD enum

Field name.

ArchMsg.Warning.decode_mem_hint_unallocated

TAGS **UNPREDICTABLE** DISPLAY There is no memory hint allocated to this bit pattern.

ArchMsg.Warning.decode_registermatch

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with %{FIELD1} and %{FIELD2} as the same register. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_registermismatch

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with %{FIELD1} and %{FIELD2} as different registers. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_registeroutofrange

TAGS **UNPREDICTABLE** DISPLAY Floating-point instruction is unpredictable indexing beyond the end of the register bank.

ArchMsg.Warning.decode_sbzsbo

TAGS **UNPREDICTABLE** DISPLAY Reserved bits in the instruction are not canonical.

ArchMsg.Warning.decode_transactiontoobig

TAGS **UNPREDICTABLE** DISPLAY Attempt to perform a memory transaction of more than 128 bytes.

ArchMsg.Warning.decode_unpred_other

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable.

ArchMsg.Warning.decode_unpreduseofpc

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable when %{FIELD} is PC. Fields:

FIELD enum

Field name.

ArchMsg.Warning.decode_unpreduseofr13

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable when %{FIELD} is R13. Fields:

FIELD enum

Field name.

ArchMsg.Warning.decode_unpreduseofr13inreglist

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with R13 in register-list.

ArchMsg.Warning.decode_useofd16orhigher

TAGS **UNPREDICTABLE** DISPLAY Attempt to use register D16 or higher where it is not allowed.

ArchMsg.Warning.decode_writebackandbaseinlist

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with write back when the base register is in the transfer list.

ArchMsg.Warning.decode_zeroregistersinlist

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable when no registers are to be transferred.

ArchMsg.Warning.misaligned_ptrs

DISPLAY TRBPTR_EL1(%{TRBPTR_EL1}) or TRBTRG_EL1(%{TRBTRG_EL1}) are not aligned as per TRBIDR_EL1.Align(%{TRBIDR_EL1}). Fields:

TRBIDR_EL1 unsigned int

TRBIDR_EL1.Align value.

TRBPTR_EL1 unsigned int

Current write pointer.

TRBTRG_EL1 unsigned int

Trigger pointer.

ArchMsg.Warning.out_of_range_ptrs

DISPLAY TRBPTR_EL1(%{TRBPTR_EL1}) is out of range as per TRBBASER_EL1(%{TRBBASER_EL1}) or TRBLIMITR_EL1.LIMIT(%{TRBLIMITR_EL1}). Fields:

TRBBASER_EL1 unsigned int

Base pointer.

TRBLIMITR_EL1 unsigned int

Limit pointer.

TRBPTR_EL1 unsigned int

Current write pointer.

ArchMsg.Warning.recursive_branch

An instruction is performing a branch that targets the same instruction. If this was intended then a WFI might be more effective DISPLAY recursive branch detected at PC=%{PC}.

PRIMARY KEY PC. Fields:

PC unsigned int

Address of instruction causing the branch.

ArchMsg.Warning.recursive_exception

An instruction has generated an exception that targets the same instruction. DISPLAY recursive %{TYPE} exception detected at PC=%{PC} (CPSR=%{CPSR}). PRIMARY KEY PC. Fields:

CPSR unsigned int

Processor state.

PC unsigned int

Address of instruction causing the exception.

TYPE enum

Type of exception.

ArchMsg.Warning.reentrant_vector_catch

TAGS **UNPREDICTABLE** DISPLAY Vector catch debug event on Prefetch or Data abort vector %{ADDR}. Fields:

ADDR unsigned int

vector address.

ArchMsg.Warning.reserved_it_state

TAGS **UNPREDICTABLE** DISPLAY Execution with reserved IT state %{ITSTATE}. Fields:

ITSTATE unsigned int

reserved IT state.

ArchMsg.Warning.tlb_contents_unknown

Translation has been enabled without invalidating the TLB. DISPLAY Translation was enabled in the %{PAS:(Secure|Non-secure|Root|Realm)} EL%{EL:d} translation regime (vmid=%{VMID:x}) without invalidating %{INVALIDITY:(|d-side |i-side |)}entries since power-on: %{INVALIDITY:(|D|I|)}TLB could contain **UNKNOWN** content. Fields:

EL unsigned int

Exception level of translation regime.

INVALIDITY unsigned int

Which TLB may be invalid (if TLBs are separate).

NON_SECURE bool

Whether the translation regime is non-secure.

PAS unsigned int

Physical address space of translation regime.

VMID unsigned int

Current VMID, if applicable.

ArchMsg.Warning.unknown_DLR_DSPSR

Exiting debug state with unknown DLR or DSPSR.

ArchMsg.Warning.unknown_ELR_SPSR

Returning from debug exception with unknown ELR or SPSR.

ArchMsg.Warning.unpred_FAR_on_LRCPC3_non_atomic_mem_access

TAGS **UNPREDICTABLE** Unpredictable FAR/FSTATUS on LRCPC3 pair instructions resulting in two separate memory accessesDISPLAY Unpredictable FAR on LRCPC3 non atomic memory access at %{ADDR1} with FSTATUS %{FSTATUS1} or %{ADDR2} with FSTATUS %{FSTATUS2}. Fields:

ADDR1 unsigned int

Possible FAR address 1.

ADDR2 unsigned int

Possible FAR address 2.

FSTATUS1 unsigned int

Possible fault status value 1.

FSTATUS2 unsigned int

Possible fault status value 2.

ArchMsg.Warning.unpred_async_tag_check_fail_on_sve_speculative_load_do_update_ffr

Unpredictable set to FALSE of the FFR element associated to SVE speculative load with an async tag check fail TAGS **UNPREDICTABLE**.

ArchMsg.Warning.unpredictable_a32_breakpoint

TAGS **UNPREDICTABLE** DISPLAY Unpredictable breakpoint on a A32 instruction at %{ADDR1} Offset %{OFFSET} BAS %{BAS} for a %{IS_ADDRESS_MISMATCH_BREAKPOINT:(Mismatch|Match)} Breakpoint number %{N:d} Breakpoint address %{ADDR2}. Fields:

ADDR1 unsigned int

Address.

ADDR2 unsigned int

Breakpoint address.

BAS unsigned int

BAS field.

IS_ADDRESS_MISMATCH_BREAKPOINT bool

Is a Mismatch breakpoint, not a Match breakpoint.

N unsigned int

Breakpoint number.

OFFSET unsigned int

Always has a value of zero.

ArchMsg.Warning.unpredictable_banked_register_access

TAGS **UNPREDICTABLE** DISPLAY Access to banked register (R=%{R:d} SYSm=%{SYSM:2x}) is unpredictable. Fields:

R unsigned int

R bit.

SYSM unsigned int

SYSm.

ArchMsg.Warning.unpredictable_ccfail_undef

TAGS **UNPREDICTABLE** DISPLAY An instruction %{OPCODE} at VA=%{VADDR} failed its condition codes check but would otherwise have caused a non-data-dependent trap or undefined exception. Behaviour is implementation-defined choice of exception or nop. Fields:

OPCODE unsigned int

opcode of the instruction.

VADDR unsigned int

Virtual address of the instruction being executed.

ArchMsg.Warning.unpredictable_ici_bits

TAGS **UNPREDICTABLE** DISPLAY ICI bits do not apply or are architecturally unpredictable for instruction.

ArchMsg.Warning.unpredictable_t32_breakpoint

TAGS **UNPREDICTABLE** DISPLAY Unpredictable breakpoint on a T32 instruction at %{ADDR1} Offset %{OFFSET} BAS %{BAS} for a %{IS_ADDRESS_MISMATCH_BREAKPOINT:(Mismatch|Match)} Breakpoint number %{N:d} Breakpoint address %{ADDR2}. Fields:

ADDR1 unsigned int

Address.

ADDR2 unsigned int

Breakpoint address.

BAS unsigned int

BAS field.

IS_ADDRESS_MISMATCH_BREAKPOINT bool

Is a Mismatch breakpoint, not a Match breakpoint.

N unsigned int

Breakpoint number.

OFFSET unsigned int

Indicates which halfword of a 32-bit Thumb instruction the breakpoint is set on. 0 for the first halfword or if the instruction is 16 bits wide, or 1 for the second halfword.

ArchMsg.Warning.unpredictable_t32_vector_catch

TAGS **UNPREDICTABLE** DISPLAY Unpredictable Vector catch exception on a T32 instruction at %{ADDR} Offset %{OFFSET}. Fields:

ADDR unsigned int

Address.

OFFSET unsigned int

Indicates which halfword of a 32-bit Thumb instruction the vector catch exception is set on. 0 for the first halfword or if the instruction is 16 bits wide, or 1 for the second halfword.

ArchMsg.Warning.unpredictable_watchpoint_far

TAGS **UNPREDICTABLE** DISPLAY The value for FAR/EDWAR for hitting this watchpoint may be anywhere in the range %{LOWER_BOUND} to %{UPPER_BOUND}. Fields:

LOWER_BOUND unsigned int

the lowest address accessed by the instruction that triggered the watchpoint.

UPPER_BOUND unsigned int

the highest watchpointed address accessed by that instruction.

ArchMsg.Warning.unpredictable_watchpoint_far_isscmp

DISPLAY The value for FAR/EDWAR for hitting this watchpoint may be anywhere in the range %{LOWER_BOUND} to %{UPPER_BOUND}. Fields:

LOWER_BOUND unsigned int

the lowest address accessed by the instruction that triggered the watchpoint.

UPPER_BOUND unsigned int

the highest watchpointed address accessed by that instruction.

ArchMsg.Warning.warning_AdvSIMDExpandImmUnexpectedZero

TAGS **UNPREDICTABLE** DISPLAY AdvSIMDExpandImm may treat this immediate value as **UNPREDICTABLE**.

ArchMsg.Warning.warning_ConditionalSMC

TAGS **UNPREDICTABLE** DISPLAY SMC instruction has **UNPREDICTABLE** effects when conditional and when combined with traps.

ArchMsg.Warning.warning_TLBID_out_of_range

TLBID out of range in TLBI according to the configuration of NIS, NVIS, NOS and NVOS. Fields:

NIS unsigned int

NIS.

NOS unsigned int

NOS.

NVIS unsigned int

NVIS.

NVOS unsigned int

NVOS.

TLBID unsigned int

The TLBID value used in the TLBI instruction.

ArchMsg.Warning.warning_TTL_or_TG_mismatches_current_translation_granule

TTL or TG mismatches current translation granule in TLBI VA/RVA/IPA/RIPA. Fields:

CURRENT_TG unsigned int

Current TG as indicated by TCR_ELx or VTCR_EL2.

TLBI_TG unsigned int

TG (or TTL[3:2]) used in TLBI instruction.

ArchMsg.Warning.warning_access_crosses_page_boundary

TAGS **UNPREDICTABLE** DISPLAY Access crosses page boundary.

ArchMsg.Warning.warning_access_crosses_page_boundary_has_fault

TAGS **UNPREDICTABLE** DISPLAY Access that crosses page boundary has fault(s).

ArchMsg.Warning.warning_access_crosses_page_boundary_has_fault_on_both_pages

TAGS **UNPREDICTABLE** DISPLAY Access that crosses page boundary has a fault on each page.

Fields:

ADDR1 unsigned int

FAR address on first page.

ADDR2 unsigned int

FAR address on second page.

FSTATUS1 unsigned int

Fault status on first page.

FSTATUS2 unsigned int

Fault status on second page.

ArchMsg.Warning.warning_access_crosses_page_boundary_spanning_different_memory

TAGS **UNPREDICTABLE** DISPLAY Access at address %{ADDR} crosses page from MEM_TYPE %{MEMTYPE_PAGE1} to %{MEMTYPE_PAGE2}. Fields:

ADDR unsigned int

address of access crossing page.

MEMTYPE_PAGE1 signed int

First Page's memory type (0-Normal, 1-Non-Normal).

MEMTYPE_PAGE2 signed int

Second Page's memory type (0-Normal, 1-Non-Normal).

ArchMsg.Warning.warning_access_wraps_around_memory

TAGS **UNPREDICTABLE** DISPLAY Access wraps around memory in %{WIDTH:d}bit mode.

Fields:

WIDTH unsigned int

Address width.

ArchMsg.Warning.warning_bcr_linking_status

TAGS **UNPREDICTABLE** DISPLAY DBGBCR.{BT2,BT} combination gives unpredictable behaviour.

ArchMsg.Warning.warning_bcr_mask_reserved

TAGS **UNPREDICTABLE** DISPLAY DBGBCR_ELn.MASK is set to a reserved value.

ArchMsg.Warning.warning_bvr_masked_bit_not_zero

TAGS **UNPREDICTABLE** DISPLAY DBGBVRn_EL1.VA contains a bit masked by DBGBCRn_EL1.MASK that is not zero.

ArchMsg.Warning.warning_ccsidr_unimplemented_level

TAGS **UNPREDICTABLE** DISPLAY CCSIDR read while CSSELR %{CSSELR:x} points to an unimplemented cache level. Fields:

CSSELR unsigned int

current effective value of CSSELR.

ArchMsg.Warning.warning_change_to_ns_when_tge_set

TAGS **UNPREDICTABLE** DISPLAY Attempting to change NS to 1 when HCR.TGE = 1.

ArchMsg.Warning.warning_contiguous_bit_check_abort

TAGS **UNPREDICTABLE** DISPLAY An abort occurred whilst attempting to check TLB entry is contiguous at %{ENTRY_ADDR}. Fields:

ENTRY_ADDR unsigned int

address of conflicting TLB entry.

ArchMsg.Warning.warning_contiguous_bit_error

TAGS **UNPREDICTABLE** DISPLAY TLB entry at %{ENTRY_ADDR} was not contiguous with entry at %{CONTIG_ADDR} Contents are %{ENTRY_DATA} but expected %{CONTIG_DATA}. Fields:

CONTIG_ADDR unsigned int

address of the TLB first read, with which this entry is expected to be contiguous.

CONTIG_DATA unsigned int

expected contents based on the entry at CONTIG_ADDR.

ENTRY_ADDR unsigned int

address of conflicting TLB entry.

ENTRY_DATA unsigned int

contents of conflicting TLB entry.

ArchMsg.Warning.warning_contiguous_bit_error_D128

TAGS **UNPREDICTABLE** DISPLAY TLB entry at %{ENTRY_ADDR} was not contiguous with entry at %{CONTIG_ADDR} Contents are %{ENTRY_DATA_HI} %{ENTRY_DATA_LO} but expected %{CONTIG_DATA}. Fields:

CONTIG_ADDR unsigned int

address of the TLB first read, with which this entry is expected to be contiguous.

CONTIG_DATA unsigned int

expected contents based on the entry at CONTIG_ADDR.

ENTRY_ADDR unsigned int

address of conflicting TLB entry.

ENTRY_DATA_HI unsigned int

Higher bit contents of conflicting TLB entry.

ENTRY_DATA_LO unsigned int

Lower bit contents of conflicting TLB entry.

ArchMsg.Warning.warning_cp10_cp11_mismatch

TAGS **UNPREDICTABLE** DISPLAY Unpredictable configuration of CP10 and CP11 controls at EL %{EL:d}. Fields:

EL unsigned int

EL owning the control.

ArchMsg.Warning.warning_cp10_cp11_reserved_value

TAGS **UNPREDICTABLE** DISPLAY Unpredictable configuration of CPACR CP10 ({CP10}) or CP11 ({CP11}). Fields:

CP10 unsigned int

CP10 access.

CP11 unsigned int

CP11 access.

ArchMsg.Warning.warning_csselr_level_out_of_range

TAGS **UNPREDICTABLE** DISPLAY CSSELR.Level out of range, written value {LEVEL} with only {IMPLEMENTED:d} levels implemented. Fields:

IMPLEMENTED unsigned int

number of cache levels implemented.

LEVEL unsigned int

written value of CSSELR.Level.

TYPE enum

The type of cache selected by CSSELR.InD and TnD.

ArchMsg.Warning.warning_debug_flow_control_bits_not_obeyed

TAGS **UNPREDICTABLE** DISPLAY Flow control bits for {REGISTER} were not obeyed, causing {ERROR}. Fields:

ERROR enum

type of error resulting.

REGISTER enum

register under consideration.

ArchMsg.Warning.warning_debug_register_access_during_reset

TAGS **UNPREDICTABLE** DISPLAY Unpredictable {IS_WRITE:(read|write)} access to debug register offset {OFFSET} during reset. Fields:

IS_WRITE unsigned int

Write Not Read.

OFFSET unsigned int

Register Offset.

ArchMsg.Warning.warning_decode_cps_inconsistent_fields

TAGS **UNPREDICTABLE** DISPLAY The mode setting fields in the CPS instruction are inconsistent.

ArchMsg.Warning.warning_decode_invalid_state

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable in current exception-level/security state.

ArchMsg.Warning.warning_default_cacheable_mmu_on

TAGS **UNPREDICTABLE** DISPLAY Default cacheable is enabled (HCR.DC = 1) while MMU is enabled.

ArchMsg.Warning.warning_default_cacheable_vmmu_off

TAGS **UNPREDICTABLE** DISPLAY Default cacheable is enabled (HCR.DC = 1) while HCR.VM = 0.

ArchMsg.Warning.warning_deprecated_wvr_bit_2_set

TAGS **UNPREDICTABLE** DISPLAY DBGWVR%{N:d}_EL1=%{WVR:h} has bit 2 set, deprecated in ARMv8. Fields:

N unsigned int

index.

WVR unsigned int

DBGWVR.

ArchMsg.Warning.warning_eret_while_software_step_active_pending

Missing ISB between setting MDSCR_EL1.SS and ERET TAGS **UNPREDICTABLE**.

ArchMsg.Warning.warning_exclusive_to_non_normal

TAGS **UNPREDICTABLE** DISPLAY Exclusive Access to Non-Normal Memory %{ADDR}. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_exclusive_to_non_writeback

TAGS **UNPREDICTABLE** DISPLAY Exclusive Access to Non-Writeback-Cacheable Memory %{ADDR}. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_execute_from_device_memory

TAGS **UNPREDICTABLE** DISPLAY An attempt was made to execute from Device memory.

ArchMsg.Warning.warning_illegal_cpsr_mode

TAGS **UNPREDICTABLE** DISPLAY Writing an illegal or unimplement mode (%{NEW_MODE:h}) to CPSR was unpredictable. Fields:

NEW_MODE unsigned int

New value of CPSR.M.

ArchMsg.Warning.warning_illegal_srs_mode

TAGS **UNPREDICTABLE** DISPLAY Illegal or **UNPREDICTABLE** mode (%{MODE:h}) used for SRS instruction. Fields:

MODE unsigned int

mode for Banked SP.

ArchMsg.Warning.warning_implementation_defined_read_debug_register_in_SCS

Read Access to Debug registers from the processor is IMPLEMENTATION DEFINED TAGS IMP_DEF. Fields:

PPB_OFFSET unsigned int
debug register offset.

ArchMsg.Warning.warning_implementation_defined_sequential_security_transitions_supported

The behavior of sequential instruction fetches that cross from non-secure to secure memory and contain an SG instruction is CONSTRAINED UNPREDICTABLE TAGS UNPREDICTABLE.

ArchMsg.Warning.warning_implementation_defined_stack_limit_check_supported

It is IMPLEMENTATION DEFINED whether stack pointer limit checking is performed for this instruction TAGS IMP_DEF. Fields:

ADDRESS unsigned int
Address of instruction.

OPCODE unsigned int
instruction opcode.

ArchMsg.Warning.warning_implementation_defined_taggednonsharedmemorytype

TAGS IMP_DEF DISPLAY Memory attribute has Tagged, non shared memory type; it is IMPLEMENTATION DEFINED whether the memory location is treated as Tagged or non-Tagged.

ArchMsg.Warning.warning_implementation_defined_write_debug_register_in_SCS

Write Access to Debug registers from the processor is IMPLEMENTATION DEFINED TAGS IMP_DEF. Fields:

DATA unsigned int
data attempted to be written.

PPB_OFFSET unsigned int
debug register offset.

ArchMsg.Warning.warning_invalid_tcr_granule

TAGS UNPREDICTABLE DISPLAY TCR.TG%{TG_ID:(0|1)} bits have been set to a granule size %{{REQUEST:d}}K not implemented - using %{{SUBSTITUTE:d}}K. Fields:

REQUEST signed int
page size requested (or 0 for reserved).

SUBSTITUTE signed int
best guess available page size.

TG_ID bool
bits TG0 or TG1.

ArchMsg.Warning.warning_load_multiple_user_registers_from_user_mode

TAGS UNPREDICTABLE DISPLAY An LDM(user registers) instruction executed from user mode.

ArchMsg.Warning.warning_mte_in_mem_attr_not_supported

TAGS UNPREDICTABLE DISPLAY Memory attribute %{ATTR} has Tagged memory type but Armv8.5-MTE is not implemented. Fields:

ATTR unsigned int

Memory attribute.

ArchMsg.Warning.warning_reserved_breakpoint_state_match

TAGS UNPREDICTABLE DISPLAY %{IS_BREAKPOINT:(Watchpoint|Breakpoint)} programmed with a reserved combination of HMC, SSC and %{IS_BREAKPOINT:(PAC|PMC)}. Fields:

IS_BREAKPOINT unsigned int

Is a breakpoint, not a watchpoint.

ArchMsg.Warning.warning_shareability

TAGS UNPREDICTABLE DISPLAY Unpredictable: combination of the 1st and 2nd stages of translation is Normal Inner Non-Cacheable, Outer Non-Cacheable.

ArchMsg.Warning.warning_software_step_set_while_enabled

TAGS UNPREDICTABLE DISPLAY MDSCR_EL1.SS set to 1 while software step debug exceptions are enabled.

ArchMsg.Warning.warning_thumb_instruction_wraps_around_memory

TAGS UNPREDICTABLE DISPLAY Thumb instruction wraps around memory.

ArchMsg.Warning.warning_ttbr_sbz_bits_are_not_zero

TAGS UNPREDICTABLE DISPLAY %{TTBR}[%{MSB:d}:0] should be zero, but are not. Fields:

MSB unsigned int

x-1.

TTBR enum

which TTBR register.

ArchMsg.Warning.warning_unaligned_address_dbgdtrrx_write

TAGS UNPREDICTABLE DISPLAY Unaligned address %{ADDR} in X0/R0 when DBGDTRRX written in memory access mode. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unaligned_address_dbgdtrtx_read

TAGS UNPREDICTABLE DISPLAY Unaligned address %{ADDR} in X0/R0 when DBGDTRTX read in memory access mode. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unaligned_to_device

TAGS UNPREDICTABLE DISPLAY Unaligned Access to Device Memory %{ADDR}. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unaligned_to_strongly_ordered

TAGS **UNPREDICTABLE** DISPLAY Unaligned Access to Strongly Ordered Memory %{ADDR}.

Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unknown_sau_rnr

TAGS **UNPREDICTABLE** DISPLAY SAU_RNR was set to an unsupported value.

ArchMsg.Warning.warning_unpredictable_AIRCR_PRIS_and_BFHFNMINs

TAGS **UNPREDICTABLE** DISPLAY The effect of setting both AIRCR.BFHFNMINs and AIRCR.PRIS to 1 is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_AIRCR_VECTCLRACTIVE_when_not_in_debug

TAGS **UNPREDICTABLE** DISPLAY The effect of writing a 1 to AIRCR.VECTCLRACTIVE if the processor is not halted in Debug state is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_AIRCR_VECTRESET_and_SYSRESETREQ

TAGS **UNPREDICTABLE** DISPLAY When the processor is halted in Debug state, if a write to the register writes a 1 to both VECTRESET and SYSRESETREQ, the behavior is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_AIRCR_VECTRESET_when_not_in_debug

TAGS **UNPREDICTABLE** DISPLAY The effect of writing a 1 to AIRCR.VECTRESET if the processor is not halted in Debug state is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_AIRCR_incorrect_VKEY

TAGS **UNPREDICTABLE** DISPLAY The value 0x05FA must be written to AIRCR.VKEY, otherwise the register write is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_EXC_RETURN_Reserved_Bit

EXC_RETURN[23:7] are reserved with the special condition that all bits should be written as one. Values other than all 1s are **UNPREDICTABLE**. TAGS **UNPREDICTABLE**. Fields:

EXC_RETURN unsigned int

Special PC value.

SBO_MASK unsigned int

Which bits Should Be One to not UNPRED.

ArchMsg.Warning.warning_unpredictable_PMEVTYPEPERn_ELO_VS_reserved_value

TAGS **UNPREDICTABLE** DISPLAY Unpredictable reserved value of PMEVTYPER<n>_ELO.VS.

ArchMsg.Warning.warning_unpredictable_change_to_DEMCR_MON_STEP_at_insufficient_priority

TAGS **UNPREDICTABLE** DISPLAY The effect of changing DEMCR.MON_STEP at an execution priority that is lower than the priority of the DebugMonitor exception is **CONSTRAINED UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_change_to_priority_of_active_exception

TAGS UNPREDICTABLE DISPLAY Changing the priority of an active exception is UNPREDICTABLE.

ArchMsg.Warning.warning_unpredictable_clear_lspact

TAGS UNPREDICTABLE DISPLAY UNPREDICTABLE state on context switch. Fields:

DRVEXC_TAKEN unsigned int

Derived exception taken, not pended.

REACHED_LAST_STORE unsigned int

Only the last faulted.

ArchMsg.Warning.warning_unpredictable_dbg_exit_unaligned_dlr

TAGS UNPREDICTABLE DISPLAY Debug exit to AArch32 with DLR[0] set to 1. PC[0] can be set to 0 or to DLR[0].

ArchMsg.Warning.warning_unpredictable_exception_catch

TAGS UNPREDICTABLE DISPLAY Generation of an exception catch event is unpredictable.

ArchMsg.Warning.warning_unpredictable_exception_return_inconsistent_state

TAGS UNPREDICTABLE DISPLAY state on exception return is unpredictable.

ArchMsg.Warning.warning_unpredictable_exception_return_instruction

TAGS UNPREDICTABLE DISPLAY exception return instruction is unpredictable.

ArchMsg.Warning.warning_unpredictable_in_debug_state

TAGS UNPREDICTABLE DISPLAY Instruction is UNPREDICTABLE when executed in debug state.

ArchMsg.Warning.warning_unpredictable_mair_encoding

TAGS UNPREDICTABLE DISPLAY Memory attribute %{ATTR} has invalid MAIR encoding; this may lead to unpredictable results. Fields:

ATTR unsigned int

Memory attribute.

ArchMsg.Warning.warning_unpredictable_pmu_counter_access

TAGS UNPREDICTABLE DISPLAY Access to PMU counters from non-secure EL0 or EL1 is UNPREDICTABLE with MDCR_EL2.HPMN set to 0.

ArchMsg.Warning.warning_unpredictable_prioritization_breakpoint_match_vector_catch

TAGS UNPREDICTABLE DISPLAY Two events with the same priority occurred at the same instruction: Address Matching Vector Catch debug event, and Breakpoint debug event. It is constrained unpredictable which is taken.

ArchMsg.Warning.warning_unpredictable_stack_selection

TAGS UNPREDICTABLE DISPLAY UNPREDICTABLE state on context switch.

ArchMsg.Warning.warning_unpredictable_tsize_out_of_range

TAGS UNPREDICTABLE DISPLAY %{TCR}.T%{TTBR:d}SZ (%{TSIZE:d} bits) is out of range. Must be between 25 and %{TSIZE_MAX:d}. Fields:

TCR enum

Which TCR.

TSIZE signed int

Value of TSize.

TSIZE_MAX signed int

The maximum allowed value.

TTBR unsigned int

Which TTBR region.

ArchMsg.Warning.warning_unpredictable_unaligned_pc_as_base_register

TAGS **UNPREDICTABLE** DISPLAY The PC value must be word-aligned, otherwise the behavior of the instruction is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_unaligned_pop_stack

TAGS **UNPREDICTABLE** DISPLAY **CONSTRAINED UNPREDICTABLE** unaligned stack frame `%{FRAME_ADDR:x}` in exception-return. Fields:

FRAME_ADDR unsigned int

unaligned stack frame address.

ArchMsg.Warning.warning_unpredictable_unaligned_to_non_writeback

TAGS **UNPREDICTABLE** DISPLAY Unaligned Access to Non-Writeback-Cacheable Memory.

ArchMsg.Warning.warning_unpredictable_vmsa_memattrib

TAGS **UNPREDICTABLE** DISPLAY Unpredictable vmsa memory attribute with texcb value programmed to `%{TEXCB}` when tex remap is `%{TEX_REMAP}`. Fields:

TEXCB unsigned int

texcb value.

TEX_REMAP bool

use tex remap.

ArchMsg.Warning.warning_unpredictable_vtcr_t0sz_sl0

TAGS **UNPREDICTABLE** DISPLAY The combination of VTCR.TOSZ=`%{TOSZ:d}` and VTCR.SL0=`%{SL0:d}` is unpredictable. Fields:

SL0 signed int

VTCR.SL0, starting level for stage 2 translation table walks.

TOSZ signed int

VTCR.TOSZ, required input address range.

ArchMsg.Warning.warning_unpredictable_write_DHCSR

TAGS **UNPREDICTABLE** DISPLAY **UNPREDICTABLE** set of bit-changes in DHCSR. Fields:

NEW unsigned int

data attempted to be written.

OLD unsigned int

previous value.

ArchMsg.Warning.warning_unpredictable_write_SHCSR

TAGS UNPREDICTABLE DISPLAY UNPREDICTABLE setting a pending bit to 1 for an exception with priority \geq execution priority.

ArchMsg.Warning.warning_unsupported_access_to_memory_mapped_register

TAGS UNPREDICTABLE DISPLAY Access to memory mapped register is unsupported.

ArchMsg.Warning.warning_user_jmcr_access

TAGS UNPREDICTABLE DISPLAY User %{WRITE:(write|read)} access to JMCR is unpredictable. Fields:

WRITE unsigned int
access is write.

ArchMsg.Warning.warning_wcr_mask_and_bas

TAGS UNPREDICTABLE DISPLAY DBGWCR_EL1.MASK is non-zero and BAS is not 0xff.

ArchMsg.Warning.warning_wcr_mask_reserved

TAGS UNPREDICTABLE DISPLAY DBGWCR_ELn.MASK is set to a reserved value.

ArchMsg.Warning.warning_wcr_non_configuous_bas

TAGS UNPREDICTABLE DISPLAY DBGWCR_EL1.BAS has non-contiguous set of ones.

ArchMsg.Warning.warning_wvr_masked_bit_not_zero

TAGS UNPREDICTABLE DISPLAY DBGWVRn_EL1.VA contains a bit masked by DBGWCRn_EL1.MASK that is not zero.

ArchMsg.Why.why_branch_target_exception

A branch target exception has occurred. DISPLAY branch target exception: target pc=%{PC}, target opcode=%{OPCODE}. PRIMARY KEY PC. Fields:

BTYPE unsigned int
pstate.BTYPE at time branch taken.

OPCODE unsigned int
opcode at branch target.

PC unsigned int
Virtual Address of branch target.

ArchMsg.Why.why_illegal_state

An illegal mode change has occurred. DISPLAY illegal mode change: %{MESSAGE}. Fields:

MESSAGE string
The message.

BRANCH_MISPREDICT

Simulating branch mispredict. Fields:

PC unsigned int
Origin address (or 0 if unavailable).

BRA_DIR

Direct branches, to immediate address. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISSET enum

The instruction set of the branch instruction.

IS_COND bool

Indicates if this is a conditional branch.

IS_HINTED bool

Whether the conditional instruction is hinted.

IS_ISB bool

Whether the instruction is ISB.

IS_LINK bool

Indicates if this is a branch with link.

PC unsigned int

The address of the branch instruction.

TARGET_ISET enum

The instruction set after the branch.

TARGET_PC unsigned int

The address the instruction branches to.

BRA_INDIR

Indirect branches, perhaps to a register. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISSET enum

The instruction set of the branch instruction.

IS_COND bool

Indicates if this is a conditional branch.

IS_HINTED bool

Whether the conditional instruction is hinted.

IS_ISB bool

Whether the instruction is ISB.

IS_LINK bool

Indicates if this is a branch with link.

PC unsigned int

The address of the branch instruction.

TARGET_ISET enum

The instruction set after the branch.

TARGET_PC unsigned int

The address the instruction branches to.

BRA_INDIR_CCFAIL

Indirect branches fail, perhaps to a register.

BRA_RET_CCFAIL

Branch return fail, perhaps to a register.

CACHE_MAINTENANCE_OP

Cache Maintenance Operation. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

DATA unsigned int

Specified MVA or Set/way.

FUNCTION enum

Clean / Invalidate.

SCOPE enum

Affected region.

SIDE enum

Inst / Data.

CCFAIL

Conditional instruction condition check fail. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CCFAIL_UNC

Unconditional instruction condition check fail. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CCPASS

Conditional instruction condition check pass. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CCPASS_UNC

Unconditional instruction condition check pass. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CODE_CACHE_MAINT

Code cache maintenance. Fields:

END_INCL unsigned int

Inclusive end address for VA or PA requests.

NS bool

Non-secure world for PA.

START unsigned int

Start address for VA or PA requests.

TYPE enum

Request type.

COMPILE_BLOCK_END

Last instruction of basic block translated. Fields:

VADDR unsigned int

Address of next instruction after this basic block.

COMPILE_BLOCK_START

First instruction of basic block translated. Fields:

VADDR unsigned int

Address of first instruction in basic block.

COMPILE_INST

ARM instruction compiled. Fields:

DISASS string

Disassembly of the instruction.

ISET enum

The instruction set of this instruction.

ITSTATE unsigned int

The ITSTATE current for the instruction.

OPCODE unsigned int

The opcode of the instruction.

PC unsigned int

The address of the instruction.

SIZE unsigned int

The size of the instruction in bytes.

CONTEXTIDR

Context ID Register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

NS bool

Secure or nonsecure banked register is accessed.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The new value written.

CONTEXT_SYNC

Called for every context synchronization event. Fields:

ARCH_CSE bool

Architectural Context Synchronization Event.

PC unsigned int

The address of the synchronization instruction.

SYSREG_SYNC bool

System register synchronization event.

CORE_ENDIAN

Core BE8 Big-Endian state changed. Fields:

BE8 bool

Core BE8 Big-Endian state.

CORE_INFO

Static processor attributes. Only triggered by a call to DumpState(). Fields:

ARCH_PROFILE enum

The architecture profile of the core.

CLUSTER_ID unsigned int

The cluster ID of this processor.

CORE_NUM unsigned int

The number of this core in an MP processor.

DETAILED_CORE_FEATURES enum

Detailed core features as a bitfield. Enum values are bitmasks to extract the info from the field.

FPU_VERSION enum

The VFP version implemented by the core.

MEM_ARCH enum

The memory architecture of the core.

MIDR_EL1 unsigned int

MIDR_EL1 value.

NUM_CORES unsigned int

The number of cores in this MP processor.

QUANTUM_SIZE unsigned int

The default quantum size of the core (in ticks).

SECURITY_FEATURES bool

Does the core have security features?.

CORE_LOADS

Processor load accesses. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ACQREL enum

Is this an acquire/release.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DATA unsigned int

The data read or written.

ELEMENT_SIZE unsigned int

Width of each element.

ENDIANNESS enum

Byte ordering of this access.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

LOCK enum

Normal, exclusive or locked access.

MECID unsigned int

Memory Encryption Context ID, if applicable.

MECID2 unsigned int

Memory Encryption Context ID of the second page, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PADDR_TAG unsigned int

If tag checked, the physical address tag for the access.

PAGE_SIZE unsigned int

Page size as $\log_2(\text{size in bytes})$.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK, 2=Exclusive Failed, 3=Unsupported.

SIZE unsigned int

Width of the access in bytes. Only required if DATA is not traced.

TAG_CHECKED bool

Is this access tag checked.

TAG_CHECKED2 bool

Is the second page access tag checked.

TRANS bool

Is this a translated access.

VADDR unsigned int

The virtual address of the access.

CORE_REGS

Changes of the core registers R0 to R14. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number, 0 to 14.

MODE enum

Bank of the register accessed.

OLD_VALUE unsigned int

The old value overwritten.

PHYS_ID enum

The physical register accessed.

VALUE unsigned int

The new value written to the register.

CORE_REGS64

Changes of the core registers X0..X30, SP_ELn. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ID enum

The register number, 0-30 for X0-X30, >=32 for SP_ELn.

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

CORE_REGS64_READ

General purpose AArch64 register has been read X0..X30, SP_ELn. Fields:

ID enum

The register number, 0-30 for X0-X30, >=32 for SP_ELn.

VALUE unsigned int

The value read from the register.

CORE_STALLED_TRANSITION

Detect when a core is prevented from progressing PChannel states. Fields:

CURRENT_STATE enum

Current power mode of core.

REASON string

The reason a transition was stalled, and what must be done to unblock it.

TARGET_STATE enum

Desired power mode of core after transition.

CORE_STORES

Processor store accesses. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ACQREL enum

Is this an acquire/release.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DATA unsigned int

The data read or written.

ELEMENT_SIZE unsigned int

Width of each element.

ENDIANNESS enum

Byte ordering of this access.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

LOCK enum

Normal, exclusive or locked access.

MECID unsigned int

Memory Encryption Context ID, if applicable.

MECID2 unsigned int

Memory Encryption Context ID of the second page, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PADDR_TAG unsigned int

If tag checked, the physical address tag for the access.

PAGE_SIZE unsigned int

Page size as $\log_2(\text{size in bytes})$.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK, 2=Exclusive Failed, 3=Unsupported.

SIZE unsigned int

Width of the access in bytes. Only required if DATA is not traced.

TAG_CHECKED bool

Is this access tag checked.

TAG_CHECKED2 bool

Is the second page access tag checked.

TRANS bool

Is this a translated access.

VADDR unsigned int

The virtual address of the access.

CP14_READ

System Coprocessor register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP14 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc1 unsigned int

opcode 1.

opc2 unsigned int

opcode 2.

CP14_WRITE

System Coprocessor register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP14 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc1 unsigned int

opcode 1.

opc2 unsigned int

opcode 2.

CP15_READ

System Coprocessor register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP15 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc1 unsigned int

opcode 1.

opc2 unsigned int

opcode 2.

CP15_READ64

System Coprocessor register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP15 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc unsigned int

opcode 1.

CP15_WRITE

System Control Coprocessor register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP15 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc1 unsigned int

opcode 1.

opc2 unsigned int

opcode 2.

CP15_WRITE64

System Coprocessor register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP15 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc unsigned int

opcode 1.

CPSR

CPSR change. (In AArch64 PSTATE represented in SPSR format). Fields:

CORE_NUM unsigned int

Core number in a multi processor.

OLD_VALUE unsigned int

The old CPSR value.

UNKNOWN unsigned int

Bits within the register that have unknown value.

VALUE unsigned int

The new CPSR value.

CRYPTO_SPEC

Every crypto instruction speculatively executed.

CorePowerStateChange

Core power-up state has changed. DISPLAY Core powered %{PU:(up|down)}. Fields:

PU bool

Whether the core powered up.

DATA_CACHE_SET_ALLOC_TAGS

Set allocation tags from DC GVA instruction. Fields:

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

NSDESC unsigned int

The security state of the access.

PADDR unsigned int

The physical address of the access.

PAS enum

The physical address space of the page.

RESPONSE enum

0=Aborted, 1=OK.

SIZE unsigned int

The size of memory in bytes.

TAG unsigned int

The value of the tag being set.

TAGGING_TYPE enum

Type of Memory tagging (Physical or Virtual).

TAG_VADDR unsigned int

The virtual address of the Virtual Tag if Virtual tagging is used; otherwise 0.

VADDR unsigned int

Virtual address used for setting allocation tags.

DATA_CACHE_ZERO

Zero and invalidate cache line from DC ZVA instruction. Fields:

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

NSDESC unsigned int

The security state of the access.

PADDR unsigned int

The physical address of the access.

PADDR_TAG unsigned int

If tag checked, the physical address tag for the access.

PAS enum

The physical address space of the page.

RESPONSE enum

0=Aborted, 1=OK.

SIZE unsigned int

The size of zero'd memory in bytes.

TAG_CHECKED bool

Is this access tag checked.

VADDR unsigned int

Virtual address of the zero'd cache line.

DEBUG_EVENT

Hardware debug support event. Fields:

EVENT enum

Description of event.

VALUE unsigned int

data value.

DEBUG_TARGET_ISET

Hardware debug target ISA. Fields:

TARGET_ISET enum

The target instruction set of the debug state.

DMI_ALLOCATE

DMI allocated. Fields:

ACCESS_DIRECTION enum

READ / WRITE.

PADDR unsigned int

The physical address of the DMI.

PTR unsigned int

DMI range pointer.

SIDE enum

DSIDE / ISIDE.

SIZE unsigned int

The size of DMI.

VADDR unsigned int

The virtual address of the DMI..

DMI_HIT

DMI hit, access used DMI (only fire for small DMI regions i.e. less than a full page). Fields:

ACCESS_DIRECTION enum

READ / WRITE.

PADDR unsigned int

The physical address of the DMI.

SIDE enum

DSIDE / ISIDE.

SIZE unsigned int

The size of the DMI.

VADDR unsigned int

The virtual address of the DMI..

DMI_REVOKE

DMI revoked. Fields:

ACCESS_DIRECTION enum

READ / WRITE.

PADDR unsigned int

The physical address of the DMI.

SIDE enum

DSIDE / ISIDE.

SIZE unsigned int

The size of DMI.

VADDR unsigned int

The virtual address of the DMI..

END_COMPILE

Compilation end. Fields:

END_OF_PAGE_COUNT unsigned int

Number of basic blocks exited due to page boundary since START_COMPILE.

FETCHFAIL_COUNT unsigned int

Number of basic blocks exited due to ifetch failure START_COMPILE.

INST_COUNT unsigned int

Number of instructions compiled since START_COMPILE.

NONSEQ_COUNT unsigned int

Number of basic blocks exited due to non-sequential instructions since START_COMPILE.

PAGE_STRADDLE_COUNT unsigned int

Number of basic blocks exited due to unaligned instructions crossing page since START_COMPILE.

ESB

Error Synchronization Barrier hit. Fields:

ASYNC bool

ESB is async.

IMPLICIT bool

ESB is implicit.

EXCEPTION

Exception taken. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ESR unsigned int

The value assigned to ESR when exception is taken.

ESR_EC unsigned int

The Value assigned to ESR.EC when exception is taken.

IS_PHYSICAL bool

Physical or Virtual exception.

IS_TRAP bool

Exception coming from a trap.

LR unsigned int

The value assigned to the link register.

PC unsigned int

The location where the exception occurred.

PREFERRED_RETURN unsigned int

The preferred return address for the exception.

TAKEN_LOCALLY bool

Whether exception is taken locally(same exception level).

TARGET_ISET enum

The instruction set of the exception handler code.

TARGET_PC unsigned int

The address the exception branches to.

VECTOR enum

The exception vector.

VECTOR_OFFSET unsigned int

The offset of an exception vector from the base address.

EXCEPTION_END

Every exception completed.

EXCEPTION_INFO

Trace human-readable reason of the current exception for data/instruction aborts, floating point exceptions and unknown exceptions. Fields:

DESCRIPTION string

Reason for current exception.

ESR_EC unsigned int

The value of ESR.EC when exception is taken.

EXCEPTION_RAISE

Every exception raised.

EXCEPTION_RETURN

Branches on leaving exception. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISET enum

The instruction set of the branch instruction.

IS_COND bool

Indicates if this is a conditional branch.

IS_HINTED bool

Whether the conditional instruction is hinted.

IS_ISB bool

Whether the instruction is ISB.

IS_LINK bool

Indicates if this is a branch with link.

PC unsigned int

The address of the branch instruction.

TARGET_ISET enum

The instruction set after the branch.

TARGET_PC unsigned int

The address the instruction branches to.

EXCEPTION_RETURN_PREBRANCH

Exception return event that must occur before PC and processor state have updated.

EXCEPTION_START

Every exception started.

FIQ_TAKEN

FIQ taken exception.

FP_EXCEPTIONS_SIGNALLED

Trace which FP exceptions were signalled as a part of the FP operations performed by an instruction. Fields:

FP_EXCEPTIONS_SIGNALLED_MASK unsigned int

Indicates which FP exceptions were signalled by an instruction. This uses the same encoding as the cumulative floating-point exception bitfields in FPSR.

FP_STATE

Floating point state. Fields:

FPSR unsigned int

FPSR register value.

MASK unsigned int

The mask indicating updated cumulative bits.

FREQ_CHANGED

Core frequency changed. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The instruction count of this core.

NEW_FREQ signed int

The new frequency of this core (expressed in Hz).

OLD_FREQ signed int

The old frequency of this core (expressed in Hz).

HLT

HLT instruction occurred. Fields:

IMM16 unsigned int

The 16-bit immediate value encoded in HLT instruction.

INST

Every instruction executed. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DEBUG_STATE bool

The instruction is executed in debug state.

DISASS string

Disassembly of instruction.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISSET enum

The current instruction set.

ITSTATE unsigned int

The current ITSTATE.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

MODE enum

The mode the core is in.

NS unsigned int

The core's non-secure bit.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

OPCODE unsigned int

The opcode of the instruction.

PADDR unsigned int

The physical address of the instruction.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the instruction.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the page.

PC unsigned int

The address of the instruction.

SECURITY_STATE enum

The core's security state.

SIZE unsigned int

The size of the instruction in bytes.

INST_END

Every instruction completed.

INST_START

Every instruction started. Fields:

DEBUG_STATE bool

The instruction is executed in debug state.

ISSET enum

The current instruction set.

MODE enum

The mode the core is in.

NS enum

The current Secure State.

PC unsigned int

The address of the conditional instruction.

SECURITY_STATE enum

The current Security State.

INST_STRADDLE

Instruction straddles boundary. Fields:

DEBUG_STATE bool

The instruction is executed in debug state.

ISSET enum

The current instruction set.

MODE enum

The mode the core is in.

NS enum

The current Secure State.

PC unsigned int

The address of the conditional instruction.

SECURITY_STATE enum

The current Security State.

IRQ_TAKEN

IRQ taken exception.

ISB

ISB instruction occurred. Fields:

PC unsigned int

The address of the ISB instruction.

LOAD_MULTIPLE_REGS_LIST

List of destination registers (not source) of a multiple registers load instruction. Fields:

REG string

Register to add into the list.

LOCAL_MONITOR

Local monitor activity. Fields:

PADDR unsigned int

Local Monitor Address.

State enum

State of the monitor (Open/Exclusive).

MAPPING_CREATE

A memory mapping has been created. This is an implementation detail of the model and so may change without notice. Fields:

ASID unsigned int

The ASID used during translation, or 0 if no ASID was used.

IS_VALID bool

Whether the mapping is valid or not.

PADDR unsigned int

Base physical address of region, or 0 if not known.

SIDE enum

I-side or D-side.

SIZE unsigned int

Size of the page, or 0 if not known.

VADDR unsigned int

Base virtual address of region.

VMID unsigned int

The VMID used during translation, or 0 if no VMID was used.

MAPPING_DESTROY

A memory mapping has been destroyed. This is an implementation detail of the model and so may change without notice. Fields:

ASID unsigned int

The ASID used during translation.

CAUSE enum

The cause of invalidation.

PADDR unsigned int

Base physical address of page.

SIDE enum

I-side or D-side.

SIZE unsigned int

Size of the page.

VADDR unsigned int

Base virtual address of page.

VMID unsigned int

The VMID used during translation.

MEMMAP_DEBUG_READ

Memory mapped reads to the debug registers. Fields:

ADDR unsigned int

address.

CORE_NUM unsigned int

Core number in a multi processor.

EXT bool

Whether access is from an external device (such as the DAP).

NS enum

Secure state of the access.

REG_NAME string

Name of the debug register accessed.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

MEMMAP_DEBUG_WRITE

Memory mapped writes to the debug registers. Fields:

ADDR **unsigned int**

address.

CORE_NUM **unsigned int**

Core number in a multi processor.

EXT **bool**

Whether access is from an external device (such as the DAP).

NS **enum**

Secure state of the access.

REG_NAME **string**

Name of the debug register accessed.

UNDEF **bool**

The register accessed is undefined.

VALUE **unsigned int**

The new value written.

MEMTAG_LOADS

Allocation tag memory read; when attributes and system configuration allow tag accesses.

Fields:

ACCESS_TYPE **enum**

The type of instruction performing the access.

ATTR **unsigned int**

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 **unsigned int**

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

NSDESC **unsigned int**

The physical address non-secure bit.

NSDESC2 **unsigned int**

The second page physical address non-secure bit.

NUM_GRANULES **unsigned int**

Number of tags to be Loaded/stored.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK.

TAGGING_TYPE enum

Type of Memory tagging (Physical or Virtual).

TAGS unsigned int

The tags read or written. Tags for subsequent granules would be found at subsequent bytes in this field (One byte for each tag).

TAG_VADDR unsigned int

The virtual address of the Virtual Tag if Virtual tagging is used; otherwise 0.

VADDR unsigned int

The virtual address of the access.

MENTAG_LOAD_INST

Allocation tag load instruction; regardless of tags being actually accessed. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

NUM_GRANULES unsigned int

Number of tags to be Loaded/stored.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK.

TAGGING_TYPE enum

Type of Memory tagging (Physical or Virtual).

TAGS unsigned int

The tags read or written. Tags for subsequent granules would be found at subsequent bytes in this field (One byte for each tag).

TAG_VADDR unsigned int

The virtual address of the Virtual Tag if Virtual tagging is used; otherwise 0.

VADDR unsigned int

The virtual address of the access.

MEMTAG_STORES

Allocation tag memory write; when attributes and system configuration allow tag accesses.

Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

NUM_GRANULES unsigned int

Number of tags to be Loaded/stored.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK.

TAGGING_TYPE enum

Type of Memory tagging (Physical or Virtual).

TAGS unsigned int

The tags read or written. Tags for subsequent granules would be found at subsequent bytes in this field (One byte for each tag).

TAG_VADDR unsigned int

The virtual address of the Virtual Tag if Virtual tagging is used; otherwise 0.

VADDR unsigned int

The virtual address of the access.

MEMTAG_STORE_INST

Allocation tag store instruction; regardless of tags being actually accessed. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

NUM_GRANULES unsigned int

Number of tags to be Loaded/stored.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK.

TAGGING_TYPE enum

Type of Memory tagging (Physical or Virtual).

TAGS unsigned int

The tags read or written. Tags for subsequent granules would be found at subsequent bytes in this field (One byte for each tag).

TAG_VADDR unsigned int

The virtual address of the Virtual Tag if Virtual tagging is used; otherwise 0.

VADDR unsigned int

The virtual address of the access.

MMU_D128_TTB_READ

This event is triggered by reads caused by a D128 translation table walk. Fields:

ABORTED bool

Was the walk successful.

DATA_HIGH unsigned int

The data upper bits read.

DATA_LOW unsigned int

The data lower bits read.

IPA unsigned int

For stage 1, the IPA of the read.

LEVEL unsigned int

Translation table level.

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

OUTADDR unsigned int

The output physical address.

PADDR unsigned int

The physical address of the read.

PASmem enum

Physical Address Space of the accessed memory.

PASreq enum

Physical Address Space of the requested lookup.

REGIME_EL enum

Entry matches in this translation regime.

REQUESTER enum

The requester of this table walk.

SIDE enum

Inst / Data.

STAGE unsigned int

Translation stage.

MMU_D128_TTB_WRITE

This event is triggered by writes caused by a D128 translation table walk. Fields:

ABORTED bool

Was the walk successful.

DATA_HIGH unsigned int

The data upper bits written.

DATA_LOW unsigned int

The data lower bits written.

IPA unsigned int

For stage 1, the IPA of the write.

IS_HACDBS_ACCESS bool

If the access was due to HACDBS_CLEAN.

LEVEL unsigned int

Translation table level.

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

OUTADDR unsigned int

The output physical address.

PADDR unsigned int

The physical address of the write.

PASmem enum

Physical Address Space of the accessed memory.

PASreq enum

Physical Address Space of the requested lookup.

REGIME_EL enum

Entry matches in this translation regime.

SIDE enum

Inst / Data.

STAGE unsigned int

Translation stage.

MMU_TRANS

Address translation information. Fields:

ASID unsigned int

Address space identifier.

CORE_NUM unsigned int

Core number in a multi processor.

Hyp bool

Entry matches in Hyp state only.

INNERCACHE_RA bool

Is the inner cache allocate on read.

INNERCACHE_TRANSIENT bool

Is the inner write-through transient.

INNERCACHE_TYPE enum

Inner Caching scheme (NC/MB/WA).

INNERCACHE_WA bool

Is the inner cache allocate on write.

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MEMTYPE enum

Memory type.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC enum

Is secure side supposed to access secure or nonsecure memory.

OUTERCACHE_RA bool

Is the outer cache allocate on read.

OUTERCACHE_TRANSIENT bool

Is the outer write-through transient.

OUTERCACHE_TYPE enum

Outer Caching scheme (NC/MB/WA).

OUTERCACHE_WA bool

Is the outer cache allocate on write.

PADDR unsigned int

Physical address of the access.

PAGESIZE unsigned int

Page size as log2(size).

PAS enum

PAS supposed to be accessed is secure or nonsecure memory or Root/Realm if RME is implemented.

SH enum

Shareability.

SIDE enum

Inst / Data.

STAGE1_PERM unsigned int

Stage 1 permissions, [5:3] Privileged access (XWR) [2:0] User access (XWR).

STAGE2_PERM unsigned int

Stage 2 permission mask [5:3] Privileged access (XWR) [2:0] User access (XWR).

VADDR unsigned int

Virtual address of the access.

VMID unsigned int

Virtual machine identifier.

XS bool

XS Attribute.

nG enum

Flag indicating whether ASID will be matched.

MMU_TTB_READ

This event is triggered by reads caused by a translation table walk. Fields:

ABORTED bool

Was the walk successful.

DATA unsigned int

The data read.

DESC_KIND enum

The encoding scheme of the descriptors used for this walk.

IPA unsigned int

For stage 1, the IPA of the read.

LEVEL unsigned int

Translation table level.

LPAE bool

Is this for an LPAE translation (*DEPRECATED*, use DESC_KIND).

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSmem bool

Non secure memory access.

NSreq bool

Non secure request.

OUTADDR **unsigned int**

The output physical address.

PADDR **unsigned int**

The physical address of the read.

PASmem **enum**

Physical Address Space of the accessed memory.

PASreq **enum**

Physical Address Space of the requested lookup.

REGIME_EL **enum**

Entry matches in this translation regime.

REQUESTER **enum**

The requester of this table walk.

SIDE **enum**

Inst / Data.

STAGE **unsigned int**

Translation stage.

MMU_TTB_WRITE

This event is triggered by writes caused by a translation table walk. Fields:

ABORTED **bool**

Was the walk successful.

DATA **unsigned int**

The data written.

DESC_KIND **enum**

The encoding scheme of the descriptors used for this walk.

IPA **unsigned int**

For stage 1, the IPA of the write.

IS_HACDBS_ACCESS **bool**

If the access was due to HACDBS_CLEAN.

LEVEL **unsigned int**

Translation table level.

LPAAE **bool**

Is this for an LPAE translation (*DEPRECATED*, use DESC_KIND).

MECID **unsigned int**

Memory Encryption Context(MEC) ID, if applicable.

MPAM_PARTID **unsigned int**

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSmem bool

Non secure memory access.

NSreq bool

Non secure request.

OUTADDR unsigned int

The output physical address.

PADDR unsigned int

The physical address of the write.

PASmem enum

Physical Address Space of the accessed memory.

PASreq enum

Physical Address Space of the requested lookup.

REGIME_EL enum

Entry matches in this translation regime.

SIDE enum

Inst / Data.

STAGE unsigned int

Translation stage.

MODE_CHANGE

Mode change. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

MODE enum

The new mode.

NON_SECURE enum

The core's new non-secure bit.

OLD_MODE enum

The old mode.

SECURITY_STATE enum

The new security state.

NOT_TAG_CHECKED_INST

MTE Tag not checked for current instruction.

PERIODIC

Called for every quantum. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The instruction count of this CPU.

PC unsigned int

The address of the next instruction to be executed on this CPU.

PMU_COUNTER_OVERFLOW

PMU counter overflow. Fields:

COUNTER_GROUP enum

Counter group.

COUNTER_RANGE enum

Counter range.

EVENT_ID unsigned int

Event Identifier, field is valid only for Event counter group.

INDEX unsigned int

Counter index (as selected by PMSELR), field is valid only for Event/Cycle counter groups.

INTERRUPT bool

Is interrupt enabled on overflow for this counter.

POINTER_AUTH

Called when execution hits a pointer authentication check. Fields:

AUTH_STATE bool

Authentication result.

KEY_USED enum

Pointer authentication key.

POTINDEX_UPDATE

Value of POT index for current pc changed. Fields:

NEW_POTINDEX unsigned int

New POTIndex.

OLD_POTINDEX unsigned int

Old POTIndex.

PREFETCH_MEMORY64

Prefetch from PRFM or PRFUM instructions. Fields:

PRFOP enum

Prefetch hint.

VADDR unsigned int

Virtual address of the location that should be prefetched.

PRELOAD64_REQUEST

Preload requests from PRFM or PRFUM instructions. Fields:

PRFOP enum

Prefetch hint.

VADDR unsigned int

Virtual address of the location that should be prefetched.

PRELOAD_DATA

Data preload from PLD instruction. Fields:

VADDR unsigned int

Virtual address of the data that should be preloaded.

PRELOAD_INST

Instruction preload from PLI instruction. Fields:

VADDR unsigned int

Virtual address of the instruction that should be preloaded.

PRE_CORE_LOAD

Trace just before a core load. Fields:

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

PRE_CORE_STORE

Trace just before a core store. Fields:

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

PRE_TTB_READ

This event is triggered before reads caused by a translation table walk. Fields:

ADDR unsigned int

The physical address of the read.

PSTATE

PSTATE fields update. Fields:

ALLINT unsigned int

All interrupt mask bit.

BTYPE unsigned int

Branch target identification bit.

CORE_NUM unsigned int

Core number in a multi processor.

DAIF unsigned int

Exception mask bits.

DIT unsigned int

Data Independent Timing bit.

EL unsigned int

Current Exception level field.

EXLOCK unsigned int

Exception return state lock bit.

IL unsigned int

Illegal Execution state bit.

NZCV unsigned int

Condition flags.

PACM unsigned int

PACM bit.

PAN unsigned int

Privileged Access Never state bit.

PM unsigned int

PMU exception mask bit.

PPEND unsigned int

PMU exception pending bit.

SM unsigned int

StreamingMode status bit.

SP unsigned int

Stack pointer register selection bit.

SS unsigned int

Software Step bit.

SSBS unsigned int

Speculative Store Bypass Safe bit.

TCO unsigned int

Tag Check Override bit.

TOUCHED unsigned int

Mask of the fields that have been touched since this trace has last been emitted. Note that touched does not necessarily mean the value changed. The index of the field is the bit index in the mask plus the index of this field plus one.

UAO unsigned int

User Access Override bit.

UINJ unsigned int

Undefined Instruction exceptions bit.

ZA unsigned int

ZA storage and ZT0 register enable bit.

nRW unsigned int

Current Execution state bit.

PSTATE_PACTIVE_OPMODE_TRANSITION

Trace pactive and Opmode mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Operating mode after the transition.

OLD_STATE enum

Operating mode before the transition.

PSTATE_PACTIVE_OPMODE_TRANSITION_DENIED

Trace the denied pactive and Opmode mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Operating mode after the transition.

OLD_STATE enum

Operating mode before the transition.

REASON string

Reason for the transition being denied.

PSTATE_PACTIVE_POWER_TRANSITION

Trace pactive and power mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Power mode after the transition.

OLD_STATE enum

Power mode before the transition.

PSTATE_PACTIVE_POWER_TRANSITION_DENIED

Trace the denied pactive and power mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Power mode after the transition.

OLD_STATE enum

Power mode before the transition.

REASON string

Reason for the transition being denied.

PSTATE_PACTIVE_THREAD_OPMODE_TRANSITION

Trace pactive and Opmode mode transitions for threads on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Operating mode after the transition.

OLD_STATE enum

Operating mode before the transition.

QUANTUM_START

Called at the start of every quantum.

RANGE_PRELOAD_REQUEST

Preload requests from RPRFM. Fields:

COUNT unsigned int

Number of blocks of data to access.

LENGTH unsigned int

Number of contiguous bytes to be accessed in each block.

OP unsigned int

Range prefetch operation .

REUSE unsigned int

Maximum number of bytes to be accessed before next RPRFM .

STRIDE signed int

Stride to advance after accessing LENGTH bytes.

VADDR unsigned int

Base virtual address of range prefetch request.

RUN_STATE

Run state transition. Fields:

INST_COUNT **unsigned int**

Ticks count at point of transition.

NEW **enum**

New run state.

OLD **enum**

Old run state.

SEMIHOSTING_CALL

Call of a semihost function occurred. Fields:

PC **unsigned int**

The program counter after the semihosting call.

SEMIHOSTING_PRECALL

About to call semihost. Fields:

EL2 **bool**

Use EL2 translation regime.

NS **bool**

Is Non-Secure.

REG_WIDTH **unsigned int**

The current register width in bytes.

SIGNAL

External signal state change. Fields:

SIGNAL **enum**

Signal that changed.

STATE **bool**

Signal asserted state.

SOFTWARE_STEP

Return the Debug Software Step state. Fields:

STATE **enum**

Software Step state.

SPSR

SPSR change. Fields:

CORE_NUM **unsigned int**

Core number in a multi processor.

MODE **enum**

Which of the banked SPSR registers is written.

OLD_VALUE unsigned int

The old SPSR value.

VALUE unsigned int

The new SPSR value.

START_COMPILE

Compilation started. Fields:

VADDR unsigned int

Instruction where compilation begins.

STG_MULTIPLE_MICRO_OPS_ADDRESS_RANGE

Address range for MTE stores with multiple uops. Fields:

ADDR_END unsigned int

Final address.

ADDR_START unsigned int

Starting address.

STORE_MULTIPLE_REGS_ADDRESS_RANGE

Address range for stores with multiple registers. Fields:

ADDR_END unsigned int

Final address.

ADDR_START unsigned int

Starting address.

SUBOPTIMAL_LDST_RETIRED

suboptimal load/store. Fields:

ACCESS_SIZE unsigned int

Log2 of the access width in bytes used for alignment checking (1: half-word, 2: word ...).

OPTIMAL_ALIGN_SELECTOR unsigned int

Selector to decide the optimal alignment size (1: access width, 2: 4-bytes, 3: 8-bytes, ..., 12: 4-kbytes).

VADDR unsigned int

The virtual address of the access.

SVE_INST_SPEC

SVE operations speculatively executed.

SVE_LD_RETIRED

SVE load instructions architecturally executed.

SVE_LOADS

SVE load accesses. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ACQREL enum

Is this an acquire/release.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DATA unsigned int

The data read or written.

ELEMENT_SIZE unsigned int

Width of each element.

ENDIANNESS enum

Byte ordering of this access.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

LOCK enum

Normal, exclusive or locked access.

MECID unsigned int

Memory Encryption Context ID, if applicable.

MECID2 unsigned int

Memory Encryption Context ID of the second page, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PADDR_TAG unsigned int

If tag checked, the physical address tag for the access.

PAGE_SIZE unsigned int

Page size as $\log_2(\text{size in bytes})$.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK, 2=Exclusive Failed, 3=Unsupported.

SIZE unsigned int

Width of the access in bytes. Only required if DATA is not traced.

TAG_CHECKED bool

Is this access tag checked.

TAG_CHECKED2 bool

Is the second page access tag checked.

TRANS bool

Is this a translated access.

VADDR unsigned int

The virtual address of the access.

SVE_MEM_PREFETCH

Scalable Vector Extension memory prefetch. Fields:

CACHE_LEVEL enum

The preferred cache level.

HINT enum

What the memory is expected to be used for.

SIZE unsigned int

The prefetch size in bytes.

TEMPORAL bool

Whether the data is temporal.

VADDR unsigned int

The virtual address of the prefetch.

SVE_REG_READ

Scalable Vector Extension register read. Fields:

ID enum

The register identifier.

SM bool

Whether the PE is in Streaming Mode.

VALUE unsigned int

The read data.

SVE_REG_UPDATE

Scalable Vector Extension register update. Fields:

ID enum

The register identifier.

SM bool

Whether the PE is in Streaming Mode.

VALUE unsigned int

The written data.

SVE_REG_WRITE

Scalable Vector Extension register write. Fields:

ID enum

The register identifier.

SM bool

Whether the PE is in Streaming Mode.

VALUE unsigned int

The written data.

SVE_STORES

SVE store accesses. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ACQREL enum

Is this an acquire/release.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DATA unsigned int

The data read or written.

ELEMENT_SIZE unsigned int

Width of each element.

ENDIANNESS enum

Byte ordering of this access.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

LOCK enum

Normal, exclusive or locked access.

MECID unsigned int

Memory Encryption Context ID, if applicable.

MECID2 unsigned int

Memory Encryption Context ID of the second page, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PADDR_TAG unsigned int

If tag checked, the physical address tag for the access.

PAGE_SIZE unsigned int

Page size as $\log_2(\text{size in bytes})$.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK, 2=Exclusive Failed, 3=Unsupported.

SIZE unsigned int

Width of the access in bytes. Only required if DATA is not traced.

TAG_CHECKED bool

Is this access tag checked.

TAG_CHECKED2 bool

Is the second page access tag checked.

TRANS bool

Is this a translated access.

VADDR unsigned int

The virtual address of the access.

SVE_ST_RETIRED

SVE store instructions architecturally executed.

SVE_UNALIGNED_LDST_RETIRED

SVE load or store instructions architecturally executed that would cause alignment fault if A bit is set.

SVE_Z_PRED_MEM_ACCESS_INFO

Information about SVE/SME predicated accesses to Z registers (non Gather/Scatter). Fields:

MEM_ELEM_SIZE unsigned int

Width of elements accessed in memory (in bytes).

RnW bool

Whether this is a Read not a Write.

VA unsigned int

Virtual base address computed for accesses.

SYNC

Called for every synchronization, that is every quantum break. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The instruction count of this CPU.

LOCAL_QUANTUM unsigned int

The local quantum of this CPU.

LOCAL_TIME unsigned int

The local time of this CPU.

SYNC_REASON enum

Reason for sync.

SYSCALL

System call instruction executed. Fields:

IMM unsigned int

Immediate value of the system call instruction.

TYPE enum

System call type.

VADDR unsigned int

Instruction that caused the system call.

SYSREG_READ128

System Coprocessor 128 bit register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

REGNUM enum

Internal register number.

REGNUM_PHYS enum

Internal register number physically accessed (e.g. in presense of VHE redirection).

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_READ64

System Coprocessor register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REGNUM enum

Internal register number.

REGNUM_OPERAND enum

Register number corresponding to the instruction operand.

REGNUM_PHYS enum

Internal register number physically accessed (e.g. in presense of VHE redirection).

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_UPDATE128

Triggers when the system updates a register. Fields:

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG enum

Register number.

UNKNOWN unsigned int

Bits of the register which became unknown.

VALUE unsigned int

Value written to the register.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_UPDATE32

Triggers when the system updates a register. Fields:

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG enum

Register number.

UNKNOWN unsigned int

Bits of the register which became unknown.

VALUE unsigned int

Value written to the register.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_UPDATE64

Triggers when the system updates a register. Fields:

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG enum

Register number.

UNKNOWN unsigned int

Bits of the register which became unknown.

VALUE unsigned int

Value written to the register.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_WRITE128

System Coprocessor 128 bit register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

REGNUM enum

Internal register number.

REGNUM_PHYS enum

Internal register number physically accessed (e.g. in presense of VHE redirection).

UNDEF bool

The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int
opcode 2.

SYSREG_WRITE64

System Coprocessor register write. Fields:

CORE_NUM unsigned int
Core number in a multi processor.

CRm unsigned int
CRm.

CRn unsigned int
CRn.

NS bool
Secure or nonsecure banked register is accessed.

REGNUM enum
Internal register number.

REGNUM_OPERAND enum
Register number corresponding to the instruction operand.

REGNUM_PHYS enum
Internal register number physically accessed (e.g. in presense of VHE redirection).

UNDEF bool
The register accessed is undefined.

UPDATED_VALUE unsigned int
Updated value of the register now it has been written.

VALUE unsigned int
The new value written.

opc unsigned int
opcode 1.

opc0 unsigned int
opcode 0.

opc2 unsigned int
opcode 2.

TRBU_BUFFER_WRAP

DISPLAY Trace buffer current write pointer wrapped.

TRBU_TRIGGER_EVENT

DISPLAY Trace buffer Trigger Event.

TRBU_WRITE

DISPLAY PETU trace data has been written to the TRBU memory buffer at `%{BUFFER_ADDRESS}` with size `%{SIZE}` and data `%{DATA}`. Fields:

BUFFER_ADDRESS unsigned int

Virtual Address of start of buffer write.

DATA unsigned int

Data written to the buffer.

SIZE unsigned int

Size of sample written to buffer, in bytes.

UNALIGNED_LDST_RETIRED

Processor unaligned load/store. Fields:

ACCESS_SIZE unsigned int

Log2 of the access width in bytes used for alignment checking (1=>half-word, 2->word ...).

NUMBER_OF_BEATS unsigned int

Number of accesses (beats) in this burst (Total size in bytes is ACCESS_SIZE * NUMBER_OF_BEATS).

VADDR unsigned int

The virtual address of the access.

VFP_D_REGS

VFP/NEON D 64 bit register write. Fields:

ALIASING enum

Type of register aliasing used.

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a write to the corresponding bit in the register. For example, 0x1 would indicate a write to VALUE[0].

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

VFP_D_REGS_READ

VFP/NEON D 64 bit register read. Fields:

ID unsigned int

The register number.

VALUE unsigned int

The current value of the register.

VFP_Q_REGS

VFP/NEON Q 128 bit register write. Fields:

ALIASING enum

Type of register aliasing used.

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a write to the corresponding byte in the register. For example, 0x1 would indicate a write to VALUE[7:0].

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

VFP_Q_REGS_READ

VFP/NEON Q 128 bit register read. Fields:

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a read to the corresponding byte in the register. For example, 0x1 would indicate a read to VALUE[7:0].

VALUE unsigned int

The current value of the register.

VFP_SYS_REGS

Writes to the VFP/NEON units system registers. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ID enum

Which VFP system register is written.

MASK unsigned int

The mask indicating updated cumulative bits.

OLD_VALUE unsigned int

The register's old value overwritten.

VALUE unsigned int

The new value written to the VFP system register.

VFP_S_REGS

VFP/NEON S 32 bit register write. Fields:

ALIASING enum

Type of register aliasing used.

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a write to the corresponding bit in the register. For example, 0x1 would indicate a write to VALUE[0].

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

VFP_S_REGS_READ

VFP/NEON S 32 bit register read. Fields:

ID unsigned int

The register number.

VALUE unsigned int

The current value of the register.

VIRTUAL_TAG_ALLOCATED

Virtual tag information when allocated. Fields:

TAG_VADDR unsigned int

Virtual tag virtual address.

VADDR unsigned int

Data Virtual address.

VIRTUAL_TAG unsigned int

Virtual tag value.

WAYPOINT

Signals end of basic block execution. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ISSET enum

Origin instruction set.

IS_COND bool

Indicates if this is a conditional waypoint.

PC unsigned int

Origin address (or 0 if unavailable).

TAKEN bool

Indicates if this waypoint was taken.

TARGET unsigned int

Destination address.

TARGET_ISET enum

Destination instruction set.

WFE_END

WFE ended. Fields:

INST_COUNT unsigned int

Ticks count when leaving WFE.

WFE_EVENT_REGISTER

WFE event register status: set/clear, reason. Fields:

INST_COUNT unsigned int

Ticks count.

REASON enum

Reason for set/clear. Only REASON==0 (Cleared by WFE) clears the bit, all other reasons set it.

WFE_IGNORED

WFE ignored. Fields:

EVENT bool

This WFE was ignored because the event register was set.

INST_COUNT unsigned int

Ticks count when ignoring WFE.

TRAPPED bool

This WFE was trapped.

WFE_START

WFE entered. Fields:

INST_COUNT unsigned int

Ticks count when entering WFE.

WFI_END

WFI ended. Fields:

INST_COUNT unsigned int

Ticks count when leaving WFI.

WFI_IGNORED

WFI ignored. Fields:

DISABLED bool

This WFI was ignored because WFI is disabled.

INST_COUNT unsigned int

Ticks count when ignoring WFI.

REASON enum

specifies reason why WFI trace was ignored.

TRAPPED bool

This WFI was trapped.

WFI_START

WFI entered. Fields:

INST_COUNT unsigned int

Ticks count when entering WFI.

WFI_WAKEUP

WFI wakeup. Fields:

INST_COUNT unsigned int

Ticks count when WFI wakeup occurred.

REASON enum

Reason for wakeup.

2.28 ARM_Cortex-A720AE

This section describes the trace sources.

AA64_ASE_SVE_REGS

Modification of AArch64 Advanced SIMD or SVE Vector Registers (via SIMD or Scalar operations), SVE Predicate Registers and ZT0. Fields:

ID enum

The register identifier.

MASK unsigned int

Bitmask of the register to signal the modified bits in the VALUE field.

SM bool

Whether the PE is in Streaming Mode.

VALUE unsigned int

Value of AArch64 vector register.

ASYNC_MEMORY_FAULT

Context ID Register write. Fields:

FAULT unsigned int

Fault status present in ESR.ISS encoding for Serror in ARCH64 and present in DFSR encoding for external abort format in ARCH32.

PADDR unsigned int

Physical Address (or 0 if unavailable).

VADDR signed int

Virtual Address (or 0 if unavailable).

ATOMIC_END_ACCESS

Bus trace access for atomics. Fields:

ACCESS_FAIL bool

Memory access failed.

ADDR unsigned int

The virtual address of the access.

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

COMPARE_VALUE unsigned int

Compare value for CAS.

LOAD_VALUE unsigned int

Loaded values.

NSDESC unsigned int

The security state of the access.

OPERAND_VALUE unsigned int

Operation's operand.

OPERATION enum

Operation type.

PADDR unsigned int

The physical address of the access.

PRIV bool

Is this a privileged access?.

ATOMIC_START_ACCESS

Bus trace access for atomics. Fields:

ADDR unsigned int

The virtual address of the access.

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

COMPARE_VALUE unsigned int

Compare value for CAS.

NSDESC unsigned int

The security state of the access.

OPERAND_VALUE unsigned int

Operation's operand.

OPERATION enum

Operation type.

PADDR unsigned int

The physical address of the access.

PRIV bool

Is this a privileged access?.

ArchMsg.Error.exit_code

DISPLAY exit code: %{EXITCODE} component kind: %{KIND} reason: %{REASON}. Fields:

EXITCODE signed int

Exit code to be returned.

KIND string

Component kind that invoked the exit code trace.

REASON string

Reason why the exit code trace was invoked (optional).

ArchMsg.Warning.IgnoredWarmResetRequest

Event fires when software attempts to request warm reset through write to deprecated DBGPRCR.CWRR.

ArchMsg.Warning.RAS_ErrselrUnpredictableState

DISPLAY ERRSELR is in unpredictable state due to value greater than number_of_error_records written to it.

ArchMsg.Warning.RAS_UnpredictableStateWrite

DISPLAY %{REG_NAME} write ignored due to unpredictable ERRSELR state. Fields:

REG_NAME string

Name of register being written.

ArchMsg.Warning.ReportedRevisionRegisterRead

DISPLAY Reported revision and patch do not match that of the cpu. Fields:

ORIGINAL_VALUE unsigned int

The value the register would have based on the modelled revision.

REGISTER string

The name of the register that has ben read.

REPORTED_VALUE unsigned int

The value the register has been overridden to.

ArchMsg.Warning.Unpredictably Indexed PM Event Register

TAGS **UNPREDICTABLE** DISPLAY Event %{IsDirect:(Selector value|Register)} %{SEL} is >= %{N}. Fields:

IsDirect unsigned int

Direct Access rather than indirect.

N unsigned int

Number of Accessible Registers.

SEL unsigned int

Selector.

ArchMsg.Warning.cache_contents_unknown

A cache has been activated without being invalidated and may contain **UNKNOWN** content. DISPLAY %{SIDE:(D|I)}-Cache was enabled in %{REGIME:(SECURE|NON_SECURE|HYP)} regime but was not invalidated since power-on: cache lines could contain **UNKNOWN** content %{SIDE:(Note: D-side caches must invalidate SECURE lines even if the cache is only used by NON_SECURE code in order to ensure that unexpected dirty lines tagged SECURE are not naturally evicted)}. Fields:

REGIME unsigned int

Current translation regime.

SIDE unsigned int

0 d-side cache or unified cache, 1 i-side.

ArchMsg.Warning.dcimvac_matches_watchpoint

TAGS **UNPREDICTABLE** DISPLAY Watchpoints matching an AArch32 DCIMVA are implementation defined.

ArchMsg.Warning.decode_fieldmismatch

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with different fields %{FIELD1} and %{FIELD2}. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_frachitsnegative

TAGS **UNPREDICTABLE** DISPLAY VCVT Instruction is unpredictable with a negative number of fraction bits.

ArchMsg.Warning.decode_initblock

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable in an IT block.

ArchMsg.Warning.decode_initblocknotlast

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable in an IT block when not the last.

ArchMsg.Warning.decode_invalid_condition

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable when the condition is not AL.

ArchMsg.Warning.decode_invalidfieldcombination

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with the values in %{FIELD1} and %{FIELD2}. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_invalidvalue

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with the value in %{FIELD}. Fields:

FIELD enum

Field name.

ArchMsg.Warning.decode_mem_hint_unallocated

TAGS **UNPREDICTABLE** DISPLAY There is no memory hint allocated to this bit pattern.

ArchMsg.Warning.decode_registermatch

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with %{FIELD1} and %{FIELD2} as the same register. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_registermismatch

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with %{FIELD1} and %{FIELD2} as different registers. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_registeroutofrange

TAGS **UNPREDICTABLE** DISPLAY Floating-point instruction is unpredictable indexing beyond the end of the register bank.

ArchMsg.Warning.decode_sbzsbo

TAGS **UNPREDICTABLE** DISPLAY Reserved bits in the instruction are not canonical.

ArchMsg.Warning.decode_transactiontoobig

TAGS **UNPREDICTABLE** DISPLAY Attempt to perform a memory transaction of more than 128 bytes.

ArchMsg.Warning.decode_unpred_other

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable.

ArchMsg.Warning.decode_unpreduseofpc

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable when %{FIELD} is PC. Fields:

FIELD enum

Field name.

ArchMsg.Warning.decode_unpreduseofr13

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable when %{FIELD} is R13. Fields:

FIELD enum

Field name.

ArchMsg.Warning.decode_unpreduseofr13inreglist

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with R13 in register-list.

ArchMsg.Warning.decode_useofd16orhigher

TAGS **UNPREDICTABLE** DISPLAY Attempt to use register D16 or higher where it is not allowed.

ArchMsg.Warning.decode_writebackandbaseinlist

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with write back when the base register is in the transfer list.

ArchMsg.Warning.decode_zeroregistersinlist

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable when no registers are to be transferred.

ArchMsg.Warning.misaligned_ptrs

DISPLAY TRBPTR_EL1(%{TRBPTR_EL1}) or TRBTRG_EL1(%{TRBTRG_EL1}) are not aligned as per TRBIDR_EL1.Align(%{TRBIDR_EL1}). Fields:

TRBIDR_EL1 unsigned int

TRBIDR_EL1.Align value.

TRBPTR_EL1 unsigned int

Current write pointer.

TRBTRG_EL1 unsigned int

Trigger pointer.

ArchMsg.Warning.out_of_range_ptrs

DISPLAY TRBPTR_EL1(%{TRBPTR_EL1}) is out of range as per TRBBASER_EL1(%{TRBBASER_EL1}) or TRBLIMITR_EL1.LIMIT(%{TRBLIMITR_EL1}). Fields:

TRBBASER_EL1 unsigned int

Base pointer.

TRBLIMITR_EL1 unsigned int

Limit pointer.

TRBPTR_EL1 unsigned int

Current write pointer.

ArchMsg.Warning.recursive_branch

An instruction is performing a branch that targets the same instruction. If this was intended then a WFI might be more effective. DISPLAY recursive branch detected at PC=%{PC}. PRIMARY KEY PC. Fields:

PC unsigned int

Address of instruction causing the branch.

ArchMsg.Warning.recursive_exception

An instruction has generated an exception that targets the same instruction. DISPLAY recursive %{TYPE} exception detected at PC=%{PC} (CPSR=%{CPSR}). PRIMARY KEY PC. Fields:

CPSR unsigned int

Processor state.

PC unsigned int

Address of instruction causing the exception.

TYPE enum

Type of exception.

ArchMsg.Warning.reentrant_vector_catch

TAGS UNPREDICTABLE DISPLAY Vector catch debug event on Prefetch or Data abort vector %{ADDR}. Fields:

ADDR unsigned int

vector address.

ArchMsg.Warning.reserved_it_state

TAGS UNPREDICTABLE DISPLAY Execution with reserved IT state %{ITSTATE}. Fields:

ITSTATE unsigned int

reserved IT state.

ArchMsg.Warning.tlb_contents_unknown

Translation has been enabled without invalidating the TLB. DISPLAY Translation was enabled in the %{PAS:(Secure|Non-secure|Root|Realm)} EL%{EL:d} translation regime (vmid=

{VMID:x}) without invalidating {INVALIDITY:(|d-side |i-side |)} entries since power-on: {INVALIDITY:(|D||)} TLB could contain **UNKNOWN** content. Fields:

EL unsigned int

Exception level of translation regime.

INVALIDITY unsigned int

Which TLB may be invalid (if TLBs are separate).

NON_SECURE bool

Whether the translation regime is non-secure.

PAS unsigned int

Physical address space of translation regime.

VMID unsigned int

Current VMID, if applicable.

ArchMsg.Warning.unknown_DLR_DSPSR

Exiting debug state with unknown DLR or DSPSR.

ArchMsg.Warning.unknown_ELR_SPSR

Returning from debug exception with unknown ELR or SPSR.

ArchMsg.Warning.unpred_FAR_on_LRCPC3_non_atomic_mem_access

TAGS **UNPREDICTABLE** Unpredictable FAR/FSTATUS on LRCPC3 pair instructions resulting in two separate memory accesses
DISPLAY Unpredictable FAR on LRCPC3 non atomic memory access at {ADDR1} with FSTATUS {FSTATUS1} or {ADDR2} with FSTATUS {FSTATUS2}. Fields:

ADDR1 unsigned int

Possible FAR address 1.

ADDR2 unsigned int

Possible FAR address 2.

FSTATUS1 unsigned int

Possible fault status value 1.

FSTATUS2 unsigned int

Possible fault status value 2.

ArchMsg.Warning.unpred_async_tag_check_fail_on_sve_speculative_load_do_update_ffr

Unpredictable set to FALSE of the FFR element associated to SVE speculative load with an async tag check fail TAGS **UNPREDICTABLE**.

ArchMsg.Warning.unpredictable_a32_breakpoint

TAGS **UNPREDICTABLE** DISPLAY Unpredictable breakpoint on a A32 instruction at {ADDR1} Offset {OFFSET} BAS {BAS} for a {IS_ADDRESS_MISMATCH_BREAKPOINT:(Mismatch|Match)} Breakpoint number {N:d} Breakpoint address {ADDR2}. Fields:

ADDR1 unsigned int

Address.

ADDR2 unsigned int

Breakpoint address.

BAS unsigned int

BAS field.

IS_ADDRESS_MISMATCH_BREAKPOINT bool

Is a Mismatch breakpoint, not a Match breakpoint.

N unsigned int

Breakpoint number.

OFFSET unsigned int

Always has a value of zero.

ArchMsg.Warning.unpredictable_banked_register_access

TAGS **UNPREDICTABLE** DISPLAY Access to banked register (R=%{R:d} SYSm=%{SYSM:2x}) is unpredictable. Fields:

R unsigned int

R bit.

SYSM unsigned int

SYSm.

ArchMsg.Warning.unpredictable_ccfail_undef

TAGS **UNPREDICTABLE** DISPLAY An instruction %{OPCODE} at VA=%{VADDR} failed its condition codes check but would otherwise have caused a non-data-dependent trap or undefined exception. Behaviour is implementation-defined choice of exception or nop. Fields:

OPCODE unsigned int

opcode of the instruction.

VADDR unsigned int

Virtual address of the instruction being executed.

ArchMsg.Warning.unpredictable_ici_bits

TAGS **UNPREDICTABLE** DISPLAY ICI bits do not apply or are architecturally unpredictable for instruction.

ArchMsg.Warning.unpredictable_t32_breakpoint

TAGS **UNPREDICTABLE** DISPLAY Unpredictable breakpoint on a T32 instruction at %{ADDR1} Offset %{OFFSET} BAS %{BAS} for a %{IS_ADDRESS_MISMATCH_BREAKPOINT:(Mismatch|Match)} Breakpoint number %{N:d} Breakpoint address %{ADDR2}. Fields:

ADDR1 unsigned int

Address.

ADDR2 unsigned int

Breakpoint address.

BAS unsigned int

BAS field.

IS_ADDRESS_MISMATCH_BREAKPOINT bool

Is a Mismatch breakpoint, not a Match breakpoint.

N unsigned int

Breakpoint number.

OFFSET unsigned int

Indicates which halfword of a 32-bit Thumb instruction the breakpoint is set on. 0 for the first halfword or if the instruction is 16 bits wide, or 1 for the second halfword.

ArchMsg.Warning.unpredictable_t32_vector_catch

TAGS **UNPREDICTABLE** DISPLAY Unpredictable Vector catch exception on a T32 instruction at %{ADDR} Offset %{OFFSET}. Fields:

ADDR unsigned int

Address.

OFFSET unsigned int

Indicates which halfword of a 32-bit Thumb instruction the vector catch exception is set on. 0 for the first halfword or if the instruction is 16 bits wide, or 1 for the second halfword.

ArchMsg.Warning.unpredictable_watchpoint_far

TAGS **UNPREDICTABLE** DISPLAY The value for FAR/EDWAR for hitting this watchpoint may be anywhere in the range %{LOWER_BOUND} to %{UPPER_BOUND}. Fields:

LOWER_BOUND unsigned int

the lowest address accessed by the instruction that triggered the watchpoint.

UPPER_BOUND unsigned int

the highest watchpointed address accessed by that instruction.

ArchMsg.Warning.unpredictable_watchpoint_far_isscmp

DISPLAY The value for FAR/EDWAR for hitting this watchpoint may be anywhere in the range %{LOWER_BOUND} to %{UPPER_BOUND}. Fields:

LOWER_BOUND unsigned int

the lowest address accessed by the instruction that triggered the watchpoint.

UPPER_BOUND unsigned int

the highest watchpointed address accessed by that instruction.

ArchMsg.Warning.warning_AdvSIMDExpandImmUnexpectedZero

TAGS **UNPREDICTABLE** DISPLAY AdvSIMDExpandImm may treat this immediate value as **UNPREDICTABLE**.

ArchMsg.Warning.warning_ConditionalSMC

TAGS **UNPREDICTABLE** DISPLAY SMC instruction has **UNPREDICTABLE** effects when conditional and when combined with traps.

ArchMsg.Warning.warning_TLBID_out_of_range

TLBID out of range in TLBI according to the configuration of NIS, NVIS, NOS and NVOS.
Fields:

NIS unsigned int

NIS.

NOS unsigned int

NOS.

NVIS unsigned int

NVIS.

NVOS unsigned int

NVOS.

TLBID unsigned int

The TLBID value used in the TLBI instruction.

ArchMsg.Warning.warning_TTL_or_TG_mismatches_current_translation_granule

TTL or TG mismatches current translation granule in TLBI VA/RVA/IPA/RIPA. Fields:

CURRENT_TG unsigned int

Current TG as indicated by TCR_ELx or VTCR_EL2.

TLBI_TG unsigned int

TG (or TTL[3:2]) used in TLBI instruction.

ArchMsg.Warning.warning_access_crosses_page_boundary

TAGS UNPREDICTABLE DISPLAY Access crosses page boundary.

ArchMsg.Warning.warning_access_crosses_page_boundary_has_fault

TAGS UNPREDICTABLE DISPLAY Access that crosses page boundary has fault(s).

ArchMsg.Warning.warning_access_crosses_page_boundary_has_fault_on_both_pages

TAGS UNPREDICTABLE DISPLAY Access that crosses page boundary has a fault on each page.
Fields:

ADDR1 unsigned int

FAR address on first page.

ADDR2 unsigned int

FAR address on second page.

FSTATUS1 unsigned int

Fault status on first page.

FSTATUS2 unsigned int

Fault status on second page.

ArchMsg.Warning.warning_access_crosses_page_boundary_spanning_different_memory

TAGS UNPREDICTABLE DISPLAY Access at address %{ADDR} crosses page from MEM_TYPE %{MEMTYPE_PAGE1} to %{MEMTYPE_PAGE2}. Fields:

ADDR unsigned int

address of access crossing page.

MEMTYPE_PAGE1 signed int

First Page's memory type (0-Normal, 1-Non-Normal).

MEMTYPE_PAGE2 signed int

Second Page's memory type (0-Normal, 1-Non-Normal).

ArchMsg.Warning.warning_access_wraps_around_memory

TAGS **UNPREDICTABLE** DISPLAY Access wraps around memory in %{WIDTH:d}bit mode.
Fields:

WIDTH unsigned int

Address width.

ArchMsg.Warning.warning_bcr_linking_status

TAGS **UNPREDICTABLE** DISPLAY DBGBCR.{BT2,BT} combination gives unpredictable behaviour.

ArchMsg.Warning.warning_bcr_mask_reserved

TAGS **UNPREDICTABLE** DISPLAY DBGBCR_ELn.MASK is set to a reserved value.

ArchMsg.Warning.warning_bvr_masked_bit_not_zero

TAGS **UNPREDICTABLE** DISPLAY DBGBVRn_EL1.VA contains a bit masked by DBGBCRn_EL1.MASK that is not zero.

ArchMsg.Warning.warning_ccsidr_unimplemented_level

TAGS **UNPREDICTABLE** DISPLAY CCSIDR read while CSSELR %{CSSELR:x} points to an unimplemented cache level. Fields:

CSSELR unsigned int

current effective value of CSSELR.

ArchMsg.Warning.warning_change_to_ns_when_tge_set

TAGS **UNPREDICTABLE** DISPLAY Attempting to change NS to 1 when HCR.TGE = 1.

ArchMsg.Warning.warning_contiguous_bit_check_abort

TAGS **UNPREDICTABLE** DISPLAY An abort occurred whilst attempting to check TLB entry is contiguous at %{ENTRY_ADDR}. Fields:

ENTRY_ADDR unsigned int

address of conflicting TLB entry.

ArchMsg.Warning.warning_contiguous_bit_error

TAGS **UNPREDICTABLE** DISPLAY TLB entry at %{ENTRY_ADDR} was not contiguous with entry at %{CONTIG_ADDR} Contents are %{ENTRY_DATA} but expected %{CONTIG_DATA}.
Fields:

CONTIG_ADDR unsigned int

address of the TLB first read, with which this entry is expected to be contiguous.

CONTIG_DATA unsigned int

expected contents based on the entry at CONTIG_ADDR.

ENTRY_ADDR unsigned int

address of conflicting TLB entry.

ENTRY_DATA unsigned int

contents of conflicting TLB entry.

ArchMsg.Warning.warning_contiguous_bit_error_D128

TAGS **UNPREDICTABLE** DISPLAY TLB entry at %{ENTRY_ADDR} was not contiguous with entry at %{CONTIG_ADDR} Contents are %{ENTRY_DATA_HI} %{ENTRY_DATA_LO} but expected %{CONTIG_DATA}. Fields:

CONTIG_ADDR unsigned int

address of the TLB first read, with which this entry is expected to be contiguous.

CONTIG_DATA unsigned int

expected contents based on the entry at CONTIG_ADDR.

ENTRY_ADDR unsigned int

address of conflicting TLB entry.

ENTRY_DATA_HI unsigned int

Higher bit contents of conflicting TLB entry.

ENTRY_DATA_LO unsigned int

Lower bit contents of conflicting TLB entry.

ArchMsg.Warning.warning_cp10_cp11_mismatch

TAGS **UNPREDICTABLE** DISPLAY Unpredictable configuration of CP10 and CP11 controls at EL %{EL:d}. Fields:

EL unsigned int

EL owning the control.

ArchMsg.Warning.warning_cp10_cp11_reserved_value

TAGS **UNPREDICTABLE** DISPLAY Unpredictable configuration of CPACR CP10 (%{CP10}) or CP11 (%{CP11}). Fields:

CP10 unsigned int

CP10 access.

CP11 unsigned int

CP11 access.

ArchMsg.Warning.warning_csselr_level_out_of_range

TAGS **UNPREDICTABLE** DISPLAY CSSELR.Level out of range, written value %{LEVEL} with only %{IMPLEMENTED:d} levels implemented. Fields:

IMPLEMENTED unsigned int

number of cache levels implemented.

LEVEL unsigned int

written value of CSSELR.Level.

TYPE enum

The type of cache selected by CSSELR.InD and TnD.

ArchMsg.Warning.warning_debug_flow_control_bits_not_obeyed

TAGS **UNPREDICTABLE** DISPLAY Flow control bits for %{REGISTER} were not obeyed, causing %{ERROR}. Fields:

ERROR enum

type of error resulting.

REGISTER enum

register under consideration.

ArchMsg.Warning.warning_debug_register_access_during_reset

TAGS **UNPREDICTABLE** DISPLAY Unpredictable %{IS_WRITE:(read|write)} access to debug register offset %{OFFSET} during reset. Fields:

IS_WRITE unsigned int

Write Not Read.

OFFSET unsigned int

Register Offset.

ArchMsg.Warning.warning_decode_cps_inconsistent_fields

TAGS **UNPREDICTABLE** DISPLAY The mode setting fields in the CPS instruction are inconsistent.

ArchMsg.Warning.warning_decode_invalid_state

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable in current exception-level/security state.

ArchMsg.Warning.warning_default_cacheable_mmu_on

TAGS **UNPREDICTABLE** DISPLAY Default cacheable is enabled (HCR.DC = 1) while MMU is enabled.

ArchMsg.Warning.warning_default_cacheable_vmmu_off

TAGS **UNPREDICTABLE** DISPLAY Default cacheable is enabled (HCR.DC = 1) while HCR.VM = 0.

ArchMsg.Warning.warning_deprecated_wvr_bit_2_set

TAGS **UNPREDICTABLE** DISPLAY DBGWVR%{N:d}_EL1=%{WVR:h} has bit 2 set, deprecated in ARMv8. Fields:

N unsigned int

index.

WVR unsigned int

DBGWVR.

ArchMsg.Warning.warning_eret_while_software_step_active_pending

Missing ISB between setting MDSCR_EL1.SS and ERET TAGS **UNPREDICTABLE**.

ArchMsg.Warning.warning_exclusive_to_non_normal

TAGS **UNPREDICTABLE** DISPLAY Exclusive Access to Non-Normal Memory %{ADDR}. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_exclusive_to_non_writeback

TAGS **UNPREDICTABLE** DISPLAY Exclusive Access to Non-Writeback-Cacheable Memory %{ADDR}. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_execute_from_device_memory

TAGS **UNPREDICTABLE** DISPLAY An attempt was made to execute from Device memory.

ArchMsg.Warning.warning_illegal_cpsr_mode

TAGS **UNPREDICTABLE** DISPLAY Writing an illegal or unimplement mode (%{NEW_MODE:h}) to CPSR was unpredictable. Fields:

NEW_MODE unsigned int

New value of CPSR.M.

ArchMsg.Warning.warning_illegal_srs_mode

TAGS **UNPREDICTABLE** DISPLAY Illegal or **UNPREDICTABLE** mode (%{MODE:h}) used for SRS instruction. Fields:

MODE unsigned int

mode for Banked SP.

ArchMsg.Warning.warning_implementation_defined_read_debug_register_in_SCS

Read Access to Debug registers from the processor is IMPLEMENTATION DEFINED TAGS IMP_DEF. Fields:

PPB_OFFSET unsigned int

debug register offset.

ArchMsg.Warning.warning_implementation_defined_sequential_security_transitions_supported

The behavior of sequential instruction fetches that cross from non-secure to secure memory and contain an SG instruction is CONSTRAINED **UNPREDICTABLE** TAGS **UNPREDICTABLE**.

ArchMsg.Warning.warning_implementation_defined_stack_limit_check_supported

It is **IMPLEMENTATION DEFINED** whether stack pointer limit checking is performed for this instruction TAGS IMP_DEF. Fields:

ADDRESS unsigned int

Address of instruction.

OPCODE unsigned int

instruction opcode.

ArchMsg.Warning.warning_implementation_defined_taggednonsharedmemorytype

TAGS IMP_DEF DISPLAY Memory attribute has Tagged, non shared memory type; it is IMPLEMENTATION DEFINED whether the memory location is treated as Tagged or non-Tagged.

ArchMsg.Warning.warning_implementation_defined_write_debug_register_in_SCS

Write Access to Debug registers from the processor is IMPLEMENTATION DEFINED TAGS IMP_DEF. Fields:

DATA unsigned int

data attempted to be written.

PPB_OFFSET unsigned int

debug register offset.

ArchMsg.Warning.warning_invalid_tcr_granule

TAGS UNPREDICTABLE DISPLAY TCR.TG%{TG_ID:(0|1)} bits have been set to a granule size %{\$REQUEST:d}K not implemented - using %{\$SUBSTITUTE:d}K. Fields:

REQUEST signed int

page size requested (or 0 for reserved).

SUBSTITUTE signed int

best guess available page size.

TG_ID bool

bits TG0 or TG1.

ArchMsg.Warning.warning_load_multiple_user_registers_from_user_mode

TAGS UNPREDICTABLE DISPLAY An LDM(user registers) instruction executed from user mode.

ArchMsg.Warning.warning_mte_in_mem_attr_not_supported

TAGS UNPREDICTABLE DISPLAY Memory attribute %{\$ATTR} has Tagged memory type but Armv8.5-MTE is not implemented. Fields:

ATTR unsigned int

Memory attribute.

ArchMsg.Warning.warning_reserved_breakpoint_state_match

TAGS UNPREDICTABLE DISPLAY %{\$IS_BREAKPOINT:(Watchpoint|Breakpoint)} programmed with a reserved combination of HMC, SSC and %{\$IS_BREAKPOINT:(PAC|PMC)}. Fields:

IS_BREAKPOINT unsigned int

Is a breakpoint, not a watchpoint.

ArchMsg.Warning.warning_shareability

TAGS UNPREDICTABLE DISPLAY Unpredictable: combination of the 1st and 2nd stages of translation is Normal Inner Non-Cacheable, Outer Non-Cacheable.

ArchMsg.Warning.warning_software_step_set_while_enabled

TAGS UNPREDICTABLE DISPLAY MDSCR_EL1.SS set to 1 while software step debug exceptions are enabled.

ArchMsg.Warning.warning_thumb_instruction_wraps_around_memory

TAGS UNPREDICTABLE DISPLAY Thumb instruction wraps around memory.

ArchMsg.Warning.warning_ttbr_sbz_bits_are_not_zero

TAGS UNPREDICTABLE DISPLAY %{TTBR}[%{MSB:d}:0] should be zero, but are not. Fields:

MSB unsigned int

x-1.

TTBR enum

which TTBR register.

ArchMsg.Warning.warning_unaligned_address_dbgdtrrx_write

TAGS UNPREDICTABLE DISPLAY Unaligned address %{ADDR} in X0/R0 when DBGDTRRX written in memory access mode. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unaligned_address_dbgdtrtx_read

TAGS UNPREDICTABLE DISPLAY Unaligned address %{ADDR} in X0/R0 when DBGDTRTX read in memory access mode. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unaligned_to_device

TAGS UNPREDICTABLE DISPLAY Unaligned Access to Device Memory %{ADDR}. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unaligned_to_strongly_ordered

TAGS UNPREDICTABLE DISPLAY Unaligned Access to Strongly Ordered Memory %{ADDR}. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unknown_sau_rnr

TAGS UNPREDICTABLE DISPLAY SAU_RNR was set to an unsupported value.

ArchMsg.Warning.warning_unpredictable_AIRCR_PRIS_and_BFHFNMINs

TAGS UNPREDICTABLE DISPLAY The effect of setting both AIRCR.BFHFNMINs and AIRCR.PRIS to 1 is UNPREDICTABLE.

ArchMsg.Warning.warning_unpredictable_AIRCR_VECTCLRACTIVE_when_not_in_debug

TAGS **UNPREDICTABLE** DISPLAY The effect of writing a 1 to AIRCR.VECTCLRACTIVE if the processor is not halted in Debug state is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_AIRCR_VECTRESET_and_SYSRESETREQ

TAGS **UNPREDICTABLE** DISPLAY When the processor is halted in Debug state, if a write to the register writes a 1 to both VECTRESET and SYSRESETREQ, the behavior is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_AIRCR_VECTRESET_when_not_in_debug

TAGS **UNPREDICTABLE** DISPLAY The effect of writing a 1 to AIRCR.VECTRESET if the processor is not halted in Debug state is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_AIRCR_incorrect_VKEY

TAGS **UNPREDICTABLE** DISPLAY The value 0x05FA must be written to AIRCR.VKEY, otherwise the register write is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_EXC_RETURN_Reserved_Bit

EXC_RETURN[23:7] are reserved with the special condition that all bits should be written as one. Values other than all 1s are **UNPREDICTABLE**. TAGS **UNPREDICTABLE**. Fields:

EXC_RETURN unsigned int

Special PC value.

SBO_MASK unsigned int

Which bits Should Be One to not UNPRED.

ArchMsg.Warning.warning_unpredictable_PMEVTYPEPn_ELO_VS_reserved_value

TAGS **UNPREDICTABLE** DISPLAY Unpredictable reserved value of PMEVTYPEN<n>_ELO.VS.

ArchMsg.Warning.warning_unpredictable_change_to_DEMCR_MON_STEP_at_insufficient_priority

TAGS **UNPREDICTABLE** DISPLAY The effect of changing DEMCR.MON_STEP at an execution priority that is lower than the priority of the DebugMonitor exception is **CONSTRAINED UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_change_to_priority_of_active_exception

TAGS **UNPREDICTABLE** DISPLAY Changing the priority of an active exception is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_clear_lspact

TAGS **UNPREDICTABLE** DISPLAY **UNPREDICTABLE** state on context switch. Fields:

DRVEXC_TAKEN unsigned int

Derived exception taken, not pended.

REACHED_LAST_STORE unsigned int

Only the last faulted.

ArchMsg.Warning.warning_unpredictable_dbg_exit_unaligned_dlr

TAGS **UNPREDICTABLE** DISPLAY Debug exit to AArch32 with DLR[0] set to 1. PC[0] can be set to 0 or to DLR[0].

ArchMsg.Warning.warning_unpredictable_exception_catch

TAGS **UNPREDICTABLE** DISPLAY Generation of an exception catch event is unpredictable.

ArchMsg.Warning.warning_unpredictable_exception_return_inconsistent_state

TAGS **UNPREDICTABLE** DISPLAY state on exception return is unpredictable.

ArchMsg.Warning.warning_unpredictable_exception_return_instruction

TAGS **UNPREDICTABLE** DISPLAY exception return instruction is unpredictable.

ArchMsg.Warning.warning_unpredictable_in_debug_state

TAGS **UNPREDICTABLE** DISPLAY Instruction is **UNPREDICTABLE** when executed in debug state.

ArchMsg.Warning.warning_unpredictable_mair_encoding

TAGS **UNPREDICTABLE** DISPLAY Memory attribute %{ATTR} has invalid MAIR encoding; this may lead to unpredictable results. Fields:

ATTR unsigned int

Memory attribute.

ArchMsg.Warning.warning_unpredictable_pmu_counter_access

TAGS **UNPREDICTABLE** DISPLAY Access to PMU counters from non-secure EL0 or EL1 is **UNPREDICTABLE** with MDCR_EL2.HPMN set to 0.

ArchMsg.Warning.warning_unpredictable_prioritization_breakpoint_match_vector_catch

TAGS **UNPREDICTABLE** DISPLAY Two events with the same priority occurred at the same instruction: Address Matching Vector Catch debug event, and Breakpoint debug event. It is constrained unpredictable which is taken.

ArchMsg.Warning.warning_unpredictable_stack_selection

TAGS **UNPREDICTABLE** DISPLAY **UNPREDICTABLE** state on context switch.

ArchMsg.Warning.warning_unpredictable_tsize_out_of_range

TAGS **UNPREDICTABLE** DISPLAY %{TCR}.T%{TTBR:d}SZ (%{TSIZE:d} bits) is out of range. Must be between 25 and %{TSIZE_MAX:d}. Fields:

TCR enum

Which TCR.

TSIZE signed int

Value of TSize.

TSIZE_MAX signed int

The maximum allowed value.

TTBR unsigned int

Which TTBR region.

ArchMsg.Warning.warning_unpredictable_unaligned_pc_as_base_register

TAGS **UNPREDICTABLE** DISPLAY The PC value must be word-aligned, otherwise the behavior of the instruction is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_unaligned_pop_stack

TAGS **UNPREDICTABLE** DISPLAY **CONSTRAINED UNPREDICTABLE** unaligned stack frame %{FRAME_ADDR:x} in exception-return. Fields:

FRAME_ADDR unsigned int
unaligned stack frame address.

ArchMsg.Warning.warning_unpredictable_unaligned_to_non_writeback
TAGS **UNPREDICTABLE** DISPLAY Unaligned Access to Non-Writeback-Cacheable Memory.

ArchMsg.Warning.warning_unpredictable_vmsa_memattrib
TAGS **UNPREDICTABLE** DISPLAY Unpredictable vmsa memory attribute with texcb value programmed to %{TEXCB} when tex remap is %{TEX_REMAP}. Fields:

TEXCB unsigned int
texcb value.

TEX_REMAP bool
use tex remap.

ArchMsg.Warning.warning_unpredictable_vtcr_t0sz_sl0
TAGS **UNPREDICTABLE** DISPLAY The combination of VTCR.TOSZ=%{TOSZ:d} and VTCR.SLO=%{SLO:d} is unpredictable. Fields:

SLO signed int
VTCR.SLO, starting level for stage 2 translation table walks.

TOSZ signed int
VTCR.TOSZ, required input address range.

ArchMsg.Warning.warning_unpredictable_write_DHCSR
TAGS **UNPREDICTABLE** DISPLAY **UNPREDICTABLE** set of bit-changes in DHCSR. Fields:

NEW unsigned int
data attempted to be written.

OLD unsigned int
previous value.

ArchMsg.Warning.warning_unpredictable_write_SHCSR
TAGS **UNPREDICTABLE** DISPLAY **UNPREDICTABLE** setting a pending bit to 1 for an exception with priority >= execution priority.

ArchMsg.Warning.warning_unsupported_access_to_memory_mapped_register
TAGS **UNPREDICTABLE** DISPLAY Access to memory mapped register is unsupported.

ArchMsg.Warning.warning_user_jmcr_access
TAGS **UNPREDICTABLE** DISPLAY User %{WRITE:(write|read)} access to JMCR is unpredictable. Fields:

WRITE unsigned int
access is write.

ArchMsg.Warning.warning_wcr_mask_and_bas
TAGS **UNPREDICTABLE** DISPLAY DBGWCR_EL1.MASK is non-zero and BAS is not 0xff.

ArchMsg.Warning.warning_wcr_mask_reserved

TAGS **UNPREDICTABLE** DISPLAY DBGWCRn_EL1.MASK is set to a reserved value.

ArchMsg.Warning.warning_wcr_non_configuous_bas

TAGS **UNPREDICTABLE** DISPLAY DBGWCRn_EL1.BAS has non-contiguous set of ones.

ArchMsg.Warning.warning_wvr_masked_bit_not_zero

TAGS **UNPREDICTABLE** DISPLAY DBGWVRn_EL1.VA contains a bit masked by DBGWCRn_EL1.MASK that is not zero.

ArchMsg.Why.why_branch_target_exception

A branch target exception has occurred. DISPLAY branch target exception: target pc=%{PC}, target opcode=%{OPCODE}. PRIMARY KEY PC. Fields:

BTYPE unsigned int

pstate.BTYPE at time branch taken.

OPCODE unsigned int

opcode at branch target.

PC unsigned int

Virtual Address of branch target.

ArchMsg.Why.why_illegal_state

An illegal mode change has occurred. DISPLAY illegal mode change: %{MESSAGE}. Fields:

MESSAGE string

The message.

BRANCH_MISPREDICT

Simulating branch mispredict. Fields:

PC unsigned int

Origin address (or 0 if unavailable).

BRA_DIR

Direct branches, to immediate address. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISSET enum

The instruction set of the branch instruction.

IS_COND bool

Indicates if this is a conditional branch.

IS_HINTED bool

Whether the conditional instruction is hinted.

IS_ISB bool

Whether the instruction is ISB.

IS_LINK bool

Indicates if this is a branch with link.

PC unsigned int

The address of the branch instruction.

TARGET_ISET enum

The instruction set after the branch.

TARGET_PC unsigned int

The address the instruction branches to.

BRA_INDIR

Indirect branches, perhaps to a register. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISET enum

The instruction set of the branch instruction.

IS_COND bool

Indicates if this is a conditional branch.

IS_HINTED bool

Whether the conditional instruction is hinted.

IS_ISB bool

Whether the instruction is ISB.

IS_LINK bool

Indicates if this is a branch with link.

PC unsigned int

The address of the branch instruction.

TARGET_ISET enum

The instruction set after the branch.

TARGET_PC unsigned int

The address the instruction branches to.

BRA_INDIR_CCFAIL

Indirect branches fail, perhaps to a register.

BRA_RET_CCFAIL

Branch return fail, perhaps to a register.

CACHE_MAINTENANCE_OP

Cache Maintenance Operation. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

DATA unsigned int

Specified MVA or Set/way.

FUNCTION enum

Clean / Invalidate.

SCOPE enum

Affected region.

SIDE enum

Inst / Data.

CCFAIL

Conditional instruction condition check fail. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CCFAIL_UNC

Unconditional instruction condition check fail. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CCPASS

Conditional instruction condition check pass. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CCPASS_UNC

Unconditional instruction condition check pass. Fields:

COND **enum**

The condition of the conditional instruction.

CORE_NUM **unsigned int**

Core number in a multi processor.

PC **unsigned int**

The address of the conditional instruction.

CODE_CACHE_MAINT

Code cache maintenance. Fields:

END_INCL **unsigned int**

Inclusive end address for VA or PA requests.

NS **bool**

Non-secure world for PA.

START **unsigned int**

Start address for VA or PA requests.

TYPE **enum**

Request type.

COMPILE_BLOCK_END

Last instruction of basic block translated. Fields:

VADDR **unsigned int**

Address of next instruction after this basic block.

COMPILE_BLOCK_START

First instruction of basic block translated. Fields:

VADDR **unsigned int**

Address of first instruction in basic block.

COMPILE_INST

ARM instruction compiled. Fields:

DISASS **string**

Disassembly of the instruction.

ISSET **enum**

The instruction set of this instruction.

ITSTATE **unsigned int**

The ITSTATE current for the instruction.

OPCODE **unsigned int**

The opcode of the instruction.

PC unsigned int

The address of the instruction.

SIZE unsigned int

The size of the instruction in bytes.

CONTEXTIDR

Context ID Register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

NS bool

Secure or nonsecure banked register is accessed.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The new value written.

CONTEXT_SYNC

Called for every context synchronization event. Fields:

ARCH_CSE bool

Architectural Context Synchronization Event.

PC unsigned int

The address of the synchronization instruction.

SYSREG_SYNC bool

System register synchronization event.

CORE_ENDIAN

Core BE8 Big-Endian state changed. Fields:

BE8 bool

Core BE8 Big-Endian state.

CORE_INFO

Static processor attributes. Only triggered by a call to DumpState(). Fields:

ARCH_PROFILE enum

The architecture profile of the core.

CLUSTER_ID unsigned int

The cluster ID of this processor.

CORE_NUM unsigned int

The number of this core in an MP processor.

DETAILED_CORE_FEATURES enum

Detailed core features as a bitfield. Enum values are bitmasks to extract the info from the field.

FPU_VERSION enum

The VFP version implemented by the core.

MEM_ARCH enum

The memory architecture of the core.

MIDR_EL1 unsigned int

MIDR_EL1 value.

NUM_CORES unsigned int

The number of cores in this MP processor.

QUANTUM_SIZE unsigned int

The default quantum size of the core (in ticks).

SECURITY_FEATURES bool

Does the core have security features?.

CORE_LOADS

Processor load accesses. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ACQREL enum

Is this an acquire/release.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DATA unsigned int

The data read or written.

ELEMENT_SIZE unsigned int

Width of each element.

ENDIANNESS enum

Byte ordering of this access.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

LOCK enum

Normal, exclusive or locked access.

MECID unsigned int

Memory Encryption Context ID, if applicable.

MECID2 unsigned int

Memory Encryption Context ID of the second page, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PADDR_TAG unsigned int

If tag checked, the physical address tag for the access.

PAGE_SIZE unsigned int

Page size as $\log_2(\text{size in bytes})$.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK, 2=Exclusive Failed, 3=Unsupported.

SIZE unsigned int

Width of the access in bytes. Only required if DATA is not traced.

TAG_CHECKED bool

Is this access tag checked.

TAG_CHECKED2 bool

Is the second page access tag checked.

TRANS bool

Is this a translated access.

VADDR unsigned int

The virtual address of the access.

CORE_REGS

Changes of the core registers R0 to R14. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number, 0 to 14.

MODE enum

Bank of the register accessed.

OLD_VALUE unsigned int

The old value overwritten.

PHYS_ID enum

The physical register accessed.

VALUE unsigned int

The new value written to the register.

CORE_REGS64

Changes of the core registers X0..X30, SP_ELn. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ID enum

The register number, 0-30 for X0-X30, >=32 for SP_ELn.

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

CORE_REGS64_READ

General purpose AArch64 register has been read X0..X30, SP_ELn. Fields:

ID enum

The register number, 0-30 for X0-X30, >=32 for SP_ELn.

VALUE unsigned int

The value read from the register.

CORE_STALLED_TRANSITION

Detect when a core is prevented from progressing PChannel states. Fields:

CURRENT_STATE enum

Current power mode of core.

REASON string

The reason a transition was stalled, and what must be done to unblock it.

TARGET_STATE enum

Desired power mode of core after transition.

CORE_STORES

Processor store accesses. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ACQREL enum

Is this an acquire/release.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DATA unsigned int

The data read or written.

ELEMENT_SIZE unsigned int

Width of each element.

ENDIANNESS enum

Byte ordering of this access.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

LOCK enum

Normal, exclusive or locked access.

MECID unsigned int

Memory Encryption Context ID, if applicable.

MECID2 unsigned int

Memory Encryption Context ID of the second page, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PADDR_TAG unsigned int

If tag checked, the physical address tag for the access.

PAGE_SIZE unsigned int

Page size as $\log_2(\text{size in bytes})$.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK, 2=Exclusive Failed, 3=Unsupported.

SIZE unsigned int

Width of the access in bytes. Only required if DATA is not traced.

TAG_CHECKED bool

Is this access tag checked.

TAG_CHECKED2 bool

Is the second page access tag checked.

TRANS bool

Is this a translated access.

VADDR unsigned int

The virtual address of the access.

CP14_READ

System Coprocessor register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP14 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc1 unsigned int

opcode 1.

opc2 unsigned int

opcode 2.

CP14_WRITE

System Coprocessor register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP14 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc1 unsigned int

opcode 1.

opc2 unsigned int

opcode 2.

CP15_READ

System Coprocessor register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP15 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc1 unsigned int

opcode 1.

opc2 unsigned int

opcode 2.

CP15_READ64

System Coprocessor register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP15 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc unsigned int

opcode 1.

CP15_WRITE

System Control Coprocessor register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP15 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc1 unsigned int

opcode 1.

opc2 unsigned int
opcode 2.

CP15_WRITE64

System Coprocessor register write. Fields:

CORE_NUM unsigned int
Core number in a multi processor.

CRm unsigned int
CRm.

NS bool
Secure or nonsecure banked register is accessed.

REG_NAME string
Name of the CP15 register accessed.

SECURITY_STATE enum
The current security state.

UNDEF bool
The register accessed is undefined.

UPDATED_VALUE unsigned int
Updated value of the register now it has been written.

VALUE unsigned int
The new value written.

opc unsigned int
opcode 1.

CPSR

CPSR change. (In AArch64 PSTATE represented in SPSR format). Fields:

CORE_NUM unsigned int
Core number in a multi processor.

OLD_VALUE unsigned int
The old CPSR value.

UNKNOWN unsigned int
Bits within the register that have unknown value.

VALUE unsigned int
The new CPSR value.

CRYPTO_SPEC

Every crypto instruction speculatively executed.

CorePowerStateChange

Core power-up state has changed. DISPLAY Core powered %{PU:(up|down)}. Fields:

PU bool

Whether the core powered up.

DATA_CACHE_SET_ALLOC_TAGS

Set allocation tags from DC GVA instruction. Fields:

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

NSDESC unsigned int

The security state of the access.

PADDR unsigned int

The physical address of the access.

PAS enum

The physical address space of the page.

RESPONSE enum

0=Aborted, 1=OK.

SIZE unsigned int

The size of memory in bytes.

TAG unsigned int

The value of the tag being set.

TAGGING_TYPE enum

Type of Memory tagging (Physical or Virtual).

TAG_VADDR unsigned int

The virtual address of the Virtual Tag if Virtual tagging is used; otherwise 0.

VADDR unsigned int

Virtual address used for setting allocation tags.

DATA_CACHE_ZERO

Zero and invalidate cache line from DC ZVA instruction. Fields:

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

NSDESC unsigned int

The security state of the access.

PADDR unsigned int

The physical address of the access.

PADDR_TAG unsigned int

If tag checked, the physical address tag for the access.

PAS enum

The physical address space of the page.

RESPONSE enum

0=Aborted, 1=OK.

SIZE unsigned int

The size of zero'd memory in bytes.

TAG_CHECKED bool

Is this access tag checked.

VADDR unsigned int

Virtual address of the zero'd cache line.

DEBUG_EVENT

Hardware debug support event. Fields:

EVENT enum

Description of event.

VALUE unsigned int

data value.

DEBUG_TARGET_ISET

Hardware debug target ISA. Fields:

TARGET_ISET enum

The target instruction set of the debug state.

DMI_ALLOCATE

DMI allocated. Fields:

ACCESS_DIRECTION enum

READ / WRITE.

PADDR unsigned int

The physical address of the DMI.

PTR unsigned int

DMI range pointer.

SIDE enum

DSIDE / ISIDE.

SIZE unsigned int

The size of DMI.

VADDR unsigned int

The virtual address of the DMI..

DMI_HIT

DMI hit, access used DMI (only fire for small DMI regions i.e. less than a full page). Fields:

ACCESS_DIRECTION enum

READ / WRITE.

PADDR unsigned int

The physical address of the DMI.

SIDE enum

DSIDE / ISIDE.

SIZE unsigned int

The size of the DMI.

VADDR unsigned int

The virtual address of the DMI..

DMI_REVOKE

DMI revoked. Fields:

ACCESS_DIRECTION enum

READ / WRITE.

PADDR unsigned int

The physical address of the DMI.

SIDE enum

DSIDE / ISIDE.

SIZE unsigned int

The size of DMI.

VADDR unsigned int

The virtual address of the DMI..

END_COMPILE

Compilation end. Fields:

END_OF_PAGE_COUNT unsigned int

Number of basic blocks exited due to page boundary since START_COMPILE.

FETCHFAIL_COUNT unsigned int

Number of basic blocks exited due to ifetch failure START_COMPILE.

INST_COUNT unsigned int

Number of instructions compiled since START_COMPILE.

NONSEQ_COUNT unsigned int

Number of basic blocks exited due to non-sequential instructions since START_COMPILE.

PAGE_STRADDLE_COUNT unsigned int

Number of basic blocks exited due to unaligned instructions crossing page since START_COMPILE.

ESB

Error Synchronization Barrier hit. Fields:

ASYNC bool

ESB is async.

IMPLICIT bool

ESB is implicit.

EXCEPTION

Exception taken. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ESR unsigned int

The value assigned to ESR when exception is taken.

ESR_EC unsigned int

The Value assigned to ESR.EC when exception is taken.

IS_PHYSICAL bool

Physical or Virtual exception.

IS_TRAP bool

Exception coming from a trap.

LR unsigned int

The value assigned to the link register.

PC unsigned int

The location where the exception occurred.

PREFERRED_RETURN unsigned int

The preferred return address for the exception.

TAKEN_LOCALLY bool

Whether exception is taken locally(same exception level).

TARGET_ISET enum

The instruction set of the exception handler code.

TARGET_PC unsigned int

The address the exception branches to.

VECTOR enum

The exception vector.

VECTOR_OFFSET unsigned int

The offset of an exception vector from the base address.

EXCEPTION_END

Every exception completed.

EXCEPTION_INFO

Trace human-readable reason of the current exception for data/instruction aborts, floating point exceptions and unknown exceptions. Fields:

DESCRIPTION string

Reason for current exception.

ESR_EC unsigned int

The value of ESR.EC when exception is taken.

EXCEPTION_RAISE

Every exception raised.

EXCEPTION_RETURN

Branches on leaving exception. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISSET enum

The instruction set of the branch instruction.

IS_COND bool

Indicates if this is a conditional branch.

IS_HINTED bool

Whether the conditional instruction is hinted.

IS_ISB bool

Whether the instruction is ISB.

IS_LINK bool

Indicates if this is a branch with link.

PC unsigned int

The address of the branch instruction.

TARGET_ISET enum

The instruction set after the branch.

TARGET_PC unsigned int

The address the instruction branches to.

EXCEPTION_RETURN_PREBRANCH

Exception return event that must occur before PC and processor state have updated.

EXCEPTION_START

Every exception started.

FIQ_TAKEN

FIQ taken exception.

FP_EXCEPTIONS_SIGNALLED

Trace which FP exceptions were signalled as a part of the FP operations performed by an instruction. Fields:

FP_EXCEPTIONS_SIGNALLED_MASK unsigned int

Indicates which FP exceptions were signalled by an instruction. This uses the same encoding as the cumulative floating-point exception bitfields in FPSR.

FP_STATE

Floating point state. Fields:

FPSR unsigned int

FPSR register value.

MASK unsigned int

The mask indicating updated cumulative bits.

FREQ_CHANGED

Core frequency changed. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The instruction count of this core.

NEW_FREQ signed int

The new frequency of this core (expressed in Hz).

OLD_FREQ signed int

The old frequency of this core (expressed in Hz).

HLT

HLT instruction occurred. Fields:

IMM16 unsigned int

The 16-bit immediate value encoded in HLT instruction.

INST

Every instruction executed. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DEBUG_STATE bool

The instruction is executed in debug state.

DISASS string

Disassembly of instruction.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISET enum

The current instruction set.

ITSTATE unsigned int

The current ITSTATE.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

MODE enum

The mode the core is in.

NS unsigned int

The core's non-secure bit.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

OPCODE unsigned int

The opcode of the instruction.

PADDR unsigned int

The physical address of the instruction.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the instruction.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the page.

PC unsigned int

The address of the instruction.

SECURITY_STATE enum

The core's security state.

SIZE unsigned int

The size of the instruction in bytes.

INST_END

Every instruction completed.

INST_START

Every instruction started. Fields:

DEBUG_STATE *bool*

The instruction is executed in debug state.

ISET *enum*

The current instruction set.

MODE *enum*

The mode the core is in.

NS *enum*

The current Secure State.

PC *unsigned int*

The address of the conditional instruction.

SECURITY_STATE *enum*

The current Security State.

INST_STRADDLE

Instruction straddles boundary. Fields:

DEBUG_STATE *bool*

The instruction is executed in debug state.

ISET *enum*

The current instruction set.

MODE *enum*

The mode the core is in.

NS *enum*

The current Secure State.

PC *unsigned int*

The address of the conditional instruction.

SECURITY_STATE *enum*

The current Security State.

IRQ_TAKEN

IRQ taken exception.

ISB

ISB instruction occurred. Fields:

PC *unsigned int*

The address of the ISB instruction.

LOAD_MULTIPLE_REGS_LIST

List of destination registers (not source) of a multiple registers load instruction. Fields:

REG string

Register to add into the list.

LOCAL_MONITOR

Local monitor activity. Fields:

PADDR unsigned int

Local Monitor Address.

State enum

State of the monitor (Open/Exclusive).

MAPPING_CREATE

A memory mapping has been created. This is an implementation detail of the model and so may change without notice. Fields:

ASID unsigned int

The ASID used during translation, or 0 if no ASID was used.

IS_VALID bool

Whether the mapping is valid or not.

PADDR unsigned int

Base physical address of region, or 0 if not known.

SIDE enum

I-side or D-side.

SIZE unsigned int

Size of the page, or 0 if not known.

VADDR unsigned int

Base virtual address of region.

VMID unsigned int

The VMID used during translation, or 0 if no VMID was used.

MAPPING_DESTROY

A memory mapping has been destroyed. This is an implementation detail of the model and so may change without notice. Fields:

ASID unsigned int

The ASID used during translation.

CAUSE enum

The cause of invalidation.

PADDR unsigned int

Base physical address of page.

SIDE enum

I-side or D-side.

SIZE unsigned int

Size of the page.

VADDR unsigned int

Base virtual address of page.

VMID unsigned int

The VMID used during translation.

MEMMAP_DEBUG_READ

Memory mapped reads to the debug registers. Fields:

ADDR unsigned int

address.

CORE_NUM unsigned int

Core number in a multi processor.

EXT bool

Whether access is from an external device (such as the DAP).

NS enum

Secure state of the access.

REG_NAME string

Name of the debug register accessed.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

MEMMAP_DEBUG_WRITE

Memory mapped writes to the debug registers. Fields:

ADDR unsigned int

address.

CORE_NUM unsigned int

Core number in a multi processor.

EXT bool

Whether access is from an external device (such as the DAP).

NS enum

Secure state of the access.

REG_NAME string

Name of the debug register accessed.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The new value written.

MEMTAG_LOADS

Allocation tag memory read; when attributes and system configuration allow tag accesses.

Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

NUM_GRANULES unsigned int

Number of tags to be Loaded/stored.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK.

TAGGING_TYPE enum

Type of Memory tagging (Physical or Virtual).

TAGS unsigned int

The tags read or written. Tags for subsequent granules would be found at subsequent bytes in this field (One byte for each tag).

TAG_VADDR unsigned int

The virtual address of the Virtual Tag if Virtual tagging is used; otherwise 0.

VADDR unsigned int

The virtual address of the access.

MEMTAG_LOAD_INST

Allocation tag load instruction; regardless of tags being actually accessed. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

NUM_GRANULES unsigned int

Number of tags to be Loaded/stored.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK.

TAGGING_TYPE enum

Type of Memory tagging (Physical or Virtual).

TAGS unsigned int

The tags read or written. Tags for subsequent granules would be found at subsequent bytes in this field (One byte for each tag).

TAG_VADDR unsigned int

The virtual address of the Virtual Tag if Virtual tagging is used; otherwise 0.

VADDR unsigned int

The virtual address of the access.

MENTAG_STORES

Allocation tag memory write; when attributes and system configuration allow tag accesses.

Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

NUM_GRANULES unsigned int

Number of tags to be Loaded/stored.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK.

TAGGING_TYPE enum

Type of Memory tagging (Physical or Virtual).

TAGS unsigned int

The tags read or written. Tags for subsequent granules would be found at subsequent bytes in this field (One byte for each tag).

TAG_VADDR unsigned int

The virtual address of the Virtual Tag if Virtual tagging is used; otherwise 0.

VADDR unsigned int

The virtual address of the access.

MENTAG_STORE_INST

Allocation tag store instruction; regardless of tags being actually accessed. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

NUM_GRANULES unsigned int

Number of tags to be Loaded/stored.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK.

TAGGING_TYPE enum

Type of Memory tagging (Physical or Virtual).

TAGS unsigned int

The tags read or written. Tags for subsequent granules would be found at subsequent bytes in this field (One byte for each tag).

TAG_VADDR unsigned int

The virtual address of the Virtual Tag if Virtual tagging is used; otherwise 0.

VADDR unsigned int

The virtual address of the access.

MMU_D128_TTB_READ

This event is triggered by reads caused by a D128 translation table walk. Fields:

ABORTED bool

Was the walk successful.

DATA_HIGH unsigned int

The data upper bits read.

DATA_LOW unsigned int

The data lower bits read.

IPA unsigned int

For stage 1, the IPA of the read.

LEVEL unsigned int

Translation table level.

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

OUTADDR unsigned int

The output physical address.

PADDR unsigned int

The physical address of the read.

PASmem enum

Physical Address Space of the accessed memory.

PASreq enum

Physical Address Space of the requested lookup.

REGIME_EL enum

Entry matches in this translation regime.

REQUESTER enum

The requester of this table walk.

SIDE enum

Inst / Data.

STAGE unsigned int

Translation stage.

MMU_D128_TTB_WRITE

This event is triggered by writes caused by a D128 translation table walk. Fields:

ABORTED bool

Was the walk successful.

DATA_HIGH unsigned int

The data upper bits written.

DATA_LOW unsigned int

The data lower bits written.

IPA unsigned int

For stage 1, the IPA of the write.

IS_HACDBS_ACCESS bool

If the access was due to HACDBS_CLEAN.

LEVEL unsigned int

Translation table level.

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

OUTADDR unsigned int

The output physical address.

PADDR unsigned int

The physical address of the write.

PASmem enum

Physical Address Space of the accessed memory.

PASreq enum

Physical Address Space of the requested lookup.

REGIME_EL enum

Entry matches in this translation regime.

SIDE enum

Inst / Data.

STAGE unsigned int

Translation stage.

MMU_TRANS

Address translation information. Fields:

ASID unsigned int

Address space identifier.

CORE_NUM unsigned int

Core number in a multi processor.

Hyp bool

Entry matches in Hyp state only.

INNERCACHE_RA bool

Is the inner cache allocate on read.

INNERCACHE_TRANSIENT bool

Is the inner write-through transient.

INNERCACHE_TYPE enum

Inner Caching scheme (NC/MB/WA).

INNERCACHE_WA bool

Is the inner cache allocate on write.

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MEMTYPE enum

Memory type.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC enum

Is secure side supposed to access secure or nonsecure memory.

OUTERCACHE_RA bool

Is the outer cache allocate on read.

OUTERCACHE_TRANSIENT bool

Is the outer write-through transient.

OUTERCACHE_TYPE enum

Outer Caching scheme (NC/MB/WA).

OUTERCACHE_WA bool

Is the outer cache allocate on write.

PADDR unsigned int

Physical address of the access.

PAGESIZE unsigned int

Page size as $\log_2(\text{size})$.

PAS enum

PAS supposed to be accessed is secure or nonsecure memory or Root/Realm if RME is implemented.

SH enum

Shareability.

SIDE enum

Inst / Data.

STAGE1_PERM unsigned int

Stage 1 permissions, [5:3] Privileged access (XWR) [2:0] User access (XWR).

STAGE2_PERM unsigned int

Stage 2 permission mask [5:3] Privileged access (XWR) [2:0] User access (XWR).

VADDR unsigned int

Virtual address of the access.

VMID unsigned int

Virtual machine identifier.

XS bool

XS Attribute.

nG enum

Flag indicating whether ASID will be matched.

MMU_TTB_READ

This event is triggered by reads caused by a translation table walk. Fields:

ABORTED bool

Was the walk successful.

DATA unsigned int

The data read.

DESC_KIND enum

The encoding scheme of the descriptors used for this walk.

IPA unsigned int

For stage 1, the IPA of the read.

LEVEL unsigned int

Translation table level.

LPAE bool

Is this for an LPAE translation (*DEPRECATED*, use DESC_KIND).

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSmem bool

Non secure memory access.

NSreq bool

Non secure request.

OUTADDR unsigned int

The output physical address.

PADDR unsigned int

The physical address of the read.

PASmem enum

Physical Address Space of the accessed memory.

PASreq enum

Physical Address Space of the requested lookup.

REGIME_EL enum

Entry matches in this translation regime.

REQUESTER enum

The requester of this table walk.

SIDE enum

Inst / Data.

STAGE unsigned int

Translation stage.

MMU_TTB_WRITE

This event is triggered by writes caused by a translation table walk. Fields:

ABORTED bool

Was the walk successful.

DATA unsigned int

The data written.

DESC_KIND enum

The encoding scheme of the descriptors used for this walk.

IPA unsigned int

For stage 1, the IPA of the write.

IS_HACDBS_ACCESS bool

If the access was due to HACDBS_CLEAN.

LEVEL unsigned int

Translation table level.

LPAE bool

Is this for an LPAE translation (*DEPRECATED*, use DESC_KIND).

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSmem bool

Non secure memory access.

NSreq bool

Non secure request.

OUTADDR unsigned int

The output physical address.

PADDR unsigned int

The physical address of the write.

PASmem enum

Physical Address Space of the accessed memory.

PASreq enum

Physical Address Space of the requested lookup.

REGIME_EL enum

Entry matches in this translation regime.

SIDE enum

Inst / Data.

STAGE unsigned int

Translation stage.

MODE_CHANGE

Mode change. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

MODE enum

The new mode.

NON_SECURE enum

The core's new non-secure bit.

OLD_MODE enum

The old mode.

SECURITY_STATE enum

The new security state.

NOT_TAG_CHECKED_INST

MTE Tag not checked for current instruction.

PERIODIC

Called for every quantum. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The instruction count of this CPU.

PC unsigned int

The address of the next instruction to be executed on this CPU.

PMU_COUNTER_OVERFLOW

PMU counter overflow. Fields:

COUNTER_GROUP enum

Counter group.

COUNTER_RANGE enum

Counter range.

EVENT_ID unsigned int

Event Identifier, field is valid only for Event counter group.

INDEX unsigned int

Counter index (as selected by PMSELR), field is valid only for Event/Cycle counter groups.

INTERRUPT bool

Is interrupt enabled on overflow for this counter.

POINTER_AUTH

Called when execution hits a pointer authentication check. Fields:

AUTH_STATE bool

Authentication result.

KEY_USED enum

Pointer authentication key.

POTINDEX_UPDATE

Value of POT index for current pc changed. Fields:

NEW_POTINDEX unsigned int

New POTIndex.

OLD_POTINDEX unsigned int

Old POTIndex.

PREFETCH_MEMORY64

Prefetch from PRFM or PRFUM instructions. Fields:

PRFOP enum

Prefetch hint.

VADDR unsigned int

Virtual address of the location that should be prefetched.

PRELOAD64_REQUEST

Preload requests from PRFM or PRFUM instructions. Fields:

PRFOP enum

Prefetch hint.

VADDR unsigned int

Virtual address of the location that should be prefetched.

PRELOAD_DATA

Data preload from PLD instruction. Fields:

VADDR unsigned int

Virtual address of the data that should be preloaded.

PRELOAD_INST

Instruction preload from PLI instruction. Fields:

VADDR unsigned int

Virtual address of the instruction that should be preloaded.

PRE_CORE_LOAD

Trace just before a core load. Fields:

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

PRE_CORE_STORE

Trace just before a core store. Fields:

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

PRE_TTB_READ

This event is triggered before reads caused by a translation table walk. Fields:

ADDR unsigned int

The physical address of the read.

PSTATE

PSTATE fields update. Fields:

ALLINT unsigned int

All interrupt mask bit.

BTYPE unsigned int

Branch target identification bit.

CORE_NUM unsigned int

Core number in a multi processor.

DAIF unsigned int

Exception mask bits.

DIT unsigned int

Data Independent Timing bit.

EL unsigned int

Current Exception level field.

EXLOCK unsigned int

Exception return state lock bit.

IL unsigned int

Illegal Execution state bit.

NZCV unsigned int

Condition flags.

PACM unsigned int

PACM bit.

PAN unsigned int

Privileged Access Never state bit.

PM unsigned int

PMU exception mask bit.

PPEND unsigned int

PMU exception pending bit.

SM unsigned int

StreamingMode status bit.

SP unsigned int

Stack pointer register selection bit.

SS unsigned int

Software Step bit.

SSBS unsigned int

Speculative Store Bypass Safe bit.

TCO unsigned int

Tag Check Override bit.

TOUCHED unsigned int

Mask of the fields that have been touched since this trace has last been emitted. Note that touched does not necessarily mean the value changed. The index of the field is the bit index in the mask plus the index of this field plus one.

UAO unsigned int

User Access Override bit.

UINJ unsigned int

Undefined Instruction exceptions bit.

ZA unsigned int

ZA storage and ZT0 register enable bit.

nRW unsigned int

Current Execution state bit.

PSTATE_PACTIVE_OPMODE_TRANSITION

Trace pactive and Opmode mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Operating mode after the transition.

OLD_STATE enum

Operating mode before the transition.

PSTATE_PACTIVE_OPMODE_TRANSITION_DENIED

Trace the denied pactive and Opmode mode transitions on PChannel. Fields:

DEVPACTIVE **unsigned int**

Integer representation of DEVPACTIVE[] signals.

NEW_STATE **enum**

Operating mode after the transition.

OLD_STATE **enum**

Operating mode before the transition.

REASON **string**

Reason for the transition being denied.

PSTATE_PACTIVE_POWER_TRANSITION

Trace pactive and power mode transitions on PChannel. Fields:

DEVPACTIVE **unsigned int**

Integer representation of DEVPACTIVE[] signals.

NEW_STATE **enum**

Power mode after the transition.

OLD_STATE **enum**

Power mode before the transition.

PSTATE_PACTIVE_POWER_TRANSITION_DENIED

Trace the denied pactive and power mode transitions on PChannel. Fields:

DEVPACTIVE **unsigned int**

Integer representation of DEVPACTIVE[] signals.

NEW_STATE **enum**

Power mode after the transition.

OLD_STATE **enum**

Power mode before the transition.

REASON **string**

Reason for the transition being denied.

PSTATE_PACTIVE_THREAD_OPMODE_TRANSITION

Trace pactive and Opmode mode transitions for threads on PChannel. Fields:

DEVPACTIVE **unsigned int**

Integer representation of DEVPACTIVE[] signals.

NEW_STATE **enum**

Operating mode after the transition.

OLD_STATE **enum**

Operating mode before the transition.

QUANTUM_START

Called at the start of every quantum.

RANGE_PRELOAD_REQUEST

Preload requests from RPRFM. Fields:

COUNT unsigned int

Number of blocks of data to access.

LENGTH unsigned int

Number of contiguous bytes to be accessed in each block.

OP unsigned int

Range prefetch operation .

REUSE unsigned int

Maximum number of bytes to be accessed before next RPRFM .

STRIDE signed int

Stride to advance after accessing LENGTH bytes.

VADDR unsigned int

Base virtual address of range prefetch request.

RUN_STATE

Run state transition. Fields:

INST_COUNT unsigned int

Ticks count at point of transition.

NEW enum

New run state.

OLD enum

Old run state.

SEMIHOSTING_CALL

Call of a semihost function occurred. Fields:

PC unsigned int

The program counter after the semihosting call.

SEMIHOSTING_PRECALL

About to call semihost. Fields:

EL2 bool

Use EL2 translation regime.

NS bool

Is Non-Secure.

REG_WIDTH unsigned int

The current register width in bytes.

SIGNAL

External signal state change. Fields:

SIGNAL *enum*

Signal that changed.

STATE *bool*

Signal asserted state.

SOFTWARE_STEP

Return the Debug Software Step state. Fields:

STATE *enum*

Software Step state.

SPSR

SPSR change. Fields:

CORE_NUM *unsigned int*

Core number in a multi processor.

MODE *enum*

Which of the banked SPSR registers is written.

OLD_VALUE *unsigned int*

The old SPSR value.

VALUE *unsigned int*

The new SPSR value.

START_COMPILE

Compilation started. Fields:

VADDR *unsigned int*

Instruction where compilation begins.

STG_MULTIPLE_MICRO_OPS_ADDRESS_RANGE

Address range for MTE stores with multiple uops. Fields:

ADDR_END *unsigned int*

Final address.

ADDR_START *unsigned int*

Starting address.

STORE_MULTIPLE_REGS_ADDRESS_RANGE

Address range for stores with multiple registers. Fields:

ADDR_END *unsigned int*

Final address.

ADDR_START *unsigned int*

Starting address.

SUBOPTIMAL_LDST_RETIRED

suboptimal load/store. Fields:

ACCESS_SIZE unsigned int

Log2 of the access width in bytes used for alignment checking (1: half-word, 2: word ...).

OPTIMAL_ALIGN_SELECTOR unsigned int

Selector to decide the optimal alignment size (1: access width, 2: 4-bytes, 3: 8-bytes, ..., 12: 4-kbytes).

VADDR unsigned int

The virtual address of the access.

SVE_INST_SPEC

SVE operations speculatively executed.

SVE_LD_RETIRED

SVE load instructions architecturally executed.

SVE_LOADS

SVE load accesses. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ACQREL enum

Is this an acquire/release.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DATA unsigned int

The data read or written.

ELEMENT_SIZE unsigned int

Width of each element.

ENDIANNESS enum

Byte ordering of this access.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

LOCK enum

Normal, exclusive or locked access.

MECID unsigned int

Memory Encryption Context ID, if applicable.

MECID2 unsigned int

Memory Encryption Context ID of the second page, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PADDR_TAG unsigned int

If tag checked, the physical address tag for the access.

PAGE_SIZE unsigned int

Page size as $\log_2(\text{size in bytes})$.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK, 2=Exclusive Failed, 3=Unsupported.

SIZE unsigned int

Width of the access in bytes. Only required if DATA is not traced.

TAG_CHECKED bool

Is this access tag checked.

TAG_CHECKED2 bool

Is the second page access tag checked.

TRANS bool

Is this a translated access.

VADDR unsigned int

The virtual address of the access.

SVE_MEM_PREFETCH

Scalable Vector Extension memory prefetch. Fields:

CACHE_LEVEL enum

The preferred cache level.

HINT enum

What the memory is expected to be used for.

SIZE unsigned int

The prefetch size in bytes.

TEMPORAL bool

Whether the data is temporal.

VADDR unsigned int

The virtual address of the prefetch.

SVE_REG_READ

Scalable Vector Extension register read. Fields:

ID enum

The register identifier.

SM bool

Whether the PE is in Streaming Mode.

VALUE unsigned int

The read data.

SVE_REG_UPDATE

Scalable Vector Extension register update. Fields:

ID enum

The register identifier.

SM bool

Whether the PE is in Streaming Mode.

VALUE unsigned int

The written data.

SVE_REG_WRITE

Scalable Vector Extension register write. Fields:

ID enum

The register identifier.

SM bool

Whether the PE is in Streaming Mode.

VALUE unsigned int

The written data.

SVE_STORES

SVE store accesses. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ACQREL enum

Is this an acquire/release.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DATA unsigned int

The data read or written.

ELEMENT_SIZE unsigned int

Width of each element.

ENDIANNESS enum

Byte ordering of this access.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

LOCK enum

Normal, exclusive or locked access.

MECID unsigned int

Memory Encryption Context ID, if applicable.

MECID2 unsigned int

Memory Encryption Context ID of the second page, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PADDR_TAG unsigned int

If tag checked, the physical address tag for the access.

PAGE_SIZE unsigned int

Page size as $\log_2(\text{size in bytes})$.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK, 2=Exclusive Failed, 3=Unsupported.

SIZE unsigned int

Width of the access in bytes. Only required if DATA is not traced.

TAG_CHECKED bool

Is this access tag checked.

TAG_CHECKED2 bool

Is the second page access tag checked.

TRANS bool

Is this a translated access.

VADDR unsigned int

The virtual address of the access.

SVE_ST_RETIRED

SVE store instructions architecturally executed.

SVE_UNALIGNED_LDST_RETIRED

SVE load or store instructions architecturally executed that would cause alignment fault if A bit is set.

SVE_Z_PRED_MEM_ACCESS_INFO

Information about SVE/SME predicated accesses to Z registers (non Gather/Scatter). Fields:

MEM_ELEM_SIZE unsigned int

Width of elements accessed in memory (in bytes).

RnW bool

Whether this is a Read not a Write.

VA unsigned int

Virtual base address computed for accesses.

SYNC

Called for every synchronization, that is every quantum break. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The instruction count of this CPU.

LOCAL_QUANTUM unsigned int

The local quantum of this CPU.

LOCAL_TIME unsigned int

The local time of this CPU.

SYNC_REASON enum

Reason for sync.

SYSCALL

System call instruction executed. Fields:

IMM unsigned int

Immediate value of the system call instruction.

TYPE enum

System call type.

VADDR unsigned int

Instruction that caused the system call.

SYSREG_READ128

System Coprocessor 128 bit register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

REGNUM enum

Internal register number.

REGNUM_PHYS enum

Internal register number physically accessed (e.g. in presense of VHE redirection).

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_READ64

System Coprocessor register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REGNUM enum

Internal register number.

REGNUM_OPERAND enum

Register number corresponding to the instruction operand.

REGNUM_PHYS enum

Internal register number physically accessed (e.g. in presense of VHE redirection).

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc unsigned int

opcode 1.

opc0 unsigned int
opcode 0.

opc2 unsigned int
opcode 2.

SYSREG_UPDATE128

Triggers when the system updates a register. Fields:

CRm unsigned int
CRm.

CRn unsigned int
CRn.

NS bool
Secure or nonsecure banked register is accessed.

REG enum
Register number.

UNKNOWN unsigned int
Bits of the register which became unknown.

VALUE unsigned int
Value written to the register.

opc unsigned int
opcode 1.

opc0 unsigned int
opcode 0.

opc2 unsigned int
opcode 2.

SYSREG_UPDATE32

Triggers when the system updates a register. Fields:

CRm unsigned int
CRm.

CRn unsigned int
CRn.

NS bool
Secure or nonsecure banked register is accessed.

REG enum
Register number.

UNKNOWN unsigned int
Bits of the register which became unknown.

VALUE unsigned int

Value written to the register.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_UPDATE64

Triggers when the system updates a register. Fields:

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG enum

Register number.

UNKNOWN unsigned int

Bits of the register which became unknown.

VALUE unsigned int

Value written to the register.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_WRITE128

System Coprocessor 128 bit register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

REGNUM enum

Internal register number.

REGNUM_PHYS enum

Internal register number physically accessed (e.g. in presense of VHE redirection).

UNDEF bool

The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_WRITE64

System Coprocessor register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REGNUM enum

Internal register number.

REGNUM_OPERAND enum

Register number corresponding to the instruction operand.

REGNUM_PHYS enum

Internal register number physically accessed (e.g. in presense of VHE redirection).

UNDEF bool

The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

TRBU_BUFFER_WRAP

DISPLAY Trace buffer current write pointer wrapped.

TRBU_TRIGGER_EVENT

DISPLAY Trace buffer Trigger Event.

TRBU_WRITE

DISPLAY PETU trace data has been written to the TRBU memory buffer at %{BUFFER_ADDRESS} with size %{SIZE} and data %{DATA}. Fields:

BUFFER_ADDRESS unsigned int

Virtual Address of start of buffer write.

DATA unsigned int

Data written to the buffer.

SIZE unsigned int

Size of sample written to buffer, in bytes.

UNALIGNED_LDST_RETIRED

Processor unaligned load/store. Fields:

ACCESS_SIZE unsigned int

Log2 of the access width in bytes used for alignment checking (1=>half-word, 2->word ...).

NUMBER_OF_BEATS unsigned int

Number of accesses (beats) in this burst (Total size in bytes is ACCESS_SIZE * NUMBER_OF_BEATS).

VADDR unsigned int

The virtual address of the access.

VFP_D_REGS

VFP/NEON D 64 bit register write. Fields:

ALIASING enum

Type of register aliasing used.

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a write to the corresponding bit in the register. For example, 0x1 would indicate a write to VALUE[0].

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

VFP_D_REGS_READ

VFP/NEON D 64 bit register read. Fields:

ID unsigned int

The register number.

VALUE unsigned int

The current value of the register.

VFP_Q_REGS

VFP/NEON Q 128 bit register write. Fields:

ALIASING enum

Type of register aliasing used.

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a write to the corresponding byte in the register. For example, 0x1 would indicate a write to VALUE[7:0].

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

VFP_Q_REGS_READ

VFP/NEON Q 128 bit register read. Fields:

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a read to the corresponding byte in the register. For example, 0x1 would indicate a read to VALUE[7:0].

VALUE unsigned int

The current value of the register.

VFP_SYS_REGS

Writes to the VFP/NEON units system registers. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ID enum

Which VFP system register is written.

MASK unsigned int

The mask indicating updated cumulative bits.

OLD_VALUE unsigned int

The register's old value overwritten.

VALUE unsigned int

The new value written to the VFP system register.

VFP_S_REGS

VFP/NEON S 32 bit register write. Fields:

ALIASING enum

Type of register aliasing used.

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a write to the corresponding bit in the register. For example, 0x1 would indicate a write to VALUE[0].

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

VFP_S_REGS_READ

VFP/NEON S 32 bit register read. Fields:

ID unsigned int

The register number.

VALUE unsigned int

The current value of the register.

VIRTUAL_TAG_ALLOCATED

Virtual tag information when allocated. Fields:

TAG_VADDR **unsigned int**

Virtual tag virtual address.

VADDR **unsigned int**

Data Virtual address.

VIRTUAL_TAG **unsigned int**

Virtual tag value.

WAYPOINT

Signals end of basic block execution. Fields:

CORE_NUM **unsigned int**

Core number in a multi processor.

ISET **enum**

Origin instruction set.

IS_COND **bool**

Indicates if this is a conditional waypoint.

PC **unsigned int**

Origin address (or 0 if unavailable).

TAKEN **bool**

Indicates if this waypoint was taken.

TARGET **unsigned int**

Destination address.

TARGET_ISET **enum**

Destination instruction set.

WFE_END

WFE ended. Fields:

INST_COUNT **unsigned int**

Ticks count when leaving WFE.

WFE_EVENT_REGISTER

WFE event register status: set/clear, reason. Fields:

INST_COUNT **unsigned int**

Ticks count.

REASON **enum**

Reason for set/clear. Only REASON==0 (Cleared by WFE) clears the bit, all other reasons set it.

WFE_IGNORED

WFE ignored. Fields:

EVENT bool

This WFE was ignored because the event register was set.

INST_COUNT unsigned int

Ticks count when ignoring WFE.

TRAPPED bool

This WFE was trapped.

WFE_START

WFE entered. Fields:

INST_COUNT unsigned int

Ticks count when entering WFE.

WFI_END

WFI ended. Fields:

INST_COUNT unsigned int

Ticks count when leaving WFI.

WFI_IGNORED

WFI ignored. Fields:

DISABLED bool

This WFI was ignored because WFI is disabled.

INST_COUNT unsigned int

Ticks count when ignoring WFI.

REASON enum

specifies reason why WFI trace was ignored.

TRAPPED bool

This WFI was trapped.

WFI_START

WFI entered. Fields:

INST_COUNT unsigned int

Ticks count when entering WFI.

WFI_WAKEUP

WFI wakeup. Fields:

INST_COUNT unsigned int

Ticks count when WFI wakeup occurred.

REASON enum

Reason for wakeup.

2.29 ARM_Cortex-A725

This section describes the trace sources.

AA64_ASE_SVE_REGS

Modification of AArch64 Advanced SIMD or SVE Vector Registers (via SIMD or Scalar operations), SVE Predicate Registers and ZT0. Fields:

ID **enum**

The register identifier.

MASK **unsigned int**

Bitmask of the register to signal the modified bits in the VALUE field.

SM **bool**

Whether the PE is in Streaming Mode.

VALUE **unsigned int**

Value of AArch64 vector register.

ASYNC_MEMORY_FAULT

Context ID Register write. Fields:

FAULT **unsigned int**

Fault status present in ESR.ISS encoding for Serror in ARCH64 and present in DFSR encoding for external abort format in ARCH32.

PADDR **unsigned int**

Physical Address (or 0 if unavailable).

VADDR **signed int**

Virtual Address (or 0 if unavailable).

ATOMIC_END_ACCESS

Bus trace access for atomics. Fields:

ACCESS_FAIL **bool**

Memory access failed.

ADDR **unsigned int**

The virtual address of the access.

ATTR **unsigned int**

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

COMPARE_VALUE **unsigned int**

Compare value for CAS.

LOAD_VALUE unsigned int

Loaded values.

NSDESC unsigned int

The security state of the access.

OPERAND_VALUE unsigned int

Operation's operand.

OPERATION enum

Operation type.

PADDR unsigned int

The physical address of the access.

PRIV bool

Is this a privileged access?.

ATOMIC_START_ACCESS

Bus trace access for atomics. Fields:

ADDR unsigned int

The virtual address of the access.

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

COMPARE_VALUE unsigned int

Compare value for CAS.

NSDESC unsigned int

The security state of the access.

OPERAND_VALUE unsigned int

Operation's operand.

OPERATION enum

Operation type.

PADDR unsigned int

The physical address of the access.

PRIV bool

Is this a privileged access?.

ArchMsg.Error.exit_code

DISPLAY exit code: %{EXITCODE} component kind: %{KIND} reason: %{REASON}. Fields:

EXITCODE signed int

Exit code to be returned.

KIND string

Component kind that invoked the exit code trace.

REASON string

Reason why the exit code trace was invoked (optional).

ArchMsg.Warning.IgnoredWarmResetRequest

Event fires when software attempts to request warm reset through write to deprecated DBGPRCR.CWRR.

ArchMsg.Warning.RAS_ErrselrUnpredictableState

DISPLAY ERRSELR is in unpredictable state due to value greater than number_of_error_records written to it.

ArchMsg.Warning.RAS_UnpredictableStateWrite

DISPLAY %{REG_NAME} write ignored due to unpredictable ERRSELR state. Fields:

REG_NAME string

Name of register being written.

ArchMsg.Warning.ReportedRevisionRegisterRead

DISPLAY Reported revision and patch do not match that of the cpu. Fields:

ORIGINAL_VALUE unsigned int

The value the register would have based on the modelled revision.

REGISTER string

The name of the register that has ben read.

REPORTED_VALUE unsigned int

The value the register has been overridden to.

ArchMsg.Warning.Unpredictably Indexed PM Event Register

TAGS UNPREDICTABLE DISPLAY Event %{IsDirect:(Selector value|Register)} %{SEL} is >= %{N}. Fields:

IsDirect unsigned int

Direct Access rather than indirect.

N unsigned int

Number of Accessible Registers.

SEL unsigned int

Selector.

ArchMsg.Warning.cache_contents_unknown

A cache has been activated without being invalidated and may contain **UNKNOWN** content. DISPLAY %{SIDE:(D|I)}-Cache was enabled in %{REGIME:(SECURE|NON_SECURE|HYP)} regime but was not invalidated since power-on: cache lines could contain **UNKNOWN** content %{SIDE:(Note: D-side caches must invalidate SECURE lines even if the cache is only used by NON_SECURE code in order to ensure that unexpected dirty lines tagged SECURE are not naturally evicted)}. Fields:

REGIME unsigned int

Current translation regime.

SIDE unsigned int

0 d-side cache or unified cache, 1 i-side.

ArchMsg.Warning.dcimvac_matches_watchpoint

TAGS **UNPREDICTABLE** DISPLAY Watchpoints matching an AArch32 DCIMVA are implementation defined.

ArchMsg.Warning.decode_fieldmismatch

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with different fields %{FIELD1} and %{FIELD2}. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_fracbitsnegative

TAGS **UNPREDICTABLE** DISPLAY VCVT Instruction is unpredictable with a negative number of fraction bits.

ArchMsg.Warning.decode_initblock

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable in an IT block.

ArchMsg.Warning.decode_initblocknotlast

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable in an IT block when not the last.

ArchMsg.Warning.decode_invalid_condition

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable when the condition is not AL.

ArchMsg.Warning.decode_invalidfieldcombination

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with the values in %{FIELD1} and %{FIELD2}. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_invalidvalue

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with the value in %{FIELD}. Fields:

FIELD enum

Field name.

ArchMsg.Warning.decode_mem_hint_unallocated

TAGS **UNPREDICTABLE** DISPLAY There is no memory hint allocated to this bit pattern.

ArchMsg.Warning.decode_registermatch

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with %{FIELD1} and %{FIELD2} as the same register. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_registermismatch

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with %{FIELD1} and %{FIELD2} as different registers. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_registeroutofrange

TAGS **UNPREDICTABLE** DISPLAY Floating-point instruction is unpredictable indexing beyond the end of the register bank.

ArchMsg.Warning.decode_sbzsbo

TAGS **UNPREDICTABLE** DISPLAY Reserved bits in the instruction are not canonical.

ArchMsg.Warning.decode_transactiontoobig

TAGS **UNPREDICTABLE** DISPLAY Attempt to perform a memory transaction of more than 128 bytes.

ArchMsg.Warning.decode_unpred_other

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable.

ArchMsg.Warning.decode_unpreduseofpc

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable when %{FIELD} is PC. Fields:

FIELD enum

Field name.

ArchMsg.Warning.decode_unpreduseofr13

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable when %{FIELD} is R13. Fields:

FIELD enum

Field name.

ArchMsg.Warning.decode_unpreduseofr13inreglist

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with R13 in register-list.

ArchMsg.Warning.decode_useofd16orhigher

TAGS **UNPREDICTABLE** DISPLAY Attempt to use register D16 or higher where it is not allowed.

ArchMsg.Warning.decode_writebackandbaseinlist

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with write back when the base register is in the transfer list.

ArchMsg.Warning.decode_zeroregistersinlist

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable when no registers are to be transferred.

ArchMsg.Warning.misaligned_ptrs

DISPLAY TRBPTR_EL1(%{TRBPTR_EL1}) or TRBTRG_EL1(%{TRBTRG_EL1}) are not aligned as per TRBIDR_EL1.Align(%{TRBIDR_EL1}). Fields:

TRBIDR_EL1 unsigned int

TRBIDR_EL1.Align value.

TRBPTR_EL1 unsigned int

Current write pointer.

TRBTRG_EL1 unsigned int

Trigger pointer.

ArchMsg.Warning.out_of_range_ptrs

DISPLAY TRBPTR_EL1(%{TRBPTR_EL1}) is out of range as per TRBBASER_EL1(%{TRBBASER_EL1}) or TRBLIMITR_EL1.LIMIT(%{TRBLIMITR_EL1}). Fields:

TRBBASER_EL1 unsigned int

Base pointer.

TRBLIMITR_EL1 unsigned int

Limit pointer.

TRBPTR_EL1 unsigned int

Current write pointer.

ArchMsg.Warning.recursive_branch

An instruction is performing a branch that targets the same instruction. If this was intended then a WFI might be more effective DISPLAY recursive branch detected at PC=%{PC}.

PRIMARY KEY PC. Fields:

PC unsigned int

Address of instruction causing the branch.

ArchMsg.Warning.recursive_exception

An instruction has generated an exception that targets the same instruction. DISPLAY recursive %{TYPE} exception detected at PC=%{PC} (CPSR=%{CPSR}). PRIMARY KEY PC. Fields:

CPSR unsigned int

Processor state.

PC unsigned int

Address of instruction causing the exception.

TYPE enum

Type of exception.

ArchMsg.Warning.reentrant_vector_catch

TAGS **UNPREDICTABLE** DISPLAY Vector catch debug event on Prefetch or Data abort vector %{ADDR}. Fields:

ADDR unsigned int

vector address.

ArchMsg.Warning.reserved_it_state

TAGS **UNPREDICTABLE** DISPLAY Execution with reserved IT state %{ITSTATE}. Fields:

ITSTATE unsigned int

reserved IT state.

ArchMsg.Warning.tlb_contents_unknown

Translation has been enabled without invalidating the TLB. DISPLAY Translation was enabled in the %{PAS:(Secure|Non-secure|Root|Realm)} EL%{EL:d} translation regime (vmid=%{VMID:x}) without invalidating %{INVALIDITY:(|d-side |i-side |)}entries since power-on: %{INVALIDITY:(|D|I|)}TLB could contain **UNKNOWN** content. Fields:

EL unsigned int

Exception level of translation regime.

INVALIDITY unsigned int

Which TLB may be invalid (if TLBs are separate).

NON_SECURE bool

Whether the translation regime is non-secure.

PAS unsigned int

Physical address space of translation regime.

VMID unsigned int

Current VMID, if applicable.

ArchMsg.Warning.unknown_DLR_DSPSR

Exiting debug state with unknown DLR or DSPSR.

ArchMsg.Warning.unknown_ELR_SPSR

Returning from debug exception with unknown ELR or SPSR.

ArchMsg.Warning.unpred_FAR_on_LRCPC3_non_atomic_mem_access

TAGS **UNPREDICTABLE** Unpredictable FAR/FSTATUS on LRCPC3 pair instructions resulting in two separate memory accessesDISPLAY Unpredictable FAR on LRCPC3 non atomic memory access at %{ADDR1} with FSTATUS %{FSTATUS1} or %{ADDR2} with FSTATUS %{FSTATUS2}. Fields:

ADDR1 unsigned int

Possible FAR address 1.

ADDR2 unsigned int

Possible FAR address 2.

FSTATUS1 unsigned int

Possible fault status value 1.

FSTATUS2 unsigned int

Possible fault status value 2.

ArchMsg.Warning.unpred_async_tag_check_fail_on_sve_speculative_load_do_update_ffr

Unpredictable set to FALSE of the FFR element associated to SVE speculative load with an async tag check fail TAGS **UNPREDICTABLE**.

ArchMsg.Warning.unpredictable_a32_breakpoint

TAGS **UNPREDICTABLE** DISPLAY Unpredictable breakpoint on a A32 instruction at %{ADDR1} Offset %{OFFSET} BAS %{BAS} for a %{IS_ADDRESS_MISMATCH_BREAKPOINT:(Mismatch|Match)} Breakpoint number %{N:d} Breakpoint address %{ADDR2}. Fields:

ADDR1 unsigned int

Address.

ADDR2 unsigned int

Breakpoint address.

BAS unsigned int

BAS field.

IS_ADDRESS_MISMATCH_BREAKPOINT bool

Is a Mismatch breakpoint, not a Match breakpoint.

N unsigned int

Breakpoint number.

OFFSET unsigned int

Always has a value of zero.

ArchMsg.Warning.unpredictable_banked_register_access

TAGS **UNPREDICTABLE** DISPLAY Access to banked register (R=%{R:d} SYSm=%{SYSM:2x}) is unpredictable. Fields:

R unsigned int

R bit.

SYSM unsigned int

SYSm.

ArchMsg.Warning.unpredictable_ccfail_undef

TAGS **UNPREDICTABLE** DISPLAY An instruction %{OPCODE} at VA=%{VADDR} failed its condition codes check but would otherwise have caused a non-data-dependent trap or undefined exception. Behaviour is implementation-defined choice of exception or nop. Fields:

OPCODE unsigned int

opcode of the instruction.

VADDR unsigned int

Virtual address of the instruction being executed.

ArchMsg.Warning.unpredictable_ici_bits

TAGS **UNPREDICTABLE** DISPLAY ICI bits do not apply or are architecturally unpredictable for instruction.

ArchMsg.Warning.unpredictable_t32_breakpoint

TAGS **UNPREDICTABLE** DISPLAY Unpredictable breakpoint on a T32 instruction at %{ADDR1} Offset %{OFFSET} BAS %{BAS} for a %{IS_ADDRESS_MISMATCH_BREAKPOINT:(Mismatch|Match)} Breakpoint number %{N:d} Breakpoint address %{ADDR2}. Fields:

ADDR1 unsigned int

Address.

ADDR2 unsigned int

Breakpoint address.

BAS unsigned int

BAS field.

IS_ADDRESS_MISMATCH_BREAKPOINT bool

Is a Mismatch breakpoint, not a Match breakpoint.

N unsigned int

Breakpoint number.

OFFSET unsigned int

Indicates which halfword of a 32-bit Thumb instruction the breakpoint is set on. 0 for the first halfword or if the instruction is 16 bits wide, or 1 for the second halfword.

ArchMsg.Warning.unpredictable_t32_vector_catch

TAGS **UNPREDICTABLE** DISPLAY Unpredictable Vector catch exception on a T32 instruction at %{ADDR} Offset %{OFFSET}. Fields:

ADDR unsigned int

Address.

OFFSET unsigned int

Indicates which halfword of a 32-bit Thumb instruction the vector catch exception is set on. 0 for the first halfword or if the instruction is 16 bits wide, or 1 for the second halfword.

ArchMsg.Warning.unpredictable_watchpoint_far

TAGS **UNPREDICTABLE** DISPLAY The value for FAR/EDWAR for hitting this watchpoint may be anywhere in the range %{LOWER_BOUND} to %{UPPER_BOUND}. Fields:

LOWER_BOUND unsigned int

the lowest address accessed by the instruction that triggered the watchpoint.

UPPER_BOUND unsigned int

the highest watchpointed address accessed by that instruction.

ArchMsg.Warning.unpredictable_watchpoint_far_isscmp

DISPLAY The value for FAR/EDWAR for hitting this watchpoint may be anywhere in the range %{LOWER_BOUND} to %{UPPER_BOUND}. Fields:

LOWER_BOUND unsigned int

the lowest address accessed by the instruction that triggered the watchpoint.

UPPER_BOUND unsigned int

the highest watchpointed address accessed by that instruction.

ArchMsg.Warning.warning_AdvSIMDExpandImmUnexpectedZero

TAGS **UNPREDICTABLE** DISPLAY AdvSIMDExpandImm may treat this immediate value as **UNPREDICTABLE**.

ArchMsg.Warning.warning_ConditionalSMC

TAGS **UNPREDICTABLE** DISPLAY SMC instruction has **UNPREDICTABLE** effects when conditional and when combined with traps.

ArchMsg.Warning.warning_TLBID_out_of_range

TLBID out of range in TLBI according to the configuration of NIS, NVIS, NOS and NVOS. Fields:

NIS unsigned int

NIS.

NOS unsigned int

NOS.

NVIS unsigned int

NVIS.

NVOS unsigned int

NVOS.

TLBID unsigned int

The TLBID value used in the TLBI instruction.

ArchMsg.Warning.warning_TTL_or_TG_mismatches_current_translation_granule

TTL or TG mismatches current translation granule in TLBI VA/RVA/IPA/RIPA. Fields:

CURRENT_TG unsigned int

Current TG as indicated by TCR_ELx or VTCR_EL2.

TLBI_TG unsigned int

TG (or TTL[3:2]) used in TLBI instruction.

ArchMsg.Warning.warning_access_crosses_page_boundary

TAGS **UNPREDICTABLE** DISPLAY Access crosses page boundary.

ArchMsg.Warning.warning_access_crosses_page_boundary_has_fault

TAGS **UNPREDICTABLE** DISPLAY Access that crosses page boundary has fault(s).

ArchMsg.Warning.warning_access_crosses_page_boundary_has_fault_on_both_pages

TAGS **UNPREDICTABLE** DISPLAY Access that crosses page boundary has a fault on each page.

Fields:

ADDR1 unsigned int

FAR address on first page.

ADDR2 unsigned int

FAR address on second page.

FSTATUS1 unsigned int

Fault status on first page.

FSTATUS2 unsigned int

Fault status on second page.

ArchMsg.Warning.warning_access_crosses_page_boundary_spanning_different_memory

TAGS **UNPREDICTABLE** DISPLAY Access at address %{ADDR} crosses page from MEM_TYPE %{MEMTYPE_PAGE1} to %{MEMTYPE_PAGE2}. Fields:

ADDR unsigned int

address of access crossing page.

MEMTYPE_PAGE1 signed int

First Page's memory type (0-Normal, 1-Non-Normal).

MEMTYPE_PAGE2 signed int

Second Page's memory type (0-Normal, 1-Non-Normal).

ArchMsg.Warning.warning_access_wraps_around_memory

TAGS **UNPREDICTABLE** DISPLAY Access wraps around memory in %{WIDTH:d}bit mode.

Fields:

WIDTH unsigned int

Address width.

ArchMsg.Warning.warning_bcr_linking_status

TAGS **UNPREDICTABLE** DISPLAY DBGBCR.{BT2,BT} combination gives unpredictable behaviour.

ArchMsg.Warning.warning_bcr_mask_reserved

TAGS **UNPREDICTABLE** DISPLAY DBGBCR_ELn.MASK is set to a reserved value.

ArchMsg.Warning.warning_bvr_masked_bit_not_zero

TAGS **UNPREDICTABLE** DISPLAY DBGBVRn_EL1.VA contains a bit masked by DBGBCRn_EL1.MASK that is not zero.

ArchMsg.Warning.warning_ccsidr_unimplemented_level

TAGS **UNPREDICTABLE** DISPLAY CCSIDR read while CSSELR %{CSSELR:x} points to an unimplemented cache level. Fields:

CSSELR unsigned int

current effective value of CSSELR.

ArchMsg.Warning.warning_change_to_ns_when_tge_set

TAGS **UNPREDICTABLE** DISPLAY Attempting to change NS to 1 when HCR.TGE = 1.

ArchMsg.Warning.warning_contiguous_bit_check_abort

TAGS **UNPREDICTABLE** DISPLAY An abort occurred whilst attempting to check TLB entry is contiguous at %{ENTRY_ADDR}. Fields:

ENTRY_ADDR unsigned int

address of conflicting TLB entry.

ArchMsg.Warning.warning_contiguous_bit_error

TAGS **UNPREDICTABLE** DISPLAY TLB entry at %{ENTRY_ADDR} was not contiguous with entry at %{CONTIG_ADDR} Contents are %{ENTRY_DATA} but expected %{CONTIG_DATA}. Fields:

CONTIG_ADDR unsigned int

address of the TLB first read, with which this entry is expected to be contiguous.

CONTIG_DATA unsigned int

expected contents based on the entry at CONTIG_ADDR.

ENTRY_ADDR unsigned int

address of conflicting TLB entry.

ENTRY_DATA unsigned int

contents of conflicting TLB entry.

ArchMsg.Warning.warning_contiguous_bit_error_D128

TAGS **UNPREDICTABLE** DISPLAY TLB entry at %{ENTRY_ADDR} was not contiguous with entry at %{CONTIG_ADDR} Contents are %{ENTRY_DATA_HI} %{ENTRY_DATA_LO} but expected %{CONTIG_DATA}. Fields:

CONTIG_ADDR unsigned int

address of the TLB first read, with which this entry is expected to be contiguous.

CONTIG_DATA unsigned int

expected contents based on the entry at CONTIG_ADDR.

ENTRY_ADDR unsigned int

address of conflicting TLB entry.

ENTRY_DATA_HI unsigned int

Higher bit contents of conflicting TLB entry.

ENTRY_DATA_LO unsigned int

Lower bit contents of conflicting TLB entry.

ArchMsg.Warning.warning_cp10_cp11_mismatch

TAGS **UNPREDICTABLE** DISPLAY Unpredictable configuration of CP10 and CP11 controls at EL %{EL:d}. Fields:

EL unsigned int

EL owning the control.

ArchMsg.Warning.warning_cp10_cp11_reserved_value

TAGS **UNPREDICTABLE** DISPLAY Unpredictable configuration of CPACR CP10 ({CP10}) or CP11 ({CP11}). Fields:

CP10 unsigned int

CP10 access.

CP11 unsigned int

CP11 access.

ArchMsg.Warning.warning_csselr_level_out_of_range

TAGS **UNPREDICTABLE** DISPLAY CSSELR.Level out of range, written value {LEVEL} with only {IMPLEMENTED:d} levels implemented. Fields:

IMPLEMENTED unsigned int

number of cache levels implemented.

LEVEL unsigned int

written value of CSSELR.Level.

TYPE enum

The type of cache selected by CSSELR.InD and TnD.

ArchMsg.Warning.warning_debug_flow_control_bits_not_obeyed

TAGS **UNPREDICTABLE** DISPLAY Flow control bits for {REGISTER} were not obeyed, causing {ERROR}. Fields:

ERROR enum

type of error resulting.

REGISTER enum

register under consideration.

ArchMsg.Warning.warning_debug_register_access_during_reset

TAGS **UNPREDICTABLE** DISPLAY Unpredictable {IS_WRITE:(read|write)} access to debug register offset {OFFSET} during reset. Fields:

IS_WRITE unsigned int

Write Not Read.

OFFSET unsigned int

Register Offset.

ArchMsg.Warning.warning_decode_cps_inconsistent_fields

TAGS **UNPREDICTABLE** DISPLAY The mode setting fields in the CPS instruction are inconsistent.

ArchMsg.Warning.warning_decode_invalid_state

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable in current exception-level/security state.

ArchMsg.Warning.warning_default_cacheable_mmu_on

TAGS **UNPREDICTABLE** DISPLAY Default cacheable is enabled (HCR.DC = 1) while MMU is enabled.

ArchMsg.Warning.warning_default_cacheable_vmmu_off

TAGS **UNPREDICTABLE** DISPLAY Default cacheable is enabled (HCR.DC = 1) while HCR.VM = 0.

ArchMsg.Warning.warning_deprecated_wvr_bit_2_set

TAGS **UNPREDICTABLE** DISPLAY DBGWVR%{N:d}_EL1=%{WVR:h} has bit 2 set, deprecated in ARMv8. Fields:

N unsigned int

index.

WVR unsigned int

DBGWVR.

ArchMsg.Warning.warning_eret_while_software_step_active_pending

Missing ISB between setting MDSCR_EL1.SS and ERET TAGS **UNPREDICTABLE**.

ArchMsg.Warning.warning_exclusive_to_non_normal

TAGS **UNPREDICTABLE** DISPLAY Exclusive Access to Non-Normal Memory %{ADDR}. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_exclusive_to_non_writeback

TAGS **UNPREDICTABLE** DISPLAY Exclusive Access to Non-Writeback-Cacheable Memory %{ADDR}. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_execute_from_device_memory

TAGS **UNPREDICTABLE** DISPLAY An attempt was made to execute from Device memory.

ArchMsg.Warning.warning_illegal_cpsr_mode

TAGS **UNPREDICTABLE** DISPLAY Writing an illegal or unimplement mode (%{NEW_MODE:h}) to CPSR was unpredictable. Fields:

NEW_MODE unsigned int

New value of CPSR.M.

ArchMsg.Warning.warning_illegal_srs_mode

TAGS **UNPREDICTABLE** DISPLAY Illegal or **UNPREDICTABLE** mode (%{MODE:h}) used for SRS instruction. Fields:

MODE unsigned int

mode for Banked SP.

ArchMsg.Warning.warning_implementation_defined_read_debug_register_in_SCS

Read Access to Debug registers from the processor is IMPLEMENTATION DEFINED TAGS IMP_DEF. Fields:

PPB_OFFSET unsigned int
debug register offset.

ArchMsg.Warning.warning_implementation_defined_sequential_security_transitions_supported

The behavior of sequential instruction fetches that cross from non-secure to secure memory and contain an SG instruction is CONSTRAINED UNPREDICTABLE TAGS UNPREDICTABLE.

ArchMsg.Warning.warning_implementation_defined_stack_limit_check_supported

It is IMPLEMENTATION DEFINED whether stack pointer limit checking is performed for this instruction TAGS IMP_DEF. Fields:

ADDRESS unsigned int
Address of instruction.

OPCODE unsigned int
instruction opcode.

ArchMsg.Warning.warning_implementation_defined_taggednonsharedmemorytype

TAGS IMP_DEF DISPLAY Memory attribute has Tagged, non shared memory type; it is IMPLEMENTATION DEFINED whether the memory location is treated as Tagged or non-Tagged.

ArchMsg.Warning.warning_implementation_defined_write_debug_register_in_SCS

Write Access to Debug registers from the processor is IMPLEMENTATION DEFINED TAGS IMP_DEF. Fields:

DATA unsigned int
data attempted to be written.

PPB_OFFSET unsigned int
debug register offset.

ArchMsg.Warning.warning_invalid_tcr_granule

TAGS UNPREDICTABLE DISPLAY TCR.TG%{TG_ID:(0|1)} bits have been set to a granule size %{{REQUEST:d}}K not implemented - using %{{SUBSTITUTE:d}}K. Fields:

REQUEST signed int
page size requested (or 0 for reserved).

SUBSTITUTE signed int
best guess available page size.

TG_ID bool
bits TG0 or TG1.

ArchMsg.Warning.warning_load_multiple_user_registers_from_user_mode

TAGS UNPREDICTABLE DISPLAY An LDM(user registers) instruction executed from user mode.

ArchMsg.Warning.warning_mte_in_mem_attr_not_supported

TAGS UNPREDICTABLE DISPLAY Memory attribute %{ATTR} has Tagged memory type but Armv8.5-MTE is not implemented. Fields:

ATTR unsigned int

Memory attribute.

ArchMsg.Warning.warning_reserved_breakpoint_state_match

TAGS UNPREDICTABLE DISPLAY %{IS_BREAKPOINT:(Watchpoint|Breakpoint)} programmed with a reserved combination of HMC, SSC and %{IS_BREAKPOINT:(PAC|PMC)}. Fields:

IS_BREAKPOINT unsigned int

Is a breakpoint, not a watchpoint.

ArchMsg.Warning.warning_shareability

TAGS UNPREDICTABLE DISPLAY Unpredictable: combination of the 1st and 2nd stages of translation is Normal Inner Non-Cacheable, Outer Non-Cacheable.

ArchMsg.Warning.warning_software_step_set_while_enabled

TAGS UNPREDICTABLE DISPLAY MDSCR_EL1.SS set to 1 while software step debug exceptions are enabled.

ArchMsg.Warning.warning_thumb_instruction_wraps_around_memory

TAGS UNPREDICTABLE DISPLAY Thumb instruction wraps around memory.

ArchMsg.Warning.warning_ttbr_sbz_bits_are_not_zero

TAGS UNPREDICTABLE DISPLAY %{TTBR}[%{MSB:d}:0] should be zero, but are not. Fields:

MSB unsigned int

x-1.

TTBR enum

which TTBR register.

ArchMsg.Warning.warning_unaligned_address_dbgdtrrx_write

TAGS UNPREDICTABLE DISPLAY Unaligned address %{ADDR} in X0/R0 when DBGDTRRX written in memory access mode. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unaligned_address_dbgdtrtx_read

TAGS UNPREDICTABLE DISPLAY Unaligned address %{ADDR} in X0/R0 when DBGDTRTX read in memory access mode. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unaligned_to_device

TAGS UNPREDICTABLE DISPLAY Unaligned Access to Device Memory %{ADDR}. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unaligned_to_strongly_ordered

TAGS **UNPREDICTABLE** DISPLAY Unaligned Access to Strongly Ordered Memory %{ADDR}.

Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unknown_sau_rnr

TAGS **UNPREDICTABLE** DISPLAY SAU_RNR was set to an unsupported value.

ArchMsg.Warning.warning_unpredictable_AIRCR_PRIS_and_BFHFNMINs

TAGS **UNPREDICTABLE** DISPLAY The effect of setting both AIRCR.BFHFNMINs and AIRCR.PRIS to 1 is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_AIRCR_VECTCLRACTIVE_when_not_in_debug

TAGS **UNPREDICTABLE** DISPLAY The effect of writing a 1 to AIRCR.VECTCLRACTIVE if the processor is not halted in Debug state is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_AIRCR_VECTRESET_and_SYSRESETREQ

TAGS **UNPREDICTABLE** DISPLAY When the processor is halted in Debug state, if a write to the register writes a 1 to both VECTRESET and SYSRESETREQ, the behavior is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_AIRCR_VECTRESET_when_not_in_debug

TAGS **UNPREDICTABLE** DISPLAY The effect of writing a 1 to AIRCR.VECTRESET if the processor is not halted in Debug state is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_AIRCR_incorrect_VKEY

TAGS **UNPREDICTABLE** DISPLAY The value 0x05FA must be written to AIRCR.VKEY, otherwise the register write is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_EXC_RETURN_Reserved_Bit

EXC_RETURN[23:7] are reserved with the special condition that all bits should be written as one. Values other than all 1s are **UNPREDICTABLE**. TAGS **UNPREDICTABLE**. Fields:

EXC_RETURN unsigned int

Special PC value.

SBO_MASK unsigned int

Which bits Should Be One to not UNPRED.

ArchMsg.Warning.warning_unpredictable_PMEVTYPEPERn_ELO_VS_reserved_value

TAGS **UNPREDICTABLE** DISPLAY Unpredictable reserved value of PMEVTYPER<n>_ELO.VS.

ArchMsg.Warning.warning_unpredictable_change_to_DEMCR_MON_STEP_at_insufficient_priority

TAGS **UNPREDICTABLE** DISPLAY The effect of changing DEMCR.MON_STEP at an execution priority that is lower than the priority of the DebugMonitor exception is **CONSTRAINED UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_change_to_priority_of_active_exception

TAGS UNPREDICTABLE DISPLAY Changing the priority of an active exception is UNPREDICTABLE.

ArchMsg.Warning.warning_unpredictable_clear_lspact

TAGS UNPREDICTABLE DISPLAY UNPREDICTABLE state on context switch. Fields:

DRVEXC_TAKEN unsigned int

Derived exception taken, not pended.

REACHED_LAST_STORE unsigned int

Only the last faulted.

ArchMsg.Warning.warning_unpredictable_dbg_exit_unaligned_dlr

TAGS UNPREDICTABLE DISPLAY Debug exit to AArch32 with DLR[0] set to 1. PC[0] can be set to 0 or to DLR[0].

ArchMsg.Warning.warning_unpredictable_exception_catch

TAGS UNPREDICTABLE DISPLAY Generation of an exception catch event is unpredictable.

ArchMsg.Warning.warning_unpredictable_exception_return_inconsistent_state

TAGS UNPREDICTABLE DISPLAY state on exception return is unpredictable.

ArchMsg.Warning.warning_unpredictable_exception_return_instruction

TAGS UNPREDICTABLE DISPLAY exception return instruction is unpredictable.

ArchMsg.Warning.warning_unpredictable_in_debug_state

TAGS UNPREDICTABLE DISPLAY Instruction is UNPREDICTABLE when executed in debug state.

ArchMsg.Warning.warning_unpredictable_mair_encoding

TAGS UNPREDICTABLE DISPLAY Memory attribute %{ATTR} has invalid MAIR encoding; this may lead to unpredictable results. Fields:

ATTR unsigned int

Memory attribute.

ArchMsg.Warning.warning_unpredictable_pmu_counter_access

TAGS UNPREDICTABLE DISPLAY Access to PMU counters from non-secure EL0 or EL1 is UNPREDICTABLE with MDCR_EL2.HPMN set to 0.

ArchMsg.Warning.warning_unpredictable_prioritization_breakpoint_match_vector_catch

TAGS UNPREDICTABLE DISPLAY Two events with the same priority occurred at the same instruction: Address Matching Vector Catch debug event, and Breakpoint debug event. It is constrained unpredictable which is taken.

ArchMsg.Warning.warning_unpredictable_stack_selection

TAGS UNPREDICTABLE DISPLAY UNPREDICTABLE state on context switch.

ArchMsg.Warning.warning_unpredictable_tsize_out_of_range

TAGS UNPREDICTABLE DISPLAY %{TCR}.T%{TTBR:d}SZ (%{TSIZE:d} bits) is out of range. Must be between 25 and %{TSIZE_MAX:d}. Fields:

TCR enum

Which TCR.

TSize signed int

Value of TSize.

TSize_MAX signed int

The maximum allowed value.

TTBR unsigned int

Which TTBR region.

ArchMsg.Warning.warning_unpredictable_unaligned_pc_as_base_register

TAGS **UNPREDICTABLE** DISPLAY The PC value must be word-aligned, otherwise the behavior of the instruction is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_unaligned_pop_stack

TAGS **UNPREDICTABLE** DISPLAY **CONSTRAINED UNPREDICTABLE** unaligned stack frame `%{FRAME_ADDR:x}` in exception-return. Fields:

FRAME_ADDR unsigned int

unaligned stack frame address.

ArchMsg.Warning.warning_unpredictable_unaligned_to_non_writeback

TAGS **UNPREDICTABLE** DISPLAY Unaligned Access to Non-Writeback-Cacheable Memory.

ArchMsg.Warning.warning_unpredictable_vmsa_memattrib

TAGS **UNPREDICTABLE** DISPLAY Unpredictable vmsa memory attribute with texcb value programmed to `%{TEXCB}` when tex remap is `%{TEX_REMAP}`. Fields:

TEXCB unsigned int

texcb value.

TEX_REMAP bool

use tex remap.

ArchMsg.Warning.warning_unpredictable_vtcr_t0sz_sl0

TAGS **UNPREDICTABLE** DISPLAY The combination of VTCR.TOSZ=`%{TOSZ:d}` and VTCR.SL0=`%{SL0:d}` is unpredictable. Fields:

SL0 signed int

VTCR.SL0, starting level for stage 2 translation table walks.

TOSZ signed int

VTCR.TOSZ, required input address range.

ArchMsg.Warning.warning_unpredictable_write_DHCSR

TAGS **UNPREDICTABLE** DISPLAY **UNPREDICTABLE** set of bit-changes in DHCSR. Fields:

NEW unsigned int

data attempted to be written.

OLD unsigned int

previous value.

ArchMsg.Warning.warning_unpredictable_write_SHCSR

TAGS UNPREDICTABLE DISPLAY UNPREDICTABLE setting a pending bit to 1 for an exception with priority \geq execution priority.

ArchMsg.Warning.warning_unsupported_access_to_memory_mapped_register

TAGS UNPREDICTABLE DISPLAY Access to memory mapped register is unsupported.

ArchMsg.Warning.warning_user_jmcr_access

TAGS UNPREDICTABLE DISPLAY User %{WRITE:(write|read)} access to JMCR is unpredictable. Fields:

WRITE unsigned int
access is write.

ArchMsg.Warning.warning_wcr_mask_and_bas

TAGS UNPREDICTABLE DISPLAY DBGWCR_EL1.MASK is non-zero and BAS is not 0xff.

ArchMsg.Warning.warning_wcr_mask_reserved

TAGS UNPREDICTABLE DISPLAY DBGWCR_ELn.MASK is set to a reserved value.

ArchMsg.Warning.warning_wcr_non_configuous_bas

TAGS UNPREDICTABLE DISPLAY DBGWCR_EL1.BAS has non-contiguous set of ones.

ArchMsg.Warning.warning_wvr_masked_bit_not_zero

TAGS UNPREDICTABLE DISPLAY DBGWVRn_EL1.VA contains a bit masked by DBGWCRn_EL1.MASK that is not zero.

ArchMsg.Why.why_branch_target_exception

A branch target exception has occurred. DISPLAY branch target exception: target pc=%{PC}, target opcode=%{OPCODE}. PRIMARY KEY PC. Fields:

BTYPE unsigned int
pstate.BTYPE at time branch taken.

OPCODE unsigned int
opcode at branch target.

PC unsigned int
Virtual Address of branch target.

ArchMsg.Why.why_illegal_state

An illegal mode change has occurred. DISPLAY illegal mode change: %{MESSAGE}. Fields:

MESSAGE string
The message.

BRANCH_MISPREDICT

Simulating branch mispredict. Fields:

PC unsigned int
Origin address (or 0 if unavailable).

BRA_DIR

Direct branches, to immediate address. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISET enum

The instruction set of the branch instruction.

IS_COND bool

Indicates if this is a conditional branch.

IS_HINTED bool

Whether the conditional instruction is hinted.

IS_ISB bool

Whether the instruction is ISB.

IS_LINK bool

Indicates if this is a branch with link.

PC unsigned int

The address of the branch instruction.

TARGET_ISET enum

The instruction set after the branch.

TARGET_PC unsigned int

The address the instruction branches to.

BRA_INDIR

Indirect branches, perhaps to a register. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISET enum

The instruction set of the branch instruction.

IS_COND bool

Indicates if this is a conditional branch.

IS_HINTED bool

Whether the conditional instruction is hinted.

IS_ISB bool

Whether the instruction is ISB.

IS_LINK bool

Indicates if this is a branch with link.

PC unsigned int

The address of the branch instruction.

TARGET_ISET enum

The instruction set after the branch.

TARGET_PC unsigned int

The address the instruction branches to.

BRA_INDIR_CCFAIL

Indirect branches fail, perhaps to a register.

BRA_RET_CCFAIL

Branch return fail, perhaps to a register.

CACHE_MAINTENANCE_OP

Cache Maintenance Operation. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

DATA unsigned int

Specified MVA or Set/way.

FUNCTION enum

Clean / Invalidate.

SCOPE enum

Affected region.

SIDE enum

Inst / Data.

CCFAIL

Conditional instruction condition check fail. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CCFAIL_UNC

Unconditional instruction condition check fail. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CCPASS

Conditional instruction condition check pass. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CCPASS_UNC

Unconditional instruction condition check pass. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CODE_CACHE_MAINT

Code cache maintenance. Fields:

END_INCL unsigned int

Inclusive end address for VA or PA requests.

NS bool

Non-secure world for PA.

START unsigned int

Start address for VA or PA requests.

TYPE enum

Request type.

COMPILE_BLOCK_END

Last instruction of basic block translated. Fields:

VADDR unsigned int

Address of next instruction after this basic block.

COMPILE_BLOCK_START

First instruction of basic block translated. Fields:

VADDR unsigned int

Address of first instruction in basic block.

COMPILE_INST

ARM instruction compiled. Fields:

DISASS string

Disassembly of the instruction.

ISSET enum

The instruction set of this instruction.

ITSTATE unsigned int

The ITSTATE current for the instruction.

OPCODE unsigned int

The opcode of the instruction.

PC unsigned int

The address of the instruction.

SIZE unsigned int

The size of the instruction in bytes.

CONTEXTIDR

Context ID Register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

NS bool

Secure or nonsecure banked register is accessed.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The new value written.

CONTEXT_SYNC

Called for every context synchronization event. Fields:

ARCH_CSE bool

Architectural Context Synchronization Event.

PC unsigned int

The address of the synchronization instruction.

SYSREG_SYNC bool

System register synchronization event.

CORE_ENDIAN

Core BE8 Big-Endian state changed. Fields:

BE8 bool

Core BE8 Big-Endian state.

CORE_INFO

Static processor attributes. Only triggered by a call to DumpState(). Fields:

ARCH_PROFILE enum

The architecture profile of the core.

CLUSTER_ID unsigned int

The cluster ID of this processor.

CORE_NUM unsigned int

The number of this core in an MP processor.

DETAILED_CORE_FEATURES enum

Detailed core features as a bitfield. Enum values are bitmasks to extract the info from the field.

FPU_VERSION enum

The VFP version implemented by the core.

MEM_ARCH enum

The memory architecture of the core.

MIDR_EL1 unsigned int

MIDR_EL1 value.

NUM_CORES unsigned int

The number of cores in this MP processor.

QUANTUM_SIZE unsigned int

The default quantum size of the core (in ticks).

SECURITY_FEATURES bool

Does the core have security features?.

CORE_LOADS

Processor load accesses. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ACQREL enum

Is this an acquire/release.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DATA unsigned int

The data read or written.

ELEMENT_SIZE unsigned int

Width of each element.

ENDIANNESS enum

Byte ordering of this access.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

LOCK enum

Normal, exclusive or locked access.

MECID unsigned int

Memory Encryption Context ID, if applicable.

MECID2 unsigned int

Memory Encryption Context ID of the second page, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PADDR_TAG unsigned int

If tag checked, the physical address tag for the access.

PAGE_SIZE unsigned int

Page size as $\log_2(\text{size in bytes})$.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK, 2=Exclusive Failed, 3=Unsupported.

SIZE unsigned int

Width of the access in bytes. Only required if DATA is not traced.

TAG_CHECKED bool

Is this access tag checked.

TAG_CHECKED2 bool

Is the second page access tag checked.

TRANS bool

Is this a translated access.

VADDR unsigned int

The virtual address of the access.

CORE_REGS

Changes of the core registers R0 to R14. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number, 0 to 14.

MODE enum

Bank of the register accessed.

OLD_VALUE unsigned int

The old value overwritten.

PHYS_ID enum

The physical register accessed.

VALUE unsigned int

The new value written to the register.

CORE_REGS64

Changes of the core registers X0..X30, SP_ELn. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ID enum

The register number, 0-30 for X0-X30, >=32 for SP_ELn.

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

CORE_REGS64_READ

General purpose AArch64 register has been read X0..X30, SP_ELn. Fields:

ID enum

The register number, 0-30 for X0-X30, >=32 for SP_ELn.

VALUE unsigned int

The value read from the register.

CORE_STALLED_TRANSITION

Detect when a core is prevented from progressing PChannel states. Fields:

CURRENT_STATE enum

Current power mode of core.

REASON string

The reason a transition was stalled, and what must be done to unblock it.

TARGET_STATE enum

Desired power mode of core after transition.

CORE_STORES

Processor store accesses. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ACQREL enum

Is this an acquire/release.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DATA unsigned int

The data read or written.

ELEMENT_SIZE unsigned int

Width of each element.

ENDIANNESS enum

Byte ordering of this access.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

LOCK enum

Normal, exclusive or locked access.

MECID unsigned int

Memory Encryption Context ID, if applicable.

MECID2 unsigned int

Memory Encryption Context ID of the second page, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PADDR_TAG unsigned int

If tag checked, the physical address tag for the access.

PAGE_SIZE unsigned int

Page size as $\log_2(\text{size in bytes})$.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK, 2=Exclusive Failed, 3=Unsupported.

SIZE unsigned int

Width of the access in bytes. Only required if DATA is not traced.

TAG_CHECKED bool

Is this access tag checked.

TAG_CHECKED2 bool

Is the second page access tag checked.

TRANS bool

Is this a translated access.

VADDR unsigned int

The virtual address of the access.

CP14_READ

System Coprocessor register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP14 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc1 unsigned int

opcode 1.

opc2 unsigned int

opcode 2.

CP14_WRITE

System Coprocessor register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP14 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc1 unsigned int

opcode 1.

opc2 unsigned int

opcode 2.

CP15_READ

System Coprocessor register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP15 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc1 unsigned int

opcode 1.

opc2 unsigned int

opcode 2.

CP15_READ64

System Coprocessor register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP15 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc unsigned int

opcode 1.

CP15_WRITE

System Control Coprocessor register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP15 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc1 unsigned int

opcode 1.

opc2 unsigned int

opcode 2.

CP15_WRITE64

System Coprocessor register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP15 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc unsigned int

opcode 1.

CPSR

CPSR change. (In AArch64 PSTATE represented in SPSR format). Fields:

CORE_NUM unsigned int

Core number in a multi processor.

OLD_VALUE unsigned int

The old CPSR value.

UNKNOWN unsigned int

Bits within the register that have unknown value.

VALUE unsigned int

The new CPSR value.

CRYPTO_SPEC

Every crypto instruction speculatively executed.

CorePowerStateChange

Core power-up state has changed. DISPLAY Core powered %{PU:(up|down)}. Fields:

PU bool

Whether the core powered up.

DATA_CACHE_SET_ALLOC_TAGS

Set allocation tags from DC GVA instruction. Fields:

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

NSDESC unsigned int

The security state of the access.

PADDR unsigned int

The physical address of the access.

PAS enum

The physical address space of the page.

RESPONSE enum

0=Aborted, 1=OK.

SIZE unsigned int

The size of memory in bytes.

TAG unsigned int

The value of the tag being set.

TAGGING_TYPE enum

Type of Memory tagging (Physical or Virtual).

TAG_VADDR unsigned int

The virtual address of the Virtual Tag if Virtual tagging is used; otherwise 0.

VADDR unsigned int

Virtual address used for setting allocation tags.

DATA_CACHE_ZERO

Zero and invalidate cache line from DC ZVA instruction. Fields:

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

NSDESC unsigned int

The security state of the access.

PADDR unsigned int

The physical address of the access.

PADDR_TAG unsigned int

If tag checked, the physical address tag for the access.

PAS enum

The physical address space of the page.

RESPONSE enum

0=Aborted, 1=OK.

SIZE unsigned int

The size of zero'd memory in bytes.

TAG_CHECKED bool

Is this access tag checked.

VADDR unsigned int

Virtual address of the zero'd cache line.

DEBUG_EVENT

Hardware debug support event. Fields:

EVENT enum

Description of event.

VALUE unsigned int

data value.

DEBUG_TARGET_ISET

Hardware debug target ISA. Fields:

TARGET_ISET enum

The target instruction set of the debug state.

DMI_ALLOCATE

DMI allocated. Fields:

ACCESS_DIRECTION enum

READ / WRITE.

PADDR unsigned int

The physical address of the DMI.

PTR unsigned int

DMI range pointer.

SIDE enum

DSIDE / ISIDE.

SIZE unsigned int

The size of DMI.

VADDR unsigned int

The virtual address of the DMI..

DMI_HIT

DMI hit, access used DMI (only fire for small DMI regions i.e. less than a full page). Fields:

ACCESS_DIRECTION enum

READ / WRITE.

PADDR unsigned int

The physical address of the DMI.

SIDE enum

DSIDE / ISIDE.

SIZE unsigned int

The size of the DMI.

VADDR unsigned int

The virtual address of the DMI..

DMI_REVOKE

DMI revoked. Fields:

ACCESS_DIRECTION enum

READ / WRITE.

PADDR unsigned int

The physical address of the DMI.

SIDE enum

DSIDE / ISIDE.

SIZE unsigned int

The size of DMI.

VADDR unsigned int

The virtual address of the DMI..

END_COMPILE

Compilation end. Fields:

END_OF_PAGE_COUNT unsigned int

Number of basic blocks exited due to page boundary since START_COMPILE.

FETCHFAIL_COUNT unsigned int

Number of basic blocks exited due to ifetch failure START_COMPILE.

INST_COUNT unsigned int

Number of instructions compiled since START_COMPILE.

NONSEQ_COUNT unsigned int

Number of basic blocks exited due to non-sequential instructions since START_COMPILE.

PAGE_STRADDLE_COUNT unsigned int

Number of basic blocks exited due to unaligned instructions crossing page since START_COMPILE.

ESB

Error Synchronization Barrier hit. Fields:

ASYNC bool

ESB is async.

IMPLICIT bool

ESB is implicit.

EXCEPTION

Exception taken. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ESR unsigned int

The value assigned to ESR when exception is taken.

ESR_EC unsigned int

The Value assigned to ESR.EC when exception is taken.

IS_PHYSICAL bool

Physical or Virtual exception.

IS_TRAP bool

Exception coming from a trap.

LR unsigned int

The value assigned to the link register.

PC unsigned int

The location where the exception occurred.

PREFERRED_RETURN unsigned int

The preferred return address for the exception.

TAKEN_LOCALLY bool

Whether exception is taken locally(same exception level).

TARGET_ISET enum

The instruction set of the exception handler code.

TARGET_PC unsigned int

The address the exception branches to.

VECTOR enum

The exception vector.

VECTOR_OFFSET unsigned int

The offset of an exception vector from the base address.

EXCEPTION_END

Every exception completed.

EXCEPTION_INFO

Trace human-readable reason of the current exception for data/instruction aborts, floating point exceptions and unknown exceptions. Fields:

DESCRIPTION string

Reason for current exception.

ESR_EC unsigned int

The value of ESR.EC when exception is taken.

EXCEPTION_RAISE

Every exception raised.

EXCEPTION_RETURN

Branches on leaving exception. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISET enum

The instruction set of the branch instruction.

IS_COND bool

Indicates if this is a conditional branch.

IS_HINTED bool

Whether the conditional instruction is hinted.

IS_ISB bool

Whether the instruction is ISB.

IS_LINK bool

Indicates if this is a branch with link.

PC unsigned int

The address of the branch instruction.

TARGET_ISET enum

The instruction set after the branch.

TARGET_PC unsigned int

The address the instruction branches to.

EXCEPTION_RETURN_PREBRANCH

Exception return event that must occur before PC and processor state have updated.

EXCEPTION_START

Every exception started.

FIQ_TAKEN

FIQ taken exception.

FP_EXCEPTIONS_SIGNALLED

Trace which FP exceptions were signalled as a part of the FP operations performed by an instruction. Fields:

FP_EXCEPTIONS_SIGNALLED_MASK unsigned int

Indicates which FP exceptions were signalled by an instruction. This uses the same encoding as the cumulative floating-point exception bitfields in FPSR.

FP_STATE

Floating point state. Fields:

FPSR unsigned int

FPSR register value.

MASK unsigned int

The mask indicating updated cumulative bits.

FREQ_CHANGED

Core frequency changed. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The instruction count of this core.

NEW_FREQ signed int

The new frequency of this core (expressed in Hz).

OLD_FREQ signed int

The old frequency of this core (expressed in Hz).

HLT

HLT instruction occurred. Fields:

IMM16 unsigned int

The 16-bit immediate value encoded in HLT instruction.

INST

Every instruction executed. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DEBUG_STATE bool

The instruction is executed in debug state.

DISASS string

Disassembly of instruction.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISSET enum

The current instruction set.

ITSTATE unsigned int

The current ITSTATE.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

MODE enum

The mode the core is in.

NS unsigned int

The core's non-secure bit.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

OPCODE unsigned int

The opcode of the instruction.

PADDR unsigned int

The physical address of the instruction.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the instruction.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the page.

PC unsigned int

The address of the instruction.

SECURITY_STATE enum

The core's security state.

SIZE unsigned int

The size of the instruction in bytes.

INST_END

Every instruction completed.

INST_START

Every instruction started. Fields:

DEBUG_STATE bool

The instruction is executed in debug state.

ISSET enum

The current instruction set.

MODE enum

The mode the core is in.

NS enum

The current Secure State.

PC unsigned int

The address of the conditional instruction.

SECURITY_STATE enum

The current Security State.

INST_STRADDLE

Instruction straddles boundary. Fields:

DEBUG_STATE bool

The instruction is executed in debug state.

ISSET enum

The current instruction set.

MODE enum

The mode the core is in.

NS enum

The current Secure State.

PC unsigned int

The address of the conditional instruction.

SECURITY_STATE enum

The current Security State.

IRQ_TAKEN

IRQ taken exception.

ISB

ISB instruction occurred. Fields:

PC unsigned int

The address of the ISB instruction.

LOAD_MULTIPLE_REGS_LIST

List of destination registers (not source) of a multiple registers load instruction. Fields:

REG string

Register to add into the list.

LOCAL_MONITOR

Local monitor activity. Fields:

PADDR unsigned int

Local Monitor Address.

State enum

State of the monitor (Open/Exclusive).

MAPPING_CREATE

A memory mapping has been created. This is an implementation detail of the model and so may change without notice. Fields:

ASID unsigned int

The ASID used during translation, or 0 if no ASID was used.

IS_VALID bool

Whether the mapping is valid or not.

PADDR unsigned int

Base physical address of region, or 0 if not known.

SIDE enum

I-side or D-side.

SIZE unsigned int

Size of the page, or 0 if not known.

VADDR unsigned int

Base virtual address of region.

VMID unsigned int

The VMID used during translation, or 0 if no VMID was used.

MAPPING_DESTROY

A memory mapping has been destroyed. This is an implementation detail of the model and so may change without notice. Fields:

ASID unsigned int

The ASID used during translation.

CAUSE enum

The cause of invalidation.

PADDR unsigned int

Base physical address of page.

SIDE enum

I-side or D-side.

SIZE unsigned int

Size of the page.

VADDR unsigned int

Base virtual address of page.

VMID unsigned int

The VMID used during translation.

MEMMAP_DEBUG_READ

Memory mapped reads to the debug registers. Fields:

ADDR unsigned int

address.

CORE_NUM unsigned int

Core number in a multi processor.

EXT bool

Whether access is from an external device (such as the DAP).

NS enum

Secure state of the access.

REG_NAME string

Name of the debug register accessed.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

MEMMAP_DEBUG_WRITE

Memory mapped writes to the debug registers. Fields:

ADDR **unsigned int**

address.

CORE_NUM **unsigned int**

Core number in a multi processor.

EXT **bool**

Whether access is from an external device (such as the DAP).

NS **enum**

Secure state of the access.

REG_NAME **string**

Name of the debug register accessed.

UNDEF **bool**

The register accessed is undefined.

VALUE **unsigned int**

The new value written.

MEMTAG_LOADS

Allocation tag memory read; when attributes and system configuration allow tag accesses.
Fields:

ACCESS_TYPE **enum**

The type of instruction performing the access.

ATTR **unsigned int**

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 **unsigned int**

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

NSDESC **unsigned int**

The physical address non-secure bit.

NSDESC2 **unsigned int**

The second page physical address non-secure bit.

NUM_GRANULES **unsigned int**

Number of tags to be Loaded/stored.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK.

TAGGING_TYPE enum

Type of Memory tagging (Physical or Virtual).

TAGS unsigned int

The tags read or written. Tags for subsequent granules would be found at subsequent bytes in this field (One byte for each tag).

TAG_VADDR unsigned int

The virtual address of the Virtual Tag if Virtual tagging is used; otherwise 0.

VADDR unsigned int

The virtual address of the access.

MENTAG_LOAD_INST

Allocation tag load instruction; regardless of tags being actually accessed. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

NUM_GRANULES unsigned int

Number of tags to be Loaded/stored.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK.

TAGGING_TYPE enum

Type of Memory tagging (Physical or Virtual).

TAGS unsigned int

The tags read or written. Tags for subsequent granules would be found at subsequent bytes in this field (One byte for each tag).

TAG_VADDR unsigned int

The virtual address of the Virtual Tag if Virtual tagging is used; otherwise 0.

VADDR unsigned int

The virtual address of the access.

MEMTAG_STORES

Allocation tag memory write; when attributes and system configuration allow tag accesses.

Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

NUM_GRANULES unsigned int

Number of tags to be Loaded/stored.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK.

TAGGING_TYPE enum

Type of Memory tagging (Physical or Virtual).

TAGS unsigned int

The tags read or written. Tags for subsequent granules would be found at subsequent bytes in this field (One byte for each tag).

TAG_VADDR unsigned int

The virtual address of the Virtual Tag if Virtual tagging is used; otherwise 0.

VADDR unsigned int

The virtual address of the access.

MEMTAG_STORE_INST

Allocation tag store instruction; regardless of tags being actually accessed. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

NUM_GRANULES unsigned int

Number of tags to be Loaded/stored.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK.

TAGGING_TYPE enum

Type of Memory tagging (Physical or Virtual).

TAGS unsigned int

The tags read or written. Tags for subsequent granules would be found at subsequent bytes in this field (One byte for each tag).

TAG_VADDR unsigned int

The virtual address of the Virtual Tag if Virtual tagging is used; otherwise 0.

VADDR unsigned int

The virtual address of the access.

MMU_D128_TTB_READ

This event is triggered by reads caused by a D128 translation table walk. Fields:

ABORTED bool

Was the walk successful.

DATA_HIGH unsigned int

The data upper bits read.

DATA_LOW unsigned int

The data lower bits read.

IPA unsigned int

For stage 1, the IPA of the read.

LEVEL unsigned int

Translation table level.

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

OUTADDR unsigned int

The output physical address.

PADDR unsigned int

The physical address of the read.

PASmem enum

Physical Address Space of the accessed memory.

PASreq enum

Physical Address Space of the requested lookup.

REGIME_EL enum

Entry matches in this translation regime.

REQUESTER enum

The requester of this table walk.

SIDE enum

Inst / Data.

STAGE unsigned int

Translation stage.

MMU_D128_TTB_WRITE

This event is triggered by writes caused by a D128 translation table walk. Fields:

ABORTED bool

Was the walk successful.

DATA_HIGH unsigned int

The data upper bits written.

DATA_LOW unsigned int

The data lower bits written.

IPA unsigned int

For stage 1, the IPA of the write.

IS_HACDBS_ACCESS bool

If the access was due to HACDBS_CLEAN.

LEVEL unsigned int

Translation table level.

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

OUTADDR unsigned int

The output physical address.

PADDR unsigned int

The physical address of the write.

PASmem enum

Physical Address Space of the accessed memory.

PASreq enum

Physical Address Space of the requested lookup.

REGIME_EL enum

Entry matches in this translation regime.

SIDE enum

Inst / Data.

STAGE unsigned int

Translation stage.

MMU_TRANS

Address translation information. Fields:

ASID unsigned int

Address space identifier.

CORE_NUM unsigned int

Core number in a multi processor.

Hyp bool

Entry matches in Hyp state only.

INNERCACHE_RA bool

Is the inner cache allocate on read.

INNERCACHE_TRANSIENT bool

Is the inner write-through transient.

INNERCACHE_TYPE enum

Inner Caching scheme (NC/MB/WA).

INNERCACHE_WA bool

Is the inner cache allocate on write.

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MEMTYPE enum

Memory type.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC enum

Is secure side supposed to access secure or nonsecure memory.

OUTERCACHE_RA bool

Is the outer cache allocate on read.

OUTERCACHE_TRANSIENT bool

Is the outer write-through transient.

OUTERCACHE_TYPE enum

Outer Caching scheme (NC/MB/WA).

OUTERCACHE_WA bool

Is the outer cache allocate on write.

PADDR unsigned int

Physical address of the access.

PAGESIZE unsigned int

Page size as $\log_2(\text{size})$.

PAS enum

PAS supposed to be accessed is secure or nonsecure memory or Root/Realm if RME is implemented.

SH enum

Shareability.

SIDE enum

Inst / Data.

STAGE1_PERM unsigned int

Stage 1 permissions, [5:3] Privileged access (XWR) [2:0] User access (XWR).

STAGE2_PERM unsigned int

Stage 2 permission mask [5:3] Privileged access (XWR) [2:0] User access (XWR).

VADDR unsigned int

Virtual address of the access.

VMID unsigned int

Virtual machine identifier.

XS bool

XS Attribute.

nG enum

Flag indicating whether ASID will be matched.

MMU_TTB_READ

This event is triggered by reads caused by a translation table walk. Fields:

ABORTED bool

Was the walk successful.

DATA unsigned int

The data read.

DESC_KIND enum

The encoding scheme of the descriptors used for this walk.

IPA unsigned int

For stage 1, the IPA of the read.

LEVEL unsigned int

Translation table level.

LPAE bool

Is this for an LPAE translation (*DEPRECATED*, use DESC_KIND).

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSmem bool

Non secure memory access.

NSreq bool

Non secure request.

OUTADDR **unsigned int**

The output physical address.

PADDR **unsigned int**

The physical address of the read.

PASmem **enum**

Physical Address Space of the accessed memory.

PASreq **enum**

Physical Address Space of the requested lookup.

REGIME_EL **enum**

Entry matches in this translation regime.

REQUESTER **enum**

The requester of this table walk.

SIDE **enum**

Inst / Data.

STAGE **unsigned int**

Translation stage.

MMU_TTB_WRITE

This event is triggered by writes caused by a translation table walk. Fields:

ABORTED **bool**

Was the walk successful.

DATA **unsigned int**

The data written.

DESC_KIND **enum**

The encoding scheme of the descriptors used for this walk.

IPA **unsigned int**

For stage 1, the IPA of the write.

IS_HACDBS_ACCESS **bool**

If the access was due to HACDBS_CLEAN.

LEVEL **unsigned int**

Translation table level.

LPAAE **bool**

Is this for an LPAAE translation (*DEPRECATED*, use DESC_KIND).

MECID **unsigned int**

Memory Encryption Context(MEC) ID, if applicable.

MPAM_PARTID **unsigned int**

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSmem bool

Non secure memory access.

NSreq bool

Non secure request.

OUTADDR unsigned int

The output physical address.

PADDR unsigned int

The physical address of the write.

PASmem enum

Physical Address Space of the accessed memory.

PASreq enum

Physical Address Space of the requested lookup.

REGIME_EL enum

Entry matches in this translation regime.

SIDE enum

Inst / Data.

STAGE unsigned int

Translation stage.

MODE_CHANGE

Mode change. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

MODE enum

The new mode.

NON_SECURE enum

The core's new non-secure bit.

OLD_MODE enum

The old mode.

SECURITY_STATE enum

The new security state.

NOT_TAG_CHECKED_INST

MTE Tag not checked for current instruction.

PERIODIC

Called for every quantum. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The instruction count of this CPU.

PC unsigned int

The address of the next instruction to be executed on this CPU.

PMU_COUNTER_OVERFLOW

PMU counter overflow. Fields:

COUNTER_GROUP enum

Counter group.

COUNTER_RANGE enum

Counter range.

EVENT_ID unsigned int

Event Identifier, field is valid only for Event counter group.

INDEX unsigned int

Counter index (as selected by PMSELR), field is valid only for Event/Cycle counter groups.

INTERRUPT bool

Is interrupt enabled on overflow for this counter.

POINTER_AUTH

Called when execution hits a pointer authentication check. Fields:

AUTH_STATE bool

Authentication result.

KEY_USED enum

Pointer authentication key.

POTINDEX_UPDATE

Value of POT index for current pc changed. Fields:

NEW_POTINDEX unsigned int

New POTIndex.

OLD_POTINDEX unsigned int

Old POTIndex.

PREFETCH_MEMORY64

Prefetch from PRFM or PRFUM instructions. Fields:

PRFOP enum

Prefetch hint.

VADDR unsigned int

Virtual address of the location that should be prefetched.

PRELOAD64_REQUEST

Preload requests from PRFM or PRFUM instructions. Fields:

PRFOP enum

Prefetch hint.

VADDR unsigned int

Virtual address of the location that should be prefetched.

PRELOAD_DATA

Data preload from PLD instruction. Fields:

VADDR unsigned int

Virtual address of the data that should be preloaded.

PRELOAD_INST

Instruction preload from PLI instruction. Fields:

VADDR unsigned int

Virtual address of the instruction that should be preloaded.

PRE_CORE_LOAD

Trace just before a core load. Fields:

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

PRE_CORE_STORE

Trace just before a core store. Fields:

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

PRE_TTB_READ

This event is triggered before reads caused by a translation table walk. Fields:

ADDR unsigned int

The physical address of the read.

PSTATE

PSTATE fields update. Fields:

ALLINT unsigned int

All interrupt mask bit.

BTYPE unsigned int

Branch target identification bit.

CORE_NUM unsigned int

Core number in a multi processor.

DAIF unsigned int

Exception mask bits.

DIT unsigned int

Data Independent Timing bit.

EL unsigned int

Current Exception level field.

EXLOCK unsigned int

Exception return state lock bit.

IL unsigned int

Illegal Execution state bit.

NZCV unsigned int

Condition flags.

PACM unsigned int

PACM bit.

PAN unsigned int

Privileged Access Never state bit.

PM unsigned int

PMU exception mask bit.

PPEND unsigned int

PMU exception pending bit.

SM unsigned int

StreamingMode status bit.

SP unsigned int

Stack pointer register selection bit.

SS unsigned int

Software Step bit.

SSBS unsigned int

Speculative Store Bypass Safe bit.

TCO unsigned int

Tag Check Override bit.

TOUCHED unsigned int

Mask of the fields that have been touched since this trace has last been emitted. Note that touched does not necessarily mean the value changed. The index of the field is the bit index in the mask plus the index of this field plus one.

UAO unsigned int

User Access Override bit.

UINJ unsigned int

Undefined Instruction exceptions bit.

ZA unsigned int

ZA storage and ZT0 register enable bit.

nRW unsigned int

Current Execution state bit.

PSTATE_PACTIVE_OPMODE_TRANSITION

Trace pactive and Opmode mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Operating mode after the transition.

OLD_STATE enum

Operating mode before the transition.

PSTATE_PACTIVE_OPMODE_TRANSITION_DENIED

Trace the denied pactive and Opmode mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Operating mode after the transition.

OLD_STATE enum

Operating mode before the transition.

REASON string

Reason for the transition being denied.

PSTATE_PACTIVE_POWER_TRANSITION

Trace pactive and power mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Power mode after the transition.

OLD_STATE enum

Power mode before the transition.

PSTATE_PACTIVE_POWER_TRANSITION_DENIED

Trace the denied pactive and power mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Power mode after the transition.

OLD_STATE enum

Power mode before the transition.

REASON string

Reason for the transition being denied.

PSTATE_PACTIVE_THREAD_OPMODE_TRANSITION

Trace pactive and Opmode mode transitions for threads on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Operating mode after the transition.

OLD_STATE enum

Operating mode before the transition.

QUANTUM_START

Called at the start of every quantum.

RANGE_PRELOAD_REQUEST

Preload requests from RPRFM. Fields:

COUNT unsigned int

Number of blocks of data to access.

LENGTH unsigned int

Number of contiguous bytes to be accessed in each block.

OP unsigned int

Range prefetch operation .

REUSE unsigned int

Maximum number of bytes to be accessed before next RPRFM .

STRIDE signed int

Stride to advance after accessing LENGTH bytes.

VADDR unsigned int

Base virtual address of range prefetch request.

RUN_STATE

Run state transition. Fields:

INST_COUNT **unsigned int**

Ticks count at point of transition.

NEW **enum**

New run state.

OLD **enum**

Old run state.

SEMIHOSTING_CALL

Call of a semihost function occurred. Fields:

PC **unsigned int**

The program counter after the semihosting call.

SEMIHOSTING_PRECALL

About to call semihost. Fields:

EL2 **bool**

Use EL2 translation regime.

NS **bool**

Is Non-Secure.

REG_WIDTH **unsigned int**

The current register width in bytes.

SIGNAL

External signal state change. Fields:

SIGNAL **enum**

Signal that changed.

STATE **bool**

Signal asserted state.

SOFTWARE_STEP

Return the Debug Software Step state. Fields:

STATE **enum**

Software Step state.

SPSR

SPSR change. Fields:

CORE_NUM **unsigned int**

Core number in a multi processor.

MODE **enum**

Which of the banked SPSR registers is written.

OLD_VALUE unsigned int

The old SPSR value.

VALUE unsigned int

The new SPSR value.

START_COMPILE

Compilation started. Fields:

VADDR unsigned int

Instruction where compilation begins.

STG_MULTIPLE_MICRO_OPS_ADDRESS_RANGE

Address range for MTE stores with multiple uops. Fields:

ADDR_END unsigned int

Final address.

ADDR_START unsigned int

Starting address.

STORE_MULTIPLE_REGS_ADDRESS_RANGE

Address range for stores with multiple registers. Fields:

ADDR_END unsigned int

Final address.

ADDR_START unsigned int

Starting address.

SUBOPTIMAL_LDST_RETIRED

suboptimal load/store. Fields:

ACCESS_SIZE unsigned int

Log2 of the access width in bytes used for alignment checking (1: half-word, 2: word ...).

OPTIMAL_ALIGN_SELECTOR unsigned int

Selector to decide the optimal alignment size (1: access width, 2: 4-bytes, 3: 8-bytes, ..., 12: 4-kbytes).

VADDR unsigned int

The virtual address of the access.

SVE_INST_SPEC

SVE operations speculatively executed.

SVE_LD_RETIRED

SVE load instructions architecturally executed.

SVE_LOADS

SVE load accesses. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ACQREL enum

Is this an acquire/release.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DATA unsigned int

The data read or written.

ELEMENT_SIZE unsigned int

Width of each element.

ENDIANNESS enum

Byte ordering of this access.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

LOCK enum

Normal, exclusive or locked access.

MECID unsigned int

Memory Encryption Context ID, if applicable.

MECID2 unsigned int

Memory Encryption Context ID of the second page, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PADDR_TAG unsigned int

If tag checked, the physical address tag for the access.

PAGE_SIZE unsigned int

Page size as $\log_2(\text{size in bytes})$.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK, 2=Exclusive Failed, 3=Unsupported.

SIZE unsigned int

Width of the access in bytes. Only required if DATA is not traced.

TAG_CHECKED bool

Is this access tag checked.

TAG_CHECKED2 bool

Is the second page access tag checked.

TRANS bool

Is this a translated access.

VADDR unsigned int

The virtual address of the access.

SVE_MEM_PREFETCH

Scalable Vector Extension memory prefetch. Fields:

CACHE_LEVEL enum

The preferred cache level.

HINT enum

What the memory is expected to be used for.

SIZE unsigned int

The prefetch size in bytes.

TEMPORAL bool

Whether the data is temporal.

VADDR unsigned int

The virtual address of the prefetch.

SVE_REG_READ

Scalable Vector Extension register read. Fields:

ID enum

The register identifier.

SM bool

Whether the PE is in Streaming Mode.

VALUE unsigned int

The read data.

SVE_REG_UPDATE

Scalable Vector Extension register update. Fields:

ID enum

The register identifier.

SM bool

Whether the PE is in Streaming Mode.

VALUE unsigned int

The written data.

SVE_REG_WRITE

Scalable Vector Extension register write. Fields:

ID enum

The register identifier.

SM bool

Whether the PE is in Streaming Mode.

VALUE unsigned int

The written data.

SVE_STORES

SVE store accesses. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ACQREL enum

Is this an acquire/release.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DATA unsigned int

The data read or written.

ELEMENT_SIZE unsigned int

Width of each element.

ENDIANNESS enum

Byte ordering of this access.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

LOCK enum

Normal, exclusive or locked access.

MECID unsigned int

Memory Encryption Context ID, if applicable.

MECID2 unsigned int

Memory Encryption Context ID of the second page, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PADDR_TAG unsigned int

If tag checked, the physical address tag for the access.

PAGE_SIZE unsigned int

Page size as $\log_2(\text{size in bytes})$.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK, 2=Exclusive Failed, 3=Unsupported.

SIZE unsigned int

Width of the access in bytes. Only required if DATA is not traced.

TAG_CHECKED bool

Is this access tag checked.

TAG_CHECKED2 bool

Is the second page access tag checked.

TRANS bool

Is this a translated access.

VADDR unsigned int

The virtual address of the access.

SVE_ST_RETIRED

SVE store instructions architecturally executed.

SVE_UNALIGNED_LDST_RETIRED

SVE load or store instructions architecturally executed that would cause alignment fault if A bit is set.

SVE_Z_PRED_MEM_ACCESS_INFO

Information about SVE/SME predicated accesses to Z registers (non Gather/Scatter). Fields:

MEM_ELEM_SIZE unsigned int

Width of elements accessed in memory (in bytes).

RnW bool

Whether this is a Read not a Write.

VA unsigned int

Virtual base address computed for accesses.

SYNC

Called for every synchronization, that is every quantum break. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The instruction count of this CPU.

LOCAL_QUANTUM unsigned int

The local quantum of this CPU.

LOCAL_TIME unsigned int

The local time of this CPU.

SYNC_REASON enum

Reason for sync.

SYSCALL

System call instruction executed. Fields:

IMM unsigned int

Immediate value of the system call instruction.

TYPE enum

System call type.

VADDR unsigned int

Instruction that caused the system call.

SYSREG_READ128

System Coprocessor 128 bit register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

REGNUM enum

Internal register number.

REGNUM_PHYS enum

Internal register number physically accessed (e.g. in presense of VHE redirection).

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_READ64

System Coprocessor register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REGNUM enum

Internal register number.

REGNUM_OPERAND enum

Register number corresponding to the instruction operand.

REGNUM_PHYS enum

Internal register number physically accessed (e.g. in presense of VHE redirection).

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_UPDATE128

Triggers when the system updates a register. Fields:

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG enum

Register number.

UNKNOWN unsigned int

Bits of the register which became unknown.

VALUE unsigned int

Value written to the register.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_UPDATE32

Triggers when the system updates a register. Fields:

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG enum

Register number.

UNKNOWN unsigned int

Bits of the register which became unknown.

VALUE unsigned int

Value written to the register.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_UPDATE64

Triggers when the system updates a register. Fields:

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG enum

Register number.

UNKNOWN unsigned int

Bits of the register which became unknown.

VALUE unsigned int

Value written to the register.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_WRITE128

System Coprocessor 128 bit register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

REGNUM enum

Internal register number.

REGNUM_PHYS enum

Internal register number physically accessed (e.g. in presense of VHE redirection).

UNDEF bool

The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int
opcode 2.

SYSREG_WRITE64

System Coprocessor register write. Fields:

CORE_NUM unsigned int
Core number in a multi processor.

CRm unsigned int
CRm.

CRn unsigned int
CRn.

NS bool
Secure or nonsecure banked register is accessed.

REGNUM enum
Internal register number.

REGNUM_OPERAND enum
Register number corresponding to the instruction operand.

REGNUM_PHYS enum
Internal register number physically accessed (e.g. in presense of VHE redirection).

UNDEF bool
The register accessed is undefined.

UPDATED_VALUE unsigned int
Updated value of the register now it has been written.

VALUE unsigned int
The new value written.

opc unsigned int
opcode 1.

opc0 unsigned int
opcode 0.

opc2 unsigned int
opcode 2.

TRBU_BUFFER_WRAP

DISPLAY Trace buffer current write pointer wrapped.

TRBU_TRIGGER_EVENT

DISPLAY Trace buffer Trigger Event.

TRBU_WRITE

DISPLAY PETU trace data has been written to the TRBU memory buffer at %{BUFFER_ADDRESS} with size %{SIZE} and data %{DATA}. Fields:

BUFFER_ADDRESS unsigned int

Virtual Address of start of buffer write.

DATA unsigned int

Data written to the buffer.

SIZE unsigned int

Size of sample written to buffer, in bytes.

UNALIGNED_LDST_RETIRED

Processor unaligned load/store. Fields:

ACCESS_SIZE unsigned int

Log2 of the access width in bytes used for alignment checking (1=>half-word, 2->word ...).

NUMBER_OF_BEATS unsigned int

Number of accesses (beats) in this burst (Total size in bytes is ACCESS_SIZE * NUMBER_OF_BEATS).

VADDR unsigned int

The virtual address of the access.

VFP_D_REGS

VFP/NEON D 64 bit register write. Fields:

ALIASING enum

Type of register aliasing used.

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a write to the corresponding bit in the register. For example, 0x1 would indicate a write to VALUE[0].

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

VFP_D_REGS_READ

VFP/NEON D 64 bit register read. Fields:

ID unsigned int

The register number.

VALUE unsigned int

The current value of the register.

VFP_Q_REGS

VFP/NEON Q 128 bit register write. Fields:

ALIASING enum

Type of register aliasing used.

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a write to the corresponding byte in the register. For example, 0x1 would indicate a write to VALUE[7:0].

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

VFP_Q_REGS_READ

VFP/NEON Q 128 bit register read. Fields:

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a read to the corresponding byte in the register. For example, 0x1 would indicate a read to VALUE[7:0].

VALUE unsigned int

The current value of the register.

VFP_SYS_REGS

Writes to the VFP/NEON units system registers. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ID enum

Which VFP system register is written.

MASK unsigned int

The mask indicating updated cumulative bits.

OLD_VALUE unsigned int

The register's old value overwritten.

VALUE unsigned int

The new value written to the VFP system register.

VFP_S_REGS

VFP/NEON S 32 bit register write. Fields:

ALIASING enum

Type of register aliasing used.

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a write to the corresponding bit in the register. For example, 0x1 would indicate a write to VALUE[0].

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

VFP_S_REGS_READ

VFP/NEON S 32 bit register read. Fields:

ID unsigned int

The register number.

VALUE unsigned int

The current value of the register.

VIRTUAL_TAG_ALLOCATED

Virtual tag information when allocated. Fields:

TAG_VADDR unsigned int

Virtual tag virtual address.

VADDR unsigned int

Data Virtual address.

VIRTUAL_TAG unsigned int

Virtual tag value.

WAYPOINT

Signals end of basic block execution. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ISSET enum

Origin instruction set.

IS_COND bool

Indicates if this is a conditional waypoint.

PC unsigned int

Origin address (or 0 if unavailable).

TAKEN bool

Indicates if this waypoint was taken.

TARGET unsigned int

Destination address.

TARGET_ISET enum

Destination instruction set.

WFE_END

WFE ended. Fields:

INST_COUNT unsigned int

Ticks count when leaving WFE.

WFE_EVENT_REGISTER

WFE event register status: set/clear, reason. Fields:

INST_COUNT unsigned int

Ticks count.

REASON enum

Reason for set/clear. Only REASON==0 (Cleared by WFE) clears the bit, all other reasons set it.

WFE_IGNORED

WFE ignored. Fields:

EVENT bool

This WFE was ignored because the event register was set.

INST_COUNT unsigned int

Ticks count when ignoring WFE.

TRAPPED bool

This WFE was trapped.

WFE_START

WFE entered. Fields:

INST_COUNT unsigned int

Ticks count when entering WFE.

WFI_END

WFI ended. Fields:

INST_COUNT unsigned int

Ticks count when leaving WFI.

WFI_IGNORED

WFI ignored. Fields:

DISABLED bool

This WFI was ignored because WFI is disabled.

INST_COUNT unsigned int

Ticks count when ignoring WFI.

REASON enum

specifies reason why WFI trace was ignored.

TRAPPED bool

This WFI was trapped.

WFI_START

WFI entered. Fields:

INST_COUNT unsigned int

Ticks count when entering WFI.

WFI_WAKEUP

WFI wakeup. Fields:

INST_COUNT unsigned int

Ticks count when WFI wakeup occurred.

REASON enum

Reason for wakeup.

2.30 ARM_Cortex-A73

This section describes the trace sources.

AA64_ASE_SVE_REGS

Modification of AArch64 Advanced SIMD or SVE Vector Registers (via SIMD or Scalar operations), SVE Predicate Registers and ZT0. Fields:

ID enum

The register identifier.

MASK unsigned int

Bitmask of the register to signal the modified bits in the VALUE field.

SM bool

Whether the PE is in Streaming Mode.

VALUE unsigned int

Value of AArch64 vector register.

ASYNC_MEMORY_FAULT

Context ID Register write. Fields:

FAULT unsigned int

Fault status present in ESR.ISS encoding for Serror in ARCH64 and present in DFSR encoding for external abort format in ARCH32.

PADDR unsigned int

Physical Address (or 0 if unavailable).

VADDR signed int

Virtual Address (or 0 if unavailable).

ATOMIC_END_ACCESS

Bus trace access for atomics. Fields:

ACCESS_FAIL bool

Memory access failed.

ADDR unsigned int

The virtual address of the access.

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

COMPARE_VALUE unsigned int

Compare value for CAS.

LOAD_VALUE unsigned int

Loaded values.

NSDESC unsigned int

The security state of the access.

OPERAND_VALUE unsigned int

Operation's operand.

OPERATION enum

Operation type.

PADDR unsigned int

The physical address of the access.

PRIV bool

Is this a privileged access?.

ATOMIC_START_ACCESS

Bus trace access for atomics. Fields:

ADDR unsigned int

The virtual address of the access.

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

COMPARE_VALUE unsigned int

Compare value for CAS.

NSDESC unsigned int

The security state of the access.

OPERAND_VALUE unsigned int

Operation's operand.

OPERATION enum

Operation type.

PADDR unsigned int

The physical address of the access.

PRIV bool

Is this a privileged access?.

ArchMsg.Error.exit_code

DISPLAY exit code: %{EXITCODE} component kind: %{KIND} reason: %{REASON}. Fields:

EXITCODE signed int

Exit code to be returned.

KIND string

Component kind that invoked the exit code trace.

REASON string

Reason why the exit code trace was invoked (optional).

ArchMsg.Warning.IgnoredWarmResetRequest

Event fires when software attempts to request warm reset through write to deprecated DBGPRCR.CWRR.

ArchMsg.Warning.ReportedRevisionRegisterRead

DISPLAY Reported revision and patch do not match that of the cpu. Fields:

ORIGINAL_VALUE unsigned int

The value the register would have based on the modelled revision.

REGISTER string

The name of the register that has ben read.

REPORTED_VALUE unsigned int

The value the register has been overridden to.

ArchMsg.Warning.Unpredictably Indexed PM Event Register

TAGS **UNPREDICTABLE** DISPLAY Event %{IsDirect:(Selector value|Register)} %{SEL} is >= %{N}.

Fields:

IsDirect unsigned int

Direct Access rather than indirect.

N unsigned int

Number of Accessible Registers.

SEL unsigned int

Selector.

ArchMsg.Warning.cache_contents_unknown

A cache has been activated without being invalidated and may contain **UNKNOWN** content. DISPLAY %{SIDE:(D|I)}-Cache was enabled in %{REGIME:(SECURE|NON_SECURE|HYP)} regime but was not invalidated since power-on: cache lines could contain **UNKNOWN** content %{SIDE:(Note: D-side caches must invalidate SECURE lines even if the cache is only used by NON_SECURE code in order to ensure that unexpected dirty lines tagged SECURE are not naturally evicted)}. Fields:

REGIME unsigned int

Current translation regime.

SIDE unsigned int

0 d-side cache or unified cache, 1 i-side.

ArchMsg.Warning.dcimvac_matches_watchpoint

TAGS **UNPREDICTABLE** DISPLAY Watchpoints matching an AArch32 DCIMVA are implementation defined.

ArchMsg.Warning.decode_fieldmismatch

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with different fields %{FIELD1} and %{FIELD2}. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_frachitsnegative

TAGS **UNPREDICTABLE** DISPLAY VCVT Instruction is unpredictable with a negative number of fraction bits.

ArchMsg.Warning.decode_initblock

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable in an IT block.

ArchMsg.Warning.decode_initblocknotlast

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable in an IT block when not the last.

ArchMsg.Warning.decode_invalid_condition

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable when the condition is not AL.

ArchMsg.Warning.decode_invalidfieldcombination

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with the values in %{FIELD1} and %{FIELD2}. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_invalidvalue

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with the value in %{FIELD}. Fields:

FIELD enum

Field name.

ArchMsg.Warning.decode_mem_hint_unallocated

TAGS **UNPREDICTABLE** DISPLAY There is no memory hint allocated to this bit pattern.

ArchMsg.Warning.decode_registermatch

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with %{FIELD1} and %{FIELD2} as the same register. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_registermismatch

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with %{FIELD1} and %{FIELD2} as different registers. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_registeroutofrange

TAGS **UNPREDICTABLE** DISPLAY Floating-point instruction is unpredictable indexing beyond the end of the register bank.

ArchMsg.Warning.decode_sbzsbo

TAGS **UNPREDICTABLE** DISPLAY Reserved bits in the instruction are not canonical.

ArchMsg.Warning.decode_transactiontoobig

TAGS UNPREDICTABLE DISPLAY Attempt to perform a memory transaction of more than 128 bytes.

ArchMsg.Warning.decode_unpred_other

TAGS UNPREDICTABLE DISPLAY Instruction is unpredictable.

ArchMsg.Warning.decode_unpreduseofpc

TAGS UNPREDICTABLE DISPLAY Instruction is unpredictable when %{FIELD} is PC. Fields:

FIELD enum

Field name.

ArchMsg.Warning.decode_unpreduseofr13

TAGS UNPREDICTABLE DISPLAY Instruction is unpredictable when %{FIELD} is R13. Fields:

FIELD enum

Field name.

ArchMsg.Warning.decode_unpreduseofr13inreglist

TAGS UNPREDICTABLE DISPLAY Instruction is unpredictable with R13 in register-list.

ArchMsg.Warning.decode_useofd16orhigher

TAGS UNPREDICTABLE DISPLAY Attempt to use register D16 or higher where it is not allowed.

ArchMsg.Warning.decode_writebackandbaseinlist

TAGS UNPREDICTABLE DISPLAY Instruction is unpredictable with write back when the base register is in the transfer list.

ArchMsg.Warning.decode_zeroregistersinlist

TAGS UNPREDICTABLE DISPLAY Instruction is unpredictable when no registers are to be transferred.

ArchMsg.Warning.recursive_branch

An instruction is performing a branch that targets the same instruction. If this was intended then a WFI might be more effective DISPLAY recursive branch detected at PC=%{PC}. PRIMARY KEY PC. Fields:

PC unsigned int

Address of instruction causing the branch.

ArchMsg.Warning.recursive_exception

An instruction has generated an exception that targets the same instruction. DISPLAY recursive %{TYPE} exception detected at PC=%{PC} (CPSR=%{CPSR}). PRIMARY KEY PC. Fields:

CPSR unsigned int

Processor state.

PC unsigned int

Address of instruction causing the exception.

TYPE enum

Type of exception.

ArchMsg.Warning.reentrant_vector_catch

TAGS **UNPREDICTABLE** DISPLAY Vector catch debug event on Prefetch or Data abort vector %{ADDR}. Fields:

ADDR unsigned int

vector address.

ArchMsg.Warning.reserved_it_state

TAGS **UNPREDICTABLE** DISPLAY Execution with reserved IT state %{ITSTATE}. Fields:

ITSTATE unsigned int

reserved IT state.

ArchMsg.Warning.tlb_contents_unknown

Translation has been enabled without invalidating the TLB. DISPLAY Translation was enabled in the %{PAS:(Secure|Non-secure|Root|Realm)} EL%{EL:d} translation regime (vmid=%{VMID:x}) without invalidating %{INVALIDITY:(|d-side |i-side |)}entries since power-on: %{INVALIDITY:(|D|I|)}TLB could contain **UNKNOWN** content. Fields:

EL unsigned int

Exception level of translation regime.

INVALIDITY unsigned int

Which TLB may be invalid (if TLBs are separate).

NON_SECURE bool

Whether the translation regime is non-secure.

PAS unsigned int

Physical address space of translation regime.

VMID unsigned int

Current VMID, if applicable.

ArchMsg.Warning.unknown_DLR_DSPSR

Exiting debug state with unknown DLR or DSPSR.

ArchMsg.Warning.unknown_ELR_SPSR

Returning from debug exception with unknown ELR or SPSR.

ArchMsg.Warning.unpred_FAR_on_LRCPC3_non_atomic_mem_access

TAGS **UNPREDICTABLE** Unpredictable FAR/FSTATUS on LRCPC3 pair instructions resulting in two separate memory accessesDISPLAY Unpredictable FAR on LRCPC3 non atomic memory access at %{ADDR1} with FSTATUS %{FSTATUS1} or %{ADDR2} with FSTATUS %{FSTATUS2}. Fields:

ADDR1 unsigned int

Possible FAR address 1.

ADDR2 unsigned int

Possible FAR address 2.

FSTATUS1 unsigned int

Possible fault status value 1.

FSTATUS2 unsigned int

Possible fault status value 2.

ArchMsg.Warning.unpred_async_tag_check_fail_on_sve_speculative_load_do_update_ffr

Unpredictable set to FALSE of the FFR element associated to SVE speculative load with an async tag check fail TAGS **UNPREDICTABLE**.

ArchMsg.Warning.unpredictable_a32_breakpoint

TAGS **UNPREDICTABLE** DISPLAY Unpredictable breakpoint on a A32 instruction at %{ADDR1} Offset %{OFFSET} BAS %{BAS} for a %{IS_ADDRESS_MISMATCH_BREAKPOINT:(Mismatch|Match)} Breakpoint number %{N:d} Breakpoint address %{ADDR2}. Fields:

ADDR1 unsigned int

Address.

ADDR2 unsigned int

Breakpoint address.

BAS unsigned int

BAS field.

IS_ADDRESS_MISMATCH_BREAKPOINT bool

Is a Mismatch breakpoint, not a Match breakpoint.

N unsigned int

Breakpoint number.

OFFSET unsigned int

Always has a value of zero.

ArchMsg.Warning.unpredictable_banked_register_access

TAGS **UNPREDICTABLE** DISPLAY Access to banked register (R=%{R:d} SYSm=%{SYSM:2x}) is unpredictable. Fields:

R unsigned int

R bit.

SYSM unsigned int

SYSm.

ArchMsg.Warning.unpredictable_ccfail_undef

TAGS **UNPREDICTABLE** DISPLAY An instruction %{OPCODE} at VA=%{VADDR} failed its condition codes check but would otherwise have caused a non-data-dependent trap or undefined exception. Behaviour is implementation-defined choice of exception or nop. Fields:

OPCODE unsigned int

opcode of the instruction.

VADDR unsigned int

Virtual address of the instruction being executed.

ArchMsg.Warning.unpredictable_ici_bits

TAGS **UNPREDICTABLE** DISPLAY ICI bits do not apply or are architecturally unpredictable for instruction.

ArchMsg.Warning.unpredictable_t32_breakpoint

TAGS **UNPREDICTABLE** DISPLAY Unpredictable breakpoint on a T32 instruction at %{ADDR1} Offset %{OFFSET} BAS %{BAS} for a %{IS_ADDRESS_MISMATCH_BREAKPOINT:(Mismatch|Match)} Breakpoint number %{N:d} Breakpoint address %{ADDR2}. Fields:

ADDR1 unsigned int

Address.

ADDR2 unsigned int

Breakpoint address.

BAS unsigned int

BAS field.

IS_ADDRESS_MISMATCH_BREAKPOINT bool

Is a Mismatch breakpoint, not a Match breakpoint.

N unsigned int

Breakpoint number.

OFFSET unsigned int

Indicates which halfword of a 32-bit Thumb instruction the breakpoint is set on. 0 for the first halfword or if the instruction is 16 bits wide, or 1 for the second halfword.

ArchMsg.Warning.unpredictable_t32_vector_catch

TAGS **UNPREDICTABLE** DISPLAY Unpredictable Vector catch exception on a T32 instruction at %{ADDR} Offset %{OFFSET}. Fields:

ADDR unsigned int

Address.

OFFSET unsigned int

Indicates which halfword of a 32-bit Thumb instruction the vector catch exception is set on. 0 for the first halfword or if the instruction is 16 bits wide, or 1 for the second halfword.

ArchMsg.Warning.unpredictable_watchpoint_far

TAGS **UNPREDICTABLE** DISPLAY The value for FAR/EDWAR for hitting this watchpoint may be anywhere in the range %{LOWER_BOUND} to %{UPPER_BOUND}. Fields:

LOWER_BOUND unsigned int

the lowest address accessed by the instruction that triggered the watchpoint.

UPPER_BOUND unsigned int

the highest watchpointed address accessed by that instruction.

ArchMsg.Warning.unpredictable_watchpoint_far_isscmp

DISPLAY The value for FAR/EDWAR for hitting this watchpoint may be anywhere in the range %{LOWER_BOUND} to %{UPPER_BOUND}. Fields:

LOWER_BOUND unsigned int

the lowest address accessed by the instruction that triggered the watchpoint.

UPPER_BOUND unsigned int

the highest watchpointed address accessed by that instruction.

ArchMsg.Warning.warning_AdvSIMDExpandImmUnexpectedZero

TAGS **UNPREDICTABLE** DISPLAY AdvSIMDExpandImm may treat this immediate value as **UNPREDICTABLE**.

ArchMsg.Warning.warning_ConditionalSMC

TAGS **UNPREDICTABLE** DISPLAY SMC instruction has **UNPREDICTABLE** effects when conditional and when combined with traps.

ArchMsg.Warning.warning_TLBID_out_of_range

TLBID out of range in TLBI according to the configuration of NIS, NVIS, NOS and NVOS. Fields:

NIS unsigned int

NIS.

NOS unsigned int

NOS.

NVIS unsigned int

NVIS.

NVOS unsigned int

NVOS.

TLBID unsigned int

The TLBID value used in the TLBI instruction.

ArchMsg.Warning.warning_TTL_or_TG_mismatches_current_translation_granule

TTL or TG mismatches current translation granule in TLBI VA/RVA/IPA/RIPA. Fields:

CURRENT_TG unsigned int

Current TG as indicated by TCR_ELx or VTCR_EL2.

TLBI_TG unsigned int

TG (or TTL[3:2]) used in TLBI instruction.

ArchMsg.Warning.warning_access_crosses_page_boundary

TAGS **UNPREDICTABLE** DISPLAY Access crosses page boundary.

ArchMsg.Warning.warning_access_crosses_page_boundary_has_fault

TAGS **UNPREDICTABLE** DISPLAY Access that crosses page boundary has fault(s).

ArchMsg.Warning.warning_access_crosses_page_boundary_has_fault_on_both_pages

TAGS **UNPREDICTABLE** DISPLAY Access that crosses page boundary has a fault on each page.

Fields:

ADDR1 unsigned int

FAR address on first page.

ADDR2 unsigned int

FAR address on second page.

FSTATUS1 unsigned int

Fault status on first page.

FSTATUS2 unsigned int

Fault status on second page.

ArchMsg.Warning.warning_access_crosses_page_boundary_spanning_different_memory

TAGS **UNPREDICTABLE** DISPLAY Access at address %{ADDR} crosses page from MEM_TYPE %{MEMTYPE_PAGE1} to %{MEMTYPE_PAGE2}. Fields:

ADDR unsigned int

address of access crossing page.

MEMTYPE_PAGE1 signed int

First Page's memory type (0-Normal, 1-Non-Normal).

MEMTYPE_PAGE2 signed int

Second Page's memory type (0-Normal, 1-Non-Normal).

ArchMsg.Warning.warning_access_wraps_around_memory

TAGS **UNPREDICTABLE** DISPLAY Access wraps around memory in %{WIDTH:d}bit mode.

Fields:

WIDTH unsigned int

Address width.

ArchMsg.Warning.warning_bcr_linking_status

TAGS **UNPREDICTABLE** DISPLAY DBGBCR.{BT2,BT} combination gives unpredictable behaviour.

ArchMsg.Warning.warning_bcr_mask_reserved

TAGS **UNPREDICTABLE** DISPLAY DBGBCR_ELn.MASK is set to a reserved value.

ArchMsg.Warning.warning_bvr_masked_bit_not_zero

TAGS **UNPREDICTABLE** DISPLAY DBGBVRn_EL1.VA contains a bit masked by DBGBCRn_EL1.MASK that is not zero.

ArchMsg.Warning.warning_ccsidr_unimplemented_level

TAGS **UNPREDICTABLE** DISPLAY CCSIDR read while CSSELR %{CSSELR:x} points to an unimplemented cache level. Fields:

CSSELR unsigned int

current effective value of CSSELR.

ArchMsg.Warning.warning_change_to_ns_when_tge_set

TAGS **UNPREDICTABLE** DISPLAY Attempting to change NS to 1 when HCR.TGE = 1.

ArchMsg.Warning.warning_contiguous_bit_check_abort

TAGS **UNPREDICTABLE** DISPLAY An abort occurred whilst attempting to check TLB entry is contiguous at %{ENTRY_ADDR}. Fields:

ENTRY_ADDR unsigned int

address of conflicting TLB entry.

ArchMsg.Warning.warning_contiguous_bit_error

TAGS **UNPREDICTABLE** DISPLAY TLB entry at %{ENTRY_ADDR} was not contiguous with entry at %{CONTIG_ADDR} Contents are %{ENTRY_DATA} but expected %{CONTIG_DATA}. Fields:

CONTIG_ADDR unsigned int

address of the TLB first read, with which this entry is expected to be contiguous.

CONTIG_DATA unsigned int

expected contents based on the entry at CONTIG_ADDR.

ENTRY_ADDR unsigned int

address of conflicting TLB entry.

ENTRY_DATA unsigned int

contents of conflicting TLB entry.

ArchMsg.Warning.warning_contiguous_bit_error_D128

TAGS **UNPREDICTABLE** DISPLAY TLB entry at %{ENTRY_ADDR} was not contiguous with entry at %{CONTIG_ADDR} Contents are %{ENTRY_DATA_HI} %{ENTRY_DATA_LO} but expected %{CONTIG_DATA}. Fields:

CONTIG_ADDR unsigned int

address of the TLB first read, with which this entry is expected to be contiguous.

CONTIG_DATA unsigned int

expected contents based on the entry at CONTIG_ADDR.

ENTRY_ADDR unsigned int

address of conflicting TLB entry.

ENTRY_DATA_HI unsigned int

Higher bit contents of conflicting TLB entry.

ENTRY_DATA_LO unsigned int

Lower bit contents of conflicting TLB entry.

ArchMsg.Warning.warning_cp10_cp11_mismatch

TAGS **UNPREDICTABLE** DISPLAY Unpredictable configuration of CP10 and CP11 controls at EL %{EL:d}. Fields:

EL unsigned int

EL owning the control.

ArchMsg.Warning.warning_cp10_cp11_reserved_value

TAGS **UNPREDICTABLE** DISPLAY Unpredictable configuration of CPACR CP10 ({CP10}) or CP11 ({CP11}). Fields:

CP10 unsigned int

CP10 access.

CP11 unsigned int

CP11 access.

ArchMsg.Warning.warning_csselr_level_out_of_range

TAGS **UNPREDICTABLE** DISPLAY CSSELR.Level out of range, written value {LEVEL} with only {IMPLEMENTED:d} levels implemented. Fields:

IMPLEMENTED unsigned int

number of cache levels implemented.

LEVEL unsigned int

written value of CSSELR.Level.

TYPE enum

The type of cache selected by CSSELR.InD and TnD.

ArchMsg.Warning.warning_debug_flow_control_bits_not_obeyed

TAGS **UNPREDICTABLE** DISPLAY Flow control bits for {REGISTER} were not obeyed, causing {ERROR}. Fields:

ERROR enum

type of error resulting.

REGISTER enum

register under consideration.

ArchMsg.Warning.warning_debug_register_access_during_reset

TAGS **UNPREDICTABLE** DISPLAY Unpredictable {IS_WRITE:(read|write)} access to debug register offset {OFFSET} during reset. Fields:

IS_WRITE unsigned int

Write Not Read.

OFFSET unsigned int

Register Offset.

ArchMsg.Warning.warning_decode_cps_inconsistent_fields

TAGS **UNPREDICTABLE** DISPLAY The mode setting fields in the CPS instruction are inconsistent.

ArchMsg.Warning.warning_decode_invalid_state

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable in current exception-level/security state.

ArchMsg.Warning.warning_default_cacheable_mmu_on

TAGS **UNPREDICTABLE** DISPLAY Default cacheable is enabled (HCR.DC = 1) while MMU is enabled.

ArchMsg.Warning.warning_default_cacheable_vmmu_off

TAGS **UNPREDICTABLE** DISPLAY Default cacheable is enabled (HCR.DC = 1) while HCR.VM = 0.

ArchMsg.Warning.warning_deprecated_wvr_bit_2_set

TAGS **UNPREDICTABLE** DISPLAY DBGWVR%{N:d}_EL1=%{WVR:h} has bit 2 set, deprecated in ARMv8. Fields:

N unsigned int

index.

WVR unsigned int

DBGWVR.

ArchMsg.Warning.warning_eret_while_software_step_active_pending

Missing ISB between setting MDSCR_EL1.SS and ERET TAGS **UNPREDICTABLE**.

ArchMsg.Warning.warning_exclusive_to_non_normal

TAGS **UNPREDICTABLE** DISPLAY Exclusive Access to Non-Normal Memory %{ADDR}. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_exclusive_to_non_writeback

TAGS **UNPREDICTABLE** DISPLAY Exclusive Access to Non-Writeback-Cacheable Memory %{ADDR}. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_execute_from_device_memory

TAGS **UNPREDICTABLE** DISPLAY An attempt was made to execute from Device memory.

ArchMsg.Warning.warning_illegal_cpsr_mode

TAGS **UNPREDICTABLE** DISPLAY Writing an illegal or unimplement mode (%{NEW_MODE:h}) to CPSR was unpredictable. Fields:

NEW_MODE unsigned int

New value of CPSR.M.

ArchMsg.Warning.warning_illegal_srs_mode

TAGS **UNPREDICTABLE** DISPLAY Illegal or **UNPREDICTABLE** mode (%{MODE:h}) used for SRS instruction. Fields:

MODE unsigned int

mode for Banked SP.

ArchMsg.Warning.warning_implementation_defined_read_debug_register_in_SCS

Read Access to Debug registers from the processor is IMPLEMENTATION DEFINED TAGS IMP_DEF. Fields:

PPB_OFFSET unsigned int
debug register offset.

ArchMsg.Warning.warning_implementation_defined_sequential_security_transitions_supported

The behavior of sequential instruction fetches that cross from non-secure to secure memory and contain an SG instruction is CONSTRAINED UNPREDICTABLE TAGS UNPREDICTABLE.

ArchMsg.Warning.warning_implementation_defined_stack_limit_check_supported

It is IMPLEMENTATION DEFINED whether stack pointer limit checking is performed for this instruction TAGS IMP_DEF. Fields:

ADDRESS unsigned int
Address of instruction.

OPCODE unsigned int
instruction opcode.

ArchMsg.Warning.warning_implementation_defined_taggednonsharedmemorytype

TAGS IMP_DEF DISPLAY Memory attribute has Tagged, non shared memory type; it is IMPLEMENTATION DEFINED whether the memory location is treated as Tagged or non-Tagged.

ArchMsg.Warning.warning_implementation_defined_write_debug_register_in_SCS

Write Access to Debug registers from the processor is IMPLEMENTATION DEFINED TAGS IMP_DEF. Fields:

DATA unsigned int
data attempted to be written.

PPB_OFFSET unsigned int
debug register offset.

ArchMsg.Warning.warning_invalid_tcr_granule

TAGS UNPREDICTABLE DISPLAY TCR.TG%{TG_ID:(0|1)} bits have been set to a granule size %{{REQUEST:d}}K not implemented - using %{{SUBSTITUTE:d}}K. Fields:

REQUEST signed int
page size requested (or 0 for reserved).

SUBSTITUTE signed int
best guess available page size.

TG_ID bool
bits TG0 or TG1.

ArchMsg.Warning.warning_load_multiple_user_registers_from_user_mode

TAGS UNPREDICTABLE DISPLAY An LDM(user registers) instruction executed from user mode.

ArchMsg.Warning.warning_mte_in_mem_attr_not_supported

TAGS UNPREDICTABLE DISPLAY Memory attribute %{ATTR} has Tagged memory type but Armv8.5-MTE is not implemented. Fields:

ATTR unsigned int

Memory attribute.

ArchMsg.Warning.warning_reserved_breakpoint_state_match

TAGS UNPREDICTABLE DISPLAY %{IS_BREAKPOINT:(Watchpoint|Breakpoint)} programmed with a reserved combination of HMC, SSC and %{IS_BREAKPOINT:(PAC|PMC)}. Fields:

IS_BREAKPOINT unsigned int

Is a breakpoint, not a watchpoint.

ArchMsg.Warning.warning_shareability

TAGS UNPREDICTABLE DISPLAY Unpredictable: combination of the 1st and 2nd stages of translation is Normal Inner Non-Cacheable, Outer Non-Cacheable.

ArchMsg.Warning.warning_software_step_set_while_enabled

TAGS UNPREDICTABLE DISPLAY MDSCR_EL1.SS set to 1 while software step debug exceptions are enabled.

ArchMsg.Warning.warning_thumb_instruction_wraps_around_memory

TAGS UNPREDICTABLE DISPLAY Thumb instruction wraps around memory.

ArchMsg.Warning.warning_ttbr_sbz_bits_are_not_zero

TAGS UNPREDICTABLE DISPLAY %{TTBR}[%{MSB:d}:0] should be zero, but are not. Fields:

MSB unsigned int

x-1.

TTBR enum

which TTBR register.

ArchMsg.Warning.warning_unaligned_address_dbgdtrrx_write

TAGS UNPREDICTABLE DISPLAY Unaligned address %{ADDR} in X0/R0 when DBGDTRRX written in memory access mode. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unaligned_address_dbgdtrtx_read

TAGS UNPREDICTABLE DISPLAY Unaligned address %{ADDR} in X0/R0 when DBGDTRTX read in memory access mode. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unaligned_to_device

TAGS UNPREDICTABLE DISPLAY Unaligned Access to Device Memory %{ADDR}. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unaligned_to_strongly_ordered

TAGS **UNPREDICTABLE** DISPLAY Unaligned Access to Strongly Ordered Memory %{ADDR}.

Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unknown_sau_rnr

TAGS **UNPREDICTABLE** DISPLAY SAU_RNR was set to an unsupported value.

ArchMsg.Warning.warning_unpredictable_AIRCR_PRIS_and_BFHFNMINs

TAGS **UNPREDICTABLE** DISPLAY The effect of setting both AIRCR.BFHFNMINs and AIRCR.PRIS to 1 is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_AIRCR_VECTCLRACTIVE_when_not_in_debug

TAGS **UNPREDICTABLE** DISPLAY The effect of writing a 1 to AIRCR.VECTCLRACTIVE if the processor is not halted in Debug state is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_AIRCR_VECTRESET_and_SYSRESETREQ

TAGS **UNPREDICTABLE** DISPLAY When the processor is halted in Debug state, if a write to the register writes a 1 to both VECTRESET and SYSRESETREQ, the behavior is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_AIRCR_VECTRESET_when_not_in_debug

TAGS **UNPREDICTABLE** DISPLAY The effect of writing a 1 to AIRCR.VECTRESET if the processor is not halted in Debug state is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_AIRCR_incorrect_VKEY

TAGS **UNPREDICTABLE** DISPLAY The value 0x05FA must be written to AIRCR.VKEY, otherwise the register write is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_EXC_RETURN_Reserved_Bit

EXC_RETURN[23:7] are reserved with the special condition that all bits should be written as one. Values other than all 1s are **UNPREDICTABLE**. TAGS **UNPREDICTABLE**. Fields:

EXC_RETURN unsigned int

Special PC value.

SBO_MASK unsigned int

Which bits Should Be One to not UNPRED.

ArchMsg.Warning.warning_unpredictable_PMEVTYPEPERn_ELO_VS_reserved_value

TAGS **UNPREDICTABLE** DISPLAY Unpredictable reserved value of PMEVTYPEPER<n>_ELO.VS.

ArchMsg.Warning.warning_unpredictable_change_to_DEMCR_MON_STEP_at_insufficient_priority

TAGS **UNPREDICTABLE** DISPLAY The effect of changing DEMCR.MON_STEP at an execution priority that is lower than the priority of the DebugMonitor exception is **CONSTRAINED UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_change_to_priority_of_active_exception

TAGS UNPREDICTABLE DISPLAY Changing the priority of an active exception is UNPREDICTABLE.

ArchMsg.Warning.warning_unpredictable_clear_lspact

TAGS UNPREDICTABLE DISPLAY UNPREDICTABLE state on context switch. Fields:

DRVEXC_TAKEN unsigned int

Derived exception taken, not pended.

REACHED_LAST_STORE unsigned int

Only the last faulted.

ArchMsg.Warning.warning_unpredictable_dbg_exit_unaligned_dlr

TAGS UNPREDICTABLE DISPLAY Debug exit to AArch32 with DLR[0] set to 1. PC[0] can be set to 0 or to DLR[0].

ArchMsg.Warning.warning_unpredictable_exception_catch

TAGS UNPREDICTABLE DISPLAY Generation of an exception catch event is unpredictable.

ArchMsg.Warning.warning_unpredictable_exception_return_inconsistent_state

TAGS UNPREDICTABLE DISPLAY state on exception return is unpredictable.

ArchMsg.Warning.warning_unpredictable_exception_return_instruction

TAGS UNPREDICTABLE DISPLAY exception return instruction is unpredictable.

ArchMsg.Warning.warning_unpredictable_in_debug_state

TAGS UNPREDICTABLE DISPLAY Instruction is UNPREDICTABLE when executed in debug state.

ArchMsg.Warning.warning_unpredictable_mair_encoding

TAGS UNPREDICTABLE DISPLAY Memory attribute %{ATTR} has invalid MAIR encoding; this may lead to unpredictable results. Fields:

ATTR unsigned int

Memory attribute.

ArchMsg.Warning.warning_unpredictable_pmu_counter_access

TAGS UNPREDICTABLE DISPLAY Access to PMU counters from non-secure EL0 or EL1 is UNPREDICTABLE with MDCR_EL2.HPMN set to 0.

ArchMsg.Warning.warning_unpredictable_prioritization_breakpoint_match_vector_catch

TAGS UNPREDICTABLE DISPLAY Two events with the same priority occurred at the same instruction: Address Matching Vector Catch debug event, and Breakpoint debug event. It is constrained unpredictable which is taken.

ArchMsg.Warning.warning_unpredictable_stack_selection

TAGS UNPREDICTABLE DISPLAY UNPREDICTABLE state on context switch.

ArchMsg.Warning.warning_unpredictable_tsize_out_of_range

TAGS UNPREDICTABLE DISPLAY %{TCR}.T%{TTBR:d}SZ (%{TSIZE:d} bits) is out of range. Must be between 25 and %{TSIZE_MAX:d}. Fields:

TCR enum

Which TCR.

TSize signed int

Value of TSize.

TSize_MAX signed int

The maximum allowed value.

TTBR unsigned int

Which TTBR region.

ArchMsg.Warning.warning_unpredictable_unaligned_pc_as_base_register

TAGS **UNPREDICTABLE** DISPLAY The PC value must be word-aligned, otherwise the behavior of the instruction is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_unaligned_pop_stack

TAGS **UNPREDICTABLE** DISPLAY **CONSTRAINED UNPREDICTABLE** unaligned stack frame `%{FRAME_ADDR:x}` in exception-return. Fields:

FRAME_ADDR unsigned int

unaligned stack frame address.

ArchMsg.Warning.warning_unpredictable_unaligned_to_non_writeback

TAGS **UNPREDICTABLE** DISPLAY Unaligned Access to Non-Writeback-Cacheable Memory.

ArchMsg.Warning.warning_unpredictable_vmsa_memattrib

TAGS **UNPREDICTABLE** DISPLAY Unpredictable vmsa memory attribute with texcb value programmed to `%{TEXCB}` when tex remap is `%{TEX_REMAP}`. Fields:

TEXCB unsigned int

texcb value.

TEX_REMAP bool

use tex remap.

ArchMsg.Warning.warning_unpredictable_vtcr_t0sz_sl0

TAGS **UNPREDICTABLE** DISPLAY The combination of VTCR.TOSZ=`%{TOSZ:d}` and VTCR.SL0=`%{SL0:d}` is unpredictable. Fields:

SL0 signed int

VTCR.SL0, starting level for stage 2 translation table walks.

TOSZ signed int

VTCR.TOSZ, required input address range.

ArchMsg.Warning.warning_unpredictable_write_DHCSR

TAGS **UNPREDICTABLE** DISPLAY **UNPREDICTABLE** set of bit-changes in DHCSR. Fields:

NEW unsigned int

data attempted to be written.

OLD unsigned int

previous value.

ArchMsg.Warning.warning_unpredictable_write_SHCSR

TAGS UNPREDICTABLE DISPLAY UNPREDICTABLE setting a pending bit to 1 for an exception with priority \geq execution priority.

ArchMsg.Warning.warning_unsupported_access_to_memory_mapped_register

TAGS UNPREDICTABLE DISPLAY Access to memory mapped register is unsupported.

ArchMsg.Warning.warning_user_jmcr_access

TAGS UNPREDICTABLE DISPLAY User %{WRITE:(write|read)} access to JMCR is unpredictable. Fields:

WRITE unsigned int
access is write.

ArchMsg.Warning.warning_wcr_mask_and_bas

TAGS UNPREDICTABLE DISPLAY DBGWCR_EL1.MASK is non-zero and BAS is not 0xff.

ArchMsg.Warning.warning_wcr_mask_reserved

TAGS UNPREDICTABLE DISPLAY DBGWCR_ELn.MASK is set to a reserved value.

ArchMsg.Warning.warning_wcr_non_configuous_bas

TAGS UNPREDICTABLE DISPLAY DBGWCR_EL1.BAS has non-contiguous set of ones.

ArchMsg.Warning.warning_wvr_masked_bit_not_zero

TAGS UNPREDICTABLE DISPLAY DBGWVRn_EL1.VA contains a bit masked by DBGWCRn_EL1.MASK that is not zero.

ArchMsg.Why.why_branch_target_exception

A branch target exception has occurred. DISPLAY branch target exception: target pc=%{PC}, target opcode=%{OPCODE}. PRIMARY KEY PC. Fields:

BTYPE unsigned int
pstate.BTYPE at time branch taken.

OPCODE unsigned int
opcode at branch target.

PC unsigned int
Virtual Address of branch target.

ArchMsg.Why.why_illegal_state

An illegal mode change has occurred. DISPLAY illegal mode change: %{MESSAGE}. Fields:

MESSAGE string
The message.

BRANCH_MISPREDICT

Simulating branch mispredict. Fields:

PC unsigned int
Origin address (or 0 if unavailable).

BRA_DIR

Direct branches, to immediate address. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISSET enum

The instruction set of the branch instruction.

IS_COND bool

Indicates if this is a conditional branch.

IS_HINTED bool

Whether the conditional instruction is hinted.

IS_ISB bool

Whether the instruction is ISB.

IS_LINK bool

Indicates if this is a branch with link.

PC unsigned int

The address of the branch instruction.

TARGET_ISET enum

The instruction set after the branch.

TARGET_PC unsigned int

The address the instruction branches to.

BRA_INDIR

Indirect branches, perhaps to a register. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISSET enum

The instruction set of the branch instruction.

IS_COND bool

Indicates if this is a conditional branch.

IS_HINTED bool

Whether the conditional instruction is hinted.

IS_ISB bool

Whether the instruction is ISB.

IS_LINK bool

Indicates if this is a branch with link.

PC unsigned int

The address of the branch instruction.

TARGET_ISET enum

The instruction set after the branch.

TARGET_PC unsigned int

The address the instruction branches to.

BRA_INDIR_CCFAIL

Indirect branches fail, perhaps to a register.

BRA_RET_CCFAIL

Branch return fail, perhaps to a register.

CACHE_MAINTENANCE_OP

Cache Maintenance Operation. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

DATA unsigned int

Specified MVA or Set/way.

FUNCTION enum

Clean / Invalidate.

SCOPE enum

Affected region.

SIDE enum

Inst / Data.

CCFAIL

Conditional instruction condition check fail. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CCFAIL_UNC

Unconditional instruction condition check fail. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CCPASS

Conditional instruction condition check pass. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CCPASS_UNC

Unconditional instruction condition check pass. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CODE_CACHE_MAINT

Code cache maintenance. Fields:

END_INCL unsigned int

Inclusive end address for VA or PA requests.

NS bool

Non-secure world for PA.

START unsigned int

Start address for VA or PA requests.

TYPE enum

Request type.

COMPILE_BLOCK_END

Last instruction of basic block translated. Fields:

VADDR unsigned int

Address of next instruction after this basic block.

COMPILE_BLOCK_START

First instruction of basic block translated. Fields:

VADDR unsigned int

Address of first instruction in basic block.

COMPILE_INST

ARM instruction compiled. Fields:

DISASS string

Disassembly of the instruction.

ISET enum

The instruction set of this instruction.

ITSTATE unsigned int

The ITSTATE current for the instruction.

OPCODE unsigned int

The opcode of the instruction.

PC unsigned int

The address of the instruction.

SIZE unsigned int

The size of the instruction in bytes.

CONTEXTIDR

Context ID Register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

NS bool

Secure or nonsecure banked register is accessed.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The new value written.

CONTEXT_SYNC

Called for every context synchronization event. Fields:

ARCH_CSE bool

Architectural Context Synchronization Event.

PC unsigned int

The address of the synchronization instruction.

SYSREG_SYNC bool

System register synchronization event.

CORE_ENDIAN

Core BE8 Big-Endian state changed. Fields:

BE8 bool

Core BE8 Big-Endian state.

CORE_INFO

Static processor attributes. Only triggered by a call to DumpState(). Fields:

ARCH_PROFILE enum

The architecture profile of the core.

CLUSTER_ID unsigned int

The cluster ID of this processor.

CORE_NUM unsigned int

The number of this core in an MP processor.

DETAILED_CORE_FEATURES enum

Detailed core features as a bitfield. Enum values are bitmasks to extract the info from the field.

FPU_VERSION enum

The VFP version implemented by the core.

MEM_ARCH enum

The memory architecture of the core.

MIDR_EL1 unsigned int

MIDR_EL1 value.

NUM_CORES unsigned int

The number of cores in this MP processor.

QUANTUM_SIZE unsigned int

The default quantum size of the core (in ticks).

SECURITY_FEATURES bool

Does the core have security features?.

CORE_LOADS

Processor load accesses. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ACQREL enum

Is this an acquire/release.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DATA unsigned int

The data read or written.

ELEMENT_SIZE unsigned int

Width of each element.

ENDIANNESS enum

Byte ordering of this access.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

LOCK enum

Normal, exclusive or locked access.

MECID unsigned int

Memory Encryption Context ID, if applicable.

MECID2 unsigned int

Memory Encryption Context ID of the second page, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PAGE_SIZE unsigned int

Page size as $\log_2(\text{size in bytes})$.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK, 2=Exclusive Failed, 3=Unsupported.

SIZE unsigned int

Width of the access in bytes. Only required if DATA is not traced.

TRANS bool

Is this a translated access.

VADDR unsigned int

The virtual address of the access.

CORE_REGS

Changes of the core registers R0 to R14. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number, 0 to 14.

MODE enum

Bank of the register accessed.

OLD_VALUE unsigned int

The old value overwritten.

PHYS_ID enum

The physical register accessed.

VALUE unsigned int

The new value written to the register.

CORE_REGS64

Changes of the core registers X0..X30, SP_ELn. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ID enum

The register number, 0-30 for X0-X30, >=32 for SP_ELn.

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

CORE_REGS64_READ

General purpose AArch64 register has been read X0..X30, SP_ELn. Fields:

ID enum

The register number, 0-30 for X0-X30, >=32 for SP_ELn.

VALUE unsigned int

The value read from the register.

CORE_STALLED_TRANSITION

Detect when a core is prevented from progressing PChannel states. Fields:

CURRENT_STATE enum

Current power mode of core.

REASON string

The reason a transition was stalled, and what must be done to unblock it.

TARGET_STATE enum

Desired power mode of core after transition.

CORE_STORES

Processor store accesses. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ACQREL enum

Is this an acquire/release.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DATA unsigned int

The data read or written.

ELEMENT_SIZE unsigned int

Width of each element.

ENDIANNESS enum

Byte ordering of this access.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

LOCK enum

Normal, exclusive or locked access.

MECID unsigned int

Memory Encryption Context ID, if applicable.

MECID2 unsigned int

Memory Encryption Context ID of the second page, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PAGE_SIZE unsigned int

Page size as $\log_2(\text{size in bytes})$.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK, 2=Exclusive Failed, 3=Unsupported.

SIZE unsigned int

Width of the access in bytes. Only required if DATA is not traced.

TRANS bool

Is this a translated access.

VADDR unsigned int

The virtual address of the access.

CP14_READ

System Coprocessor register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP14 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc1 unsigned int

opcode 1.

opc2 unsigned int

opcode 2.

CP14_WRITE

System Coprocessor register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP14 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc1 unsigned int

opcode 1.

opc2 unsigned int

opcode 2.

CP15_READ

System Coprocessor register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP15 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc1 unsigned int

opcode 1.

opc2 unsigned int

opcode 2.

CP15_READ64

System Coprocessor register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP15 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc unsigned int

opcode 1.

CP15_WRITE

System Control Coprocessor register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP15 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc1 unsigned int

opcode 1.

opc2 unsigned int
opcode 2.

CP15_WRITE64

System Coprocessor register write. Fields:

CORE_NUM unsigned int
Core number in a multi processor.

CRm unsigned int
CRm.

NS bool
Secure or nonsecure banked register is accessed.

REG_NAME string
Name of the CP15 register accessed.

SECURITY_STATE enum
The current security state.

UNDEF bool
The register accessed is undefined.

UPDATED_VALUE unsigned int
Updated value of the register now it has been written.

VALUE unsigned int
The new value written.

opc unsigned int
opcode 1.

CPSR

CPSR change. (In AArch64 PSTATE represented in SPSR format). Fields:

CORE_NUM unsigned int
Core number in a multi processor.

OLD_VALUE unsigned int
The old CPSR value.

UNKNOWN unsigned int
Bits within the register that have unknown value.

VALUE unsigned int
The new CPSR value.

CRYPTO_SPEC

Every crypto instruction speculatively executed.

CorePowerStateChange

Core power-up state has changed. DISPLAY Core powered %{PU:{up|down}}. Fields:

PU bool

Whether the core powered up.

DATA_CACHE_SET_ALLOC_TAGS

Set allocation tags from DC GVA instruction. Fields:

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

NSDESC unsigned int

The security state of the access.

PADDR unsigned int

The physical address of the access.

PAS enum

The physical address space of the page.

RESPONSE enum

0=Aborted, 1=OK.

SIZE unsigned int

The size of memory in bytes.

TAG unsigned int

The value of the tag being set.

TAGGING_TYPE enum

Type of Memory tagging (Physical or Virtual).

TAG_VADDR unsigned int

The virtual address of the Virtual Tag if Virtual tagging is used; otherwise 0.

VADDR unsigned int

Virtual address used for setting allocation tags.

DATA_CACHE_ZERO

Zero and invalidate cache line from DC ZVA instruction. Fields:

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

NSDESC unsigned int

The security state of the access.

PADDR unsigned int

The physical address of the access.

PADDR_TAG unsigned int

If tag checked, the physical address tag for the access.

PAS enum

The physical address space of the page.

RESPONSE enum

0=Aborted, 1=OK.

SIZE unsigned int

The size of zero'd memory in bytes.

TAG_CHECKED bool

Is this access tag checked.

VADDR unsigned int

Virtual address of the zero'd cache line.

DEBUG_EVENT

Hardware debug support event. Fields:

EVENT enum

Description of event.

VALUE unsigned int

data value.

DEBUG_TARGET_ISET

Hardware debug target ISA. Fields:

TARGET_ISET enum

The target instruction set of the debug state.

DMI_ALLOCATE

DMI allocated. Fields:

ACCESS_DIRECTION enum

READ / WRITE.

PADDR unsigned int

The physical address of the DMI.

PTR unsigned int

DMI range pointer.

SIDE enum

DSIDE / ISIDE.

SIZE unsigned int

The size of DMI.

VADDR unsigned int

The virtual address of the DMI..

DMI_HIT

DMI hit, access used DMI (only fire for small DMI regions i.e. less than a full page). Fields:

ACCESS_DIRECTION enum

READ / WRITE.

PADDR unsigned int

The physical address of the DMI.

SIDE enum

DSIDE / ISIDE.

SIZE unsigned int

The size of the DMI.

VADDR unsigned int

The virtual address of the DMI..

DMI_REVOKE

DMI revoked. Fields:

ACCESS_DIRECTION enum

READ / WRITE.

PADDR unsigned int

The physical address of the DMI.

SIDE enum

DSIDE / ISIDE.

SIZE unsigned int

The size of DMI.

VADDR unsigned int

The virtual address of the DMI..

END_COMPILE

Compilation end. Fields:

END_OF_PAGE_COUNT unsigned int

Number of basic blocks exited due to page boundary since START_COMPILE.

FETCHFAIL_COUNT unsigned int

Number of basic blocks exited due to ifetch failure START_COMPILE.

INST_COUNT unsigned int

Number of instructions compiled since START_COMPILE.

NONSEQ_COUNT unsigned int

Number of basic blocks exited due to non-sequential instructions since START_COMPILE.

PAGE_STRADDLE_COUNT unsigned int

Number of basic blocks exited due to unaligned instructions crossing page since START_COMPILE.

ESB

Error Synchronization Barrier hit. Fields:

ASYNC bool

ESB is async.

IMPLICIT bool

ESB is implicit.

EXCEPTION

Exception taken. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ESR unsigned int

The value assigned to ESR when exception is taken.

ESR_EC unsigned int

The Value assigned to ESR.EC when exception is taken.

IS_PHYSICAL bool

Physical or Virtual exception.

IS_TRAP bool

Exception coming from a trap.

LR unsigned int

The value assigned to the link register.

PC unsigned int

The location where the exception occurred.

PREFERRED_RETURN unsigned int

The preferred return address for the exception.

TAKEN_LOCALLY bool

Whether exception is taken locally(same exception level).

TARGET_ISET enum

The instruction set of the exception handler code.

TARGET_PC unsigned int

The address the exception branches to.

VECTOR enum

The exception vector.

VECTOR_OFFSET unsigned int

The offset of an exception vector from the base address.

EXCEPTION_END

Every exception completed.

EXCEPTION_INFO

Trace human-readable reason of the current exception for data/instruction aborts, floating point exceptions and unknown exceptions. Fields:

DESCRIPTION string

Reason for current exception.

ESR_EC unsigned int

The value of ESR.EC when exception is taken.

EXCEPTION_RAISE

Every exception raised.

EXCEPTION_RETURN

Branches on leaving exception. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISSET enum

The instruction set of the branch instruction.

IS_COND bool

Indicates if this is a conditional branch.

IS_HINTED bool

Whether the conditional instruction is hinted.

IS_ISB bool

Whether the instruction is ISB.

IS_LINK bool

Indicates if this is a branch with link.

PC unsigned int

The address of the branch instruction.

TARGET_ISET enum

The instruction set after the branch.

TARGET_PC unsigned int

The address the instruction branches to.

EXCEPTION_RETURN_PREBRANCH

Exception return event that must occur before PC and processor state have updated.

EXCEPTION_START

Every exception started.

FIQ_TAKEN

FIQ taken exception.

FP_EXCEPTIONS_SIGNALLED

Trace which FP exceptions were signalled as a part of the FP operations performed by an instruction. Fields:

FP_EXCEPTIONS_SIGNALLED_MASK unsigned int

Indicates which FP exceptions were signalled by an instruction. This uses the same encoding as the cumulative floating-point exception bitfields in FPSR.

FP_STATE

Floating point state. Fields:

FPSR unsigned int

FPSR register value.

MASK unsigned int

The mask indicating updated cumulative bits.

FREQ_CHANGED

Core frequency changed. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The instruction count of this core.

NEW_FREQ signed int

The new frequency of this core (expressed in Hz).

OLD_FREQ signed int

The old frequency of this core (expressed in Hz).

HLT

HLT instruction occurred. Fields:

IMM16 unsigned int

The 16-bit immediate value encoded in HLT instruction.

INST

Every instruction executed. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DEBUG_STATE bool

The instruction is executed in debug state.

DISASS string

Disassembly of instruction.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISET enum

The current instruction set.

ITSTATE unsigned int

The current ITSTATE.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

MODE enum

The mode the core is in.

NS unsigned int

The core's non-secure bit.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

OPCODE unsigned int

The opcode of the instruction.

PADDR unsigned int

The physical address of the instruction.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the instruction.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the page.

PC unsigned int

The address of the instruction.

SECURITY_STATE enum

The core's security state.

SIZE unsigned int

The size of the instruction in bytes.

INST_END

Every instruction completed.

INST_START

Every instruction started. Fields:

DEBUG_STATE *bool*

The instruction is executed in debug state.

ISET *enum*

The current instruction set.

MODE *enum*

The mode the core is in.

NS *enum*

The current Secure State.

PC *unsigned int*

The address of the conditional instruction.

SECURITY_STATE *enum*

The current Security State.

INST_STRADDLE

Instruction straddles boundary. Fields:

DEBUG_STATE *bool*

The instruction is executed in debug state.

ISET *enum*

The current instruction set.

MODE *enum*

The mode the core is in.

NS *enum*

The current Secure State.

PC *unsigned int*

The address of the conditional instruction.

SECURITY_STATE *enum*

The current Security State.

IRQ_TAKEN

IRQ taken exception.

ISB

ISB instruction occurred. Fields:

PC *unsigned int*

The address of the ISB instruction.

LOAD_MULTIPLE_REGS_LIST

List of destination registers (not source) of a multiple registers load instruction. Fields:

REG string

Register to add into the list.

LOCAL_MONITOR

Local monitor activity. Fields:

PADDR unsigned int

Local Monitor Address.

State enum

State of the monitor (Open/Exclusive).

MAPPING_CREATE

A memory mapping has been created. This is an implementation detail of the model and so may change without notice. Fields:

ASID unsigned int

The ASID used during translation, or 0 if no ASID was used.

IS_VALID bool

Whether the mapping is valid or not.

PADDR unsigned int

Base physical address of region, or 0 if not known.

SIDE enum

I-side or D-side.

SIZE unsigned int

Size of the page, or 0 if not known.

VADDR unsigned int

Base virtual address of region.

VMID unsigned int

The VMID used during translation, or 0 if no VMID was used.

MAPPING_DESTROY

A memory mapping has been destroyed. This is an implementation detail of the model and so may change without notice. Fields:

ASID unsigned int

The ASID used during translation.

CAUSE enum

The cause of invalidation.

PADDR unsigned int

Base physical address of page.

SIDE enum

I-side or D-side.

SIZE unsigned int

Size of the page.

VADDR unsigned int

Base virtual address of page.

VMID unsigned int

The VMID used during translation.

MEMMAP_DEBUG_READ

Memory mapped reads to the debug registers. Fields:

ADDR unsigned int

address.

CORE_NUM unsigned int

Core number in a multi processor.

EXT bool

Whether access is from an external device (such as the DAP).

NS enum

Secure state of the access.

REG_NAME string

Name of the debug register accessed.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

MEMMAP_DEBUG_WRITE

Memory mapped writes to the debug registers. Fields:

ADDR unsigned int

address.

CORE_NUM unsigned int

Core number in a multi processor.

EXT bool

Whether access is from an external device (such as the DAP).

NS enum

Secure state of the access.

REG_NAME string

Name of the debug register accessed.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The new value written.

MMU_D128_TTB_READ

This event is triggered by reads caused by a D128 translation table walk. Fields:

ABORTED bool

Was the walk successful.

DATA_HIGH unsigned int

The data upper bits read.

DATA_LOW unsigned int

The data lower bits read.

IPA unsigned int

For stage 1, the IPA of the read.

LEVEL unsigned int

Translation table level.

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

OUTADDR unsigned int

The output physical address.

PADDR unsigned int

The physical address of the read.

PASmem enum

Physical Address Space of the accessed memory.

PASreq enum

Physical Address Space of the requested lookup.

REGIME_EL enum

Entry matches in this translation regime.

REQUESTER enum

The requester of this table walk.

SIDE enum

Inst / Data.

STAGE unsigned int

Translation stage.

MMU_D128_TTB_WRITE

This event is triggered by writes caused by a D128 translation table walk. Fields:

ABORTED bool

Was the walk successful.

DATA_HIGH unsigned int

The data upper bits written.

DATA_LOW unsigned int

The data lower bits written.

IPA unsigned int

For stage 1, the IPA of the write.

IS_HACDBS_ACCESS bool

If the access was due to HACDBS_CLEAN.

LEVEL unsigned int

Translation table level.

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

OUTADDR unsigned int

The output physical address.

PADDR unsigned int

The physical address of the write.

PASmem enum

Physical Address Space of the accessed memory.

PASreq enum

Physical Address Space of the requested lookup.

REGIME_EL enum

Entry matches in this translation regime.

SIDE enum

Inst / Data.

STAGE unsigned int

Translation stage.

MMU_TRANS

Address translation information. Fields:

ASID unsigned int

Address space identifier.

CORE_NUM unsigned int

Core number in a multi processor.

Hyp bool

Entry matches in Hyp state only.

INNERCACHE_RA bool

Is the inner cache allocate on read.

INNERCACHE_TRANSIENT bool

Is the inner write-through transient.

INNERCACHE_TYPE enum

Inner Caching scheme (NC/MB/WA).

INNERCACHE_WA bool

Is the inner cache allocate on write.

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MEMTYPE enum

Memory type.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC enum

Is secure side supposed to access secure or nonsecure memory.

OUTERCACHE_RA bool

Is the outer cache allocate on read.

OUTERCACHE_TRANSIENT bool

Is the outer write-through transient.

OUTERCACHE_TYPE enum

Outer Caching scheme (NC/MB/WA).

OUTERCACHE_WA bool

Is the outer cache allocate on write.

PADDR unsigned int

Physical address of the access.

PAGESIZE unsigned int

Page size as log2(size).

PAS enum

PAS supposed to be accessed is secure or nonsecure memory or Root/Realm if RME is implemented.

SH enum

Shareability.

SIDE enum

Inst / Data.

STAGE1_PERM unsigned int

Stage 1 permissions, [5:3] Privileged access (XWR) [2:0] User access (XWR).

STAGE2_PERM unsigned int

Stage 2 permission mask [5:3] Privileged access (XWR) [2:0] User access (XWR).

VADDR unsigned int

Virtual address of the access.

VMID unsigned int

Virtual machine identifier.

XS bool

XS Attribute.

nG enum

Flag indicating whether ASID will be matched.

MMU_TTB_READ

This event is triggered by reads caused by a translation table walk. Fields:

ABORTED bool

Was the walk successful.

DATA unsigned int

The data read.

DESC_KIND enum

The encoding scheme of the descriptors used for this walk.

IPA unsigned int

For stage 1, the IPA of the read.

LEVEL unsigned int

Translation table level.

LPAE bool

Is this for an LPAE translation (*DEPRECATED*, use DESC_KIND).

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSmem bool

Non secure memory access.

NSreq bool

Non secure request.

OUTADDR unsigned int

The output physical address.

PADDR unsigned int

The physical address of the read.

PASmem enum

Physical Address Space of the accessed memory.

PASreq enum

Physical Address Space of the requested lookup.

REGIME_EL enum

Entry matches in this translation regime.

REQUESTER enum

The requester of this table walk.

SIDE enum

Inst / Data.

STAGE unsigned int

Translation stage.

MMU_TTB_WRITE

This event is triggered by writes caused by a translation table walk. Fields:

ABORTED bool

Was the walk successful.

DATA unsigned int

The data written.

DESC_KIND enum

The encoding scheme of the descriptors used for this walk.

IPA unsigned int

For stage 1, the IPA of the write.

IS_HACDBS_ACCESS bool

If the access was due to HACDBS_CLEAN.

LEVEL unsigned int

Translation table level.

LPAE bool

Is this for an LPAE translation (*DEPRECATED*, use DESC_KIND).

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSmem bool

Non secure memory access.

NSreq bool

Non secure request.

OUTADDR unsigned int

The output physical address.

PADDR unsigned int

The physical address of the write.

PASmem enum

Physical Address Space of the accessed memory.

PASreq enum

Physical Address Space of the requested lookup.

REGIME_EL enum

Entry matches in this translation regime.

SIDE enum

Inst / Data.

STAGE unsigned int

Translation stage.

MODE_CHANGE

Mode change. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

MODE enum

The new mode.

NON_SECURE enum

The core's new non-secure bit.

OLD_MODE enum

The old mode.

SECURITY_STATE enum

The new security state.

NOT_TAG_CHECKED_INST

MTE Tag not checked for current instruction.

PERIODIC

Called for every quantum. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The instruction count of this CPU.

PC unsigned int

The address of the next instruction to be executed on this CPU.

PMU_COUNTER_OVERFLOW

PMU counter overflow. Fields:

COUNTER_GROUP enum

Counter group.

COUNTER_RANGE enum

Counter range.

EVENT_ID unsigned int

Event Identifier, field is valid only for Event counter group.

INDEX unsigned int

Counter index (as selected by PMSELR), field is valid only for Event/Cycle counter groups.

INTERRUPT bool

Is interrupt enabled on overflow for this counter.

POTINDEX_UPDATE

Value of POT index for current pc changed. Fields:

NEW_POTINDEX unsigned int

New POTIndex.

OLD_POTINDEX unsigned int

Old POTIndex.

PREFETCH_MEMORY64

Prefetch from PRFM or PRFUM instructions. Fields:

PRFOP enum

Prefetch hint.

VADDR unsigned int

Virtual address of the location that should be prefetched.

PRELOAD64_REQUEST

Preload requests from PRFM or PRFUM instructions. Fields:

PRFOP enum

Prefetch hint.

VADDR unsigned int

Virtual address of the location that should be prefetched.

PRELOAD_DATA

Data preload from PLD instruction. Fields:

VADDR unsigned int

Virtual address of the data that should be preloaded.

PRELOAD_INST

Instruction preload from PLI instruction. Fields:

VADDR unsigned int

Virtual address of the instruction that should be preloaded.

PRE_CORE_LOAD

Trace just before a core load. Fields:

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

PRE_CORE_STORE

Trace just before a core store. Fields:

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

PRE_TTB_READ

This event is triggered before reads caused by a translation table walk. Fields:

ADDR unsigned int

The physical address of the read.

PSTATE

PSTATE fields update. Fields:

ALLINT unsigned int

All interrupt mask bit.

BTYPE unsigned int

Branch target identification bit.

CORE_NUM unsigned int

Core number in a multi processor.

DAIF unsigned int

Exception mask bits.

DIT unsigned int

Data Independent Timing bit.

EL unsigned int

Current Exception level field.

EXLOCK unsigned int

Exception return state lock bit.

IL unsigned int

Illegal Execution state bit.

NZCV unsigned int

Condition flags.

PACM unsigned int

PACM bit.

PAN unsigned int

Privileged Access Never state bit.

PM unsigned int

PMU exception mask bit.

PPEND unsigned int

PMU exception pending bit.

SM unsigned int

StreamingMode status bit.

SP unsigned int

Stack pointer register selection bit.

SS unsigned int

Software Step bit.

SSBS unsigned int

Speculative Store Bypass Safe bit.

TCO unsigned int

Tag Check Override bit.

TOUCHED unsigned int

Mask of the fields that have been touched since this trace has last been emitted. Note that touched does not necessarily mean the value changed. The index of the field is the bit index in the mask plus the index of this field plus one.

UAO unsigned int

User Access Override bit.

UINJ unsigned int

Undefined Instruction exceptions bit.

ZA unsigned int

ZA storage and ZT0 register enable bit.

nRW unsigned int

Current Execution state bit.

PSTATE_PACTIVE_OPMODE_TRANSITION

Trace pactive and Opmode mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Operating mode after the transition.

OLD_STATE enum

Operating mode before the transition.

PSTATE_PACTIVE_OPMODE_TRANSITION_DENIED

Trace the denied pactive and Opmode mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Operating mode after the transition.

OLD_STATE enum

Operating mode before the transition.

REASON string

Reason for the transition being denied.

PSTATE_PACTIVE_POWER_TRANSITION

Trace pactive and power mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Power mode after the transition.

OLD_STATE enum

Power mode before the transition.

PSTATE_PACTIVE_POWER_TRANSITION_DENIED

Trace the denied pactive and power mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Power mode after the transition.

OLD_STATE enum

Power mode before the transition.

REASON string

Reason for the transition being denied.

PSTATE_PACTIVE_THREAD_OPMODE_TRANSITION

Trace pactive and Opmode mode transitions for threads on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Operating mode after the transition.

OLD_STATE enum

Operating mode before the transition.

QUANTUM_START

Called at the start of every quantum.

RANGE_PRELOAD_REQUEST

Preload requests from RPRFM. Fields:

COUNT unsigned int

Number of blocks of data to access.

LENGTH unsigned int

Number of contiguous bytes to be accessed in each block.

OP unsigned int

Range prefetch operation .

REUSE unsigned int

Maximum number of bytes to be accessed before next RPRFM .

STRIDE signed int

Stride to advance after accessing LENGTH bytes.

VADDR unsigned int

Base virtual address of range prefetch request.

RUN_STATE

Run state transition. Fields:

INST_COUNT unsigned int

Ticks count at point of transition.

NEW enum

New run state.

OLD enum

Old run state.

SEMIHOSTING_CALL

Call of a semihost function occurred. Fields:

PC unsigned int

The program counter after the semihosting call.

SEMIHOSTING_PRECALL

About to call semihost. Fields:

EL2 bool

Use EL2 translation regime.

NS bool

Is Non-Secure.

REG_WIDTH unsigned int

The current register width in bytes.

SIGNAL

External signal state change. Fields:

SIGNAL enum

Signal that changed.

STATE bool

Signal asserted state.

SOFTWARE_STEP

Return the Debug Software Step state. Fields:

STATE enum

Software Step state.

SPSR

SPSR change. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

MODE enum

Which of the banked SPSR registers is written.

OLD_VALUE unsigned int

The old SPSR value.

VALUE unsigned int

The new SPSR value.

START_COMPILE

Compilation started. Fields:

VADDR unsigned int

Instruction where compilation begins.

STG_MULTIPLE_MICRO_OPS_ADDRESS_RANGE

Address range for MTE stores with multiple uops. Fields:

ADDR_END unsigned int

Final address.

ADDR_START unsigned int

Starting address.

STORE_MULTIPLE_REGS_ADDRESS_RANGE

Address range for stores with multiple registers. Fields:

ADDR_END unsigned int

Final address.

ADDR_START unsigned int

Starting address.

SUBOPTIMAL_LDST_RETIRED

suboptimal load/store. Fields:

ACCESS_SIZE unsigned int

Log2 of the access width in bytes used for alignment checking (1: half-word, 2: word ...).

OPTIMAL_ALIGN_SELECTOR unsigned int

Selector to decide the optimal alignment size (1: access width, 2: 4-bytes, 3: 8-bytes, ..., 12: 4-kbytes).

VADDR unsigned int

The virtual address of the access.

SYNC

Called for every synchronization, that is every quantum break. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The instruction count of this CPU.

LOCAL_QUANTUM unsigned int

The local quantum of this CPU.

LOCAL_TIME unsigned int

The local time of this CPU.

SYNC_REASON enum

Reason for sync.

SYSCALL

System call instruction executed. Fields:

IMM unsigned int

Immediate value of the system call instruction.

TYPE enum

System call type.

VADDR unsigned int

Instruction that caused the system call.

SYSREG_READ128

System Coprocessor 128 bit register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

REGNUM enum

Internal register number.

REGNUM_PHYS enum

Internal register number physically accessed (e.g. in presense of VHE redirection).

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_READ64

System Coprocessor register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REGNUM enum

Internal register number.

REGNUM_OPERAND enum

Register number corresponding to the instruction operand.

REGNUM_PHYS enum

Internal register number physically accessed (e.g. in presense of VHE redirection).

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_UPDATE128

Triggers when the system updates a register. Fields:

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG enum

Register number.

UNKNOWN unsigned int

Bits of the register which became unknown.

VALUE unsigned int

Value written to the register.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_UPDATE32

Triggers when the system updates a register. Fields:

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG enum

Register number.

UNKNOWN unsigned int

Bits of the register which became unknown.

VALUE unsigned int

Value written to the register.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int
opcode 2.

SYSREG_UPDATE64

Triggers when the system updates a register. Fields:

CRm unsigned int
CRm.

CRn unsigned int
CRn.

NS bool
Secure or nonsecure banked register is accessed.

REG enum
Register number.

UNKNOWN unsigned int
Bits of the register which became unknown.

VALUE unsigned int
Value written to the register.

opc unsigned int
opcode 1.

opc0 unsigned int
opcode 0.

opc2 unsigned int
opcode 2.

SYSREG_WRITE128

System Coprocessor 128 bit register write. Fields:

CORE_NUM unsigned int
Core number in a multi processor.

CRm unsigned int
CRm.

CRn unsigned int
CRn.

REGNUM enum
Internal register number.

REGNUM_PHYS enum
Internal register number physically accessed (e.g. in presense of VHE redirection).

UNDEF bool
The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_WRITE64

System Coprocessor register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REGNUM enum

Internal register number.

REGNUM_OPERAND enum

Register number corresponding to the instruction operand.

REGNUM_PHYS enum

Internal register number physically accessed (e.g. in presense of VHE redirection).

UNDEF bool

The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

UNALIGNED_LDST_RETIRED

Processor unaligned load/store. Fields:

ACCESS_SIZE unsigned int

Log2 of the access width in bytes used for alignment checking (1=>half-word, 2->word ...).

NUMBER_OF_BEATS unsigned int

Number of accesses (beats) in this burst (Total size in bytes is ACCESS_SIZE * NUMBER_OF_BEATS).

VADDR unsigned int

The virtual address of the access.

VFP_D_REGS

VFP/NEON D 64 bit register write. Fields:

ALIASING enum

Type of register aliasing used.

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a write to the corresponding bit in the register. For example, 0x1 would indicate a write to VALUE[0].

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

VFP_D_REGS_READ

VFP/NEON D 64 bit register read. Fields:

ID unsigned int

The register number.

VALUE unsigned int

The current value of the register.

VFP_Q_REGS

VFP/NEON Q 128 bit register write. Fields:

ALIASING enum

Type of register aliasing used.

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a write to the corresponding byte in the register. For example, 0x1 would indicate a write to VALUE[7:0].

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

VFP_Q_REGS_READ

VFP/NEON Q 128 bit register read. Fields:

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a read to the corresponding byte in the register. For example, 0x1 would indicate a read to VALUE[7:0].

VALUE unsigned int

The current value of the register.

VFP_SYS_REGS

Writes to the VFP/NEON units system registers. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ID enum

Which VFP system register is written.

MASK unsigned int

The mask indicating updated cumulative bits.

OLD_VALUE unsigned int

The register's old value overwritten.

VALUE unsigned int

The new value written to the VFP system register.

VFP_S_REGS

VFP/NEON S 32 bit register write. Fields:

ALIASING enum

Type of register aliasing used.

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a write to the corresponding bit in the register. For example, 0x1 would indicate a write to VALUE[0].

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

VFP_S_REGS_READ

VFP/NEON S 32 bit register read. Fields:

ID unsigned int

The register number.

VALUE unsigned int

The current value of the register.

VIRTUAL_TAG_ALLOCATED

Virtual tag information when allocated. Fields:

TAG_VADDR unsigned int

Virtual tag virtual address.

VADDR unsigned int

Data Virtual address.

VIRTUAL_TAG unsigned int

Virtual tag value.

WAYPOINT

Signals end of basic block execution. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ISSET enum

Origin instruction set.

IS_COND bool

Indicates if this is a conditional waypoint.

PC unsigned int

Origin address (or 0 if unavailable).

TAKEN bool

Indicates if this waypoint was taken.

TARGET unsigned int

Destination address.

TARGET_ISET enum

Destination instruction set.

WFE_END

WFE ended. Fields:

INST_COUNT unsigned int

Ticks count when leaving WFE.

WFE_EVENT_REGISTER

WFE event register status: set/clear, reason. Fields:

INST_COUNT unsigned int

Ticks count.

REASON enum

Reason for set/clear. Only REASON==0 (Cleared by WFE) clears the bit, all other reasons set it.

WFE_IGNORED

WFE ignored. Fields:

EVENT bool

This WFE was ignored because the event register was set.

INST_COUNT unsigned int

Ticks count when ignoring WFE.

TRAPPED bool

This WFE was trapped.

WFE_START

WFE entered. Fields:

INST_COUNT unsigned int

Ticks count when entering WFE.

WFI_END

WFI ended. Fields:

INST_COUNT unsigned int

Ticks count when leaving WFI.

WFI_IGNORED

WFI ignored. Fields:

DISABLED bool

This WFI was ignored because WFI is disabled.

INST_COUNT unsigned int

Ticks count when ignoring WFI.

REASON enum

specifies reason why WFI trace was ignored.

TRAPPED bool

This WFI was trapped.

WFI_START

WFI entered. Fields:

INST_COUNT unsigned int

Ticks count when entering WFI.

WFI_WAKEUP

WFI wakeup. Fields:

INST_COUNT unsigned int

Ticks count when WFI wakeup occurred.

REASON enum

Reason for wakeup.

2.31 ARM_Cortex-A75

This section describes the trace sources.

AA64_ASE_SVE_REGS

Modification of AArch64 Advanced SIMD or SVE Vector Registers (via SIMD or Scalar operations), SVE Predicate Registers and ZT0. Fields:

ID enum

The register identifier.

MASK unsigned int

Bitmask of the register to signal the modified bits in the VALUE field.

SM bool

Whether the PE is in Streaming Mode.

VALUE unsigned int

Value of AArch64 vector register.

ASYNC_MEMORY_FAULT

Context ID Register write. Fields:

FAULT unsigned int

Fault status present in ESR.ISS encoding for Serror in ARCH64 and present in DFSR encoding for external abort format in ARCH32.

PADDR unsigned int

Physical Address (or 0 if unavailable).

VADDR signed int

Virtual Address (or 0 if unavailable).

ATOMIC_END_ACCESS

Bus trace access for atomics. Fields:

ACCESS_FAIL bool

Memory access failed.

ADDR unsigned int

The virtual address of the access.

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

COMPARE_VALUE unsigned int

Compare value for CAS.

LOAD_VALUE unsigned int

Loaded values.

NSDESC unsigned int

The security state of the access.

OPERAND_VALUE unsigned int

Operation's operand.

OPERATION enum

Operation type.

PADDR unsigned int

The physical address of the access.

PRIV bool

Is this a privileged access?.

ATOMIC_START_ACCESS

Bus trace access for atomics. Fields:

ADDR unsigned int

The virtual address of the access.

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

COMPARE_VALUE unsigned int

Compare value for CAS.

NSDESC unsigned int

The security state of the access.

OPERAND_VALUE unsigned int

Operation's operand.

OPERATION enum

Operation type.

PADDR unsigned int

The physical address of the access.

PRIV bool

Is this a privileged access?.

ArchMsg.Error.exit_code

DISPLAY exit code: %{EXITCODE} component kind: %{KIND} reason: %{REASON}. Fields:

EXITCODE signed int

Exit code to be returned.

KIND string

Component kind that invoked the exit code trace.

REASON string

Reason why the exit code trace was invoked (optional).

ArchMsg.Warning.IgnoredWarmResetRequest

Event fires when software attempts to request warm reset through write to deprecated DBGPRCR.CWRR.

ArchMsg.Warning.RAS_ErrselrUnpredictableState

DISPLAY ERRSELR is in unpredictable state due to value greater than number_of_error_records written to it.

ArchMsg.Warning.RAS_UnpredictableStateWrite

DISPLAY %{REG_NAME} write ignored due to unpredictable ERRSELR state. Fields:

REG_NAME string

Name of register being written.

ArchMsg.Warning.ReportedRevisionRegisterRead

DISPLAY Reported revision and patch do not match that of the cpu. Fields:

ORIGINAL_VALUE unsigned int

The value the register would have based on the modelled revision.

REGISTER string

The name of the register that has been read.

REPORTED_VALUE unsigned int

The value the register has been overridden to.

ArchMsg.Warning.Unpredictably Indexed PM Event Register

TAGS **UNPREDICTABLE** DISPLAY Event %{IsDirect:(Selector value|Register)} %{SEL} is >= %{N}. Fields:

IsDirect unsigned int

Direct Access rather than indirect.

N unsigned int

Number of Accessible Registers.

SEL unsigned int

Selector.

ArchMsg.Warning.cache_contents_unknown

A cache has been activated without being invalidated and may contain **UNKNOWN** content. DISPLAY %{SIDE:(D|I)}-Cache was enabled in %{REGIME:(SECURE|NON_SECURE|HYP)} regime but was not invalidated since power-on: cache lines could contain **UNKNOWN** content %{SIDE:(Note: D-side caches must invalidate SECURE lines even if the cache is only used by NON_SECURE code in order to ensure that unexpected dirty lines tagged SECURE are not naturally evicted)}. Fields:

REGIME unsigned int

Current translation regime.

SIDE unsigned int

0 d-side cache or unified cache, 1 i-side.

ArchMsg.Warning.dcimvac_matches_watchpoint

TAGS **UNPREDICTABLE** DISPLAY Watchpoints matching an AArch32 DCIMVA are implementation defined.

ArchMsg.Warning.decode_fieldmismatch

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with different fields %{FIELD1} and %{FIELD2}. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_frachitsnegative

TAGS **UNPREDICTABLE** DISPLAY VCVT Instruction is unpredictable with a negative number of fraction bits.

ArchMsg.Warning.decode_initblock

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable in an IT block.

ArchMsg.Warning.decode_initblocknotlast

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable in an IT block when not the last.

ArchMsg.Warning.decode_invalid_condition

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable when the condition is not AL.

ArchMsg.Warning.decode_invalidfieldcombination

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with the values in %{FIELD1} and %{FIELD2}. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_invalidvalue

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with the value in %{FIELD}. Fields:

FIELD enum

Field name.

ArchMsg.Warning.decode_mem_hint_unallocated

TAGS **UNPREDICTABLE** DISPLAY There is no memory hint allocated to this bit pattern.

ArchMsg.Warning.decode_registermatch

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with %{FIELD1} and %{FIELD2} as the same register. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_registermismatch

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with %{FIELD1} and %{FIELD2} as different registers. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_registeroutofrange

TAGS **UNPREDICTABLE** DISPLAY Floating-point instruction is unpredictable indexing beyond the end of the register bank.

ArchMsg.Warning.decode_sbzsbo

TAGS **UNPREDICTABLE** DISPLAY Reserved bits in the instruction are not canonical.

ArchMsg.Warning.decode_transactiontoobig

TAGS **UNPREDICTABLE** DISPLAY Attempt to perform a memory transaction of more than 128 bytes.

ArchMsg.Warning.decode_unpred_other

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable.

ArchMsg.Warning.decode_unpreduseofpc

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable when %{FIELD} is PC. Fields:

FIELD enum

Field name.

ArchMsg.Warning.decode_unpreduseofr13

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable when %{FIELD} is R13. Fields:

FIELD enum

Field name.

ArchMsg.Warning.decode_unpreduseofr13inreglist

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with R13 in register-list.

ArchMsg.Warning.decode_useofd16orhigher

TAGS **UNPREDICTABLE** DISPLAY Attempt to use register D16 or higher where it is not allowed.

ArchMsg.Warning.decode_writebackandbaseinlist

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with write back when the base register is in the transfer list.

ArchMsg.Warning.decode_zeroregistersinlist

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable when no registers are to be transferred.

ArchMsg.Warning.recursive_branch

An instruction is performing a branch that targets the same instruction. If this was intended then a WFI might be more effective DISPLAY recursive branch detected at PC=%{PC}.

PRIMARY KEY PC. Fields:

PC unsigned int

Address of instruction causing the branch.

ArchMsg.Warning.recursive_exception

An instruction has generated an exception that targets the same instruction. DISPLAY recursive %{TYPE} exception detected at PC=%{PC} (CPSR=%{CPSR}). PRIMARY KEY PC. Fields:

CPSR unsigned int

Processor state.

PC unsigned int

Address of instruction causing the exception.

TYPE enum

Type of exception.

ArchMsg.Warning.reentrant_vector_catch

TAGS **UNPREDICTABLE** DISPLAY Vector catch debug event on Prefetch or Data abort vector %**{ADDR}**. Fields:

ADDR unsigned int

vector address.

ArchMsg.Warning.reserved_it_state

TAGS **UNPREDICTABLE** DISPLAY Execution with reserved IT state %**{ITSTATE}**. Fields:

ITSTATE unsigned int

reserved IT state.

ArchMsg.Warning.tlb_contents_unknown

Translation has been enabled without invalidating the TLB. DISPLAY Translation was enabled in the %**{PAS:(Secure|Non-secure|Root|Realm)}** EL%**{EL:d}** translation regime (vmid=%**{VMID:x}**) without invalidating %**{INVALIDITY:(|d-side |i-side |)}**entries since power-on: %**{INVALIDITY:(|D|I)}**TLB could contain **UNKNOWN** content. Fields:

EL unsigned int

Exception level of translation regime.

INVALIDITY unsigned int

Which TLB may be invalid (if TLBs are separate).

NON_SECURE bool

Whether the translation regime is non-secure.

PAS unsigned int

Physical address space of translation regime.

VMID unsigned int

Current VMID, if applicable.

ArchMsg.Warning.unknown_DLR_DSPSR

Exiting debug state with unknown DLR or DSPSR.

ArchMsg.Warning.unknown_ELR_SPSR

Returning from debug exception with unknown ELR or SPSR.

ArchMsg.Warning.unpred_FAR_on_LRCPC3_non_atomic_mem_access

TAGS **UNPREDICTABLE** Unpredictable FAR/FSTATUS on LRCPC3 pair instructions resulting in two separate memory accessesDISPLAY Unpredictable FAR on LRCPC3 non atomic memory access at %**{ADDR1}** with FSTATUS %**{FSTATUS1}** or %**{ADDR2}** with FSTATUS %**{FSTATUS2}**. Fields:

ADDR1 unsigned int

Possible FAR address 1.

ADDR2 unsigned int

Possible FAR address 2.

FSTATUS1 unsigned int

Possible fault status value 1.

FSTATUS2 unsigned int

Possible fault status value 2.

ArchMsg.Warning.unpred_async_tag_check_fail_on_sve_speculative_load_do_update_ffr

Unpredictable set to FALSE of the FFR element associated to SVE speculative load with an async tag check fail TAGS **UNPREDICTABLE**.

ArchMsg.Warning.unpredictable_a32_breakpoint

TAGS **UNPREDICTABLE** DISPLAY Unpredictable breakpoint on a A32 instruction at %{ADDR1} Offset %{OFFSET} BAS %{BAS} for a %{IS_ADDRESS_MISMATCH_BREAKPOINT:(Mismatch|Match)} Breakpoint number %{N:d} Breakpoint address %{ADDR2}. Fields:

ADDR1 unsigned int

Address.

ADDR2 unsigned int

Breakpoint address.

BAS unsigned int

BAS field.

IS_ADDRESS_MISMATCH_BREAKPOINT bool

Is a Mismatch breakpoint, not a Match breakpoint.

N unsigned int

Breakpoint number.

OFFSET unsigned int

Always has a value of zero.

ArchMsg.Warning.unpredictable_banked_register_access

TAGS **UNPREDICTABLE** DISPLAY Access to banked register (R=%{R:d} SYSm=%{SYSM:2x}) is unpredictable. Fields:

R unsigned int

R bit.

SYSM unsigned int

SYSm.

ArchMsg.Warning.unpredictable_ccfail_undef

TAGS **UNPREDICTABLE** DISPLAY An instruction %{OPCODE} at VA=%{VADDR} failed its condition codes check but would otherwise have caused a non-data-dependent trap or undefined exception. Behaviour is implementation-defined choice of exception or nop. Fields:

OPCODE unsigned int

opcode of the instruction.

VADDR unsigned int

Virtual address of the instruction being executed.

ArchMsg.Warning.unpredictable_ici_bits

TAGS **UNPREDICTABLE** DISPLAY ICI bits do not apply or are architecturally unpredictable for instruction.

ArchMsg.Warning.unpredictable_t32_breakpoint

TAGS **UNPREDICTABLE** DISPLAY Unpredictable breakpoint on a T32 instruction at %{ADDR1} Offset %{OFFSET} BAS %{BAS} for a %{IS_ADDRESS_MISMATCH_BREAKPOINT:(Mismatch|Match)} Breakpoint number %{N:d} Breakpoint address %{ADDR2}. Fields:

ADDR1 unsigned int

Address.

ADDR2 unsigned int

Breakpoint address.

BAS unsigned int

BAS field.

IS_ADDRESS_MISMATCH_BREAKPOINT bool

Is a Mismatch breakpoint, not a Match breakpoint.

N unsigned int

Breakpoint number.

OFFSET unsigned int

Indicates which halfword of a 32-bit Thumb instruction the breakpoint is set on. 0 for the first halfword or if the instruction is 16 bits wide, or 1 for the second halfword.

ArchMsg.Warning.unpredictable_t32_vector_catch

TAGS **UNPREDICTABLE** DISPLAY Unpredictable Vector catch exception on a T32 instruction at %{ADDR} Offset %{OFFSET}. Fields:

ADDR unsigned int

Address.

OFFSET unsigned int

Indicates which halfword of a 32-bit Thumb instruction the vector catch exception is set on. 0 for the first halfword or if the instruction is 16 bits wide, or 1 for the second halfword.

ArchMsg.Warning.unpredictable_watchpoint_far

TAGS **UNPREDICTABLE** DISPLAY The value for FAR/EDWAR for hitting this watchpoint may be anywhere in the range %{LOWER_BOUND} to %{UPPER_BOUND}. Fields:

LOWER_BOUND unsigned int

the lowest address accessed by the instruction that triggered the watchpoint.

UPPER_BOUND unsigned int

the highest watchpointed address accessed by that instruction.

ArchMsg.Warning.unpredictable_watchpoint_far_isscmp

DISPLAY The value for FAR/EDWAR for hitting this watchpoint may be anywhere in the range %{LOWER_BOUND} to %{UPPER_BOUND}. Fields:

LOWER_BOUND unsigned int

the lowest address accessed by the instruction that triggered the watchpoint.

UPPER_BOUND unsigned int

the highest watchpointed address accessed by that instruction.

ArchMsg.Warning.warning_AdvSIMDExpandImmUnexpectedZero

TAGS **UNPREDICTABLE** DISPLAY AdvSIMDExpandImm may treat this immediate value as **UNPREDICTABLE**.

ArchMsg.Warning.warning_ConditionalSMC

TAGS **UNPREDICTABLE** DISPLAY SMC instruction has **UNPREDICTABLE** effects when conditional and when combined with traps.

ArchMsg.Warning.warning_TLBID_out_of_range

TLBID out of range in TLBI according to the configuration of NIS, NVIS, NOS and NVOS. Fields:

NIS unsigned int

NIS.

NOS unsigned int

NOS.

NVIS unsigned int

NVIS.

NVOS unsigned int

NVOS.

TLBID unsigned int

The TLBID value used in the TLBI instruction.

ArchMsg.Warning.warning_TTL_or_TG_mismatches_current_translation_granule

TTL or TG mismatches current translation granule in TLBI VA/RVA/IPA/RIPA. Fields:

CURRENT_TG unsigned int

Current TG as indicated by TCR_ELx or VTCR_EL2.

TLBI_TG unsigned int

TG (or TTL[3:2]) used in TLBI instruction.

ArchMsg.Warning.warning_access_crosses_page_boundary

TAGS **UNPREDICTABLE** DISPLAY Access crosses page boundary.

ArchMsg.Warning.warning_access_crosses_page_boundary_has_fault

TAGS **UNPREDICTABLE** DISPLAY Access that crosses page boundary has fault(s).

ArchMsg.Warning.warning_access_crosses_page_boundary_has_fault_on_both_pages

TAGS **UNPREDICTABLE** DISPLAY Access that crosses page boundary has a fault on each page.

Fields:

ADDR1 unsigned int

FAR address on first page.

ADDR2 unsigned int

FAR address on second page.

FSTATUS1 unsigned int

Fault status on first page.

FSTATUS2 unsigned int

Fault status on second page.

ArchMsg.Warning.warning_access_crosses_page_boundary_spanning_different_memory

TAGS **UNPREDICTABLE** DISPLAY Access at address %{ADDR} crosses page from MEM_TYPE %{MEMTYPE_PAGE1} to %{MEMTYPE_PAGE2}. Fields:

ADDR unsigned int

address of access crossing page.

MEMTYPE_PAGE1 signed int

First Page's memory type (0-Normal, 1-Non-Normal).

MEMTYPE_PAGE2 signed int

Second Page's memory type (0-Normal, 1-Non-Normal).

ArchMsg.Warning.warning_access_wraps_around_memory

TAGS **UNPREDICTABLE** DISPLAY Access wraps around memory in %{WIDTH:d}bit mode.

Fields:

WIDTH unsigned int

Address width.

ArchMsg.Warning.warning_bcr_linking_status

TAGS **UNPREDICTABLE** DISPLAY DBGBCR.{BT2,BT} combination gives unpredictable behaviour.

ArchMsg.Warning.warning_bcr_mask_reserved

TAGS **UNPREDICTABLE** DISPLAY DBGBCR_ELn.MASK is set to a reserved value.

ArchMsg.Warning.warning_bvr_masked_bit_not_zero

TAGS **UNPREDICTABLE** DISPLAY DBGBVRn_EL1.VA contains a bit masked by DBGBCRn_EL1.MASK that is not zero.

ArchMsg.Warning.warning_ccsidr_unimplemented_level

TAGS **UNPREDICTABLE** DISPLAY CCSIDR read while CSSELR %{CSSELR:x} points to an unimplemented cache level. Fields:

CSSELR unsigned int

current effective value of CSSELR.

ArchMsg.Warning.warning_change_to_ns_when_tge_set

TAGS **UNPREDICTABLE** DISPLAY Attempting to change NS to 1 when HCR.TGE = 1.

ArchMsg.Warning.warning_contiguous_bit_check_abort

TAGS **UNPREDICTABLE** DISPLAY An abort occurred whilst attempting to check TLB entry is contiguous at %{ENTRY_ADDR}. Fields:

ENTRY_ADDR unsigned int

address of conflicting TLB entry.

ArchMsg.Warning.warning_contiguous_bit_error

TAGS **UNPREDICTABLE** DISPLAY TLB entry at %{ENTRY_ADDR} was not contiguous with entry at %{CONTIG_ADDR} Contents are %{ENTRY_DATA} but expected %{CONTIG_DATA}. Fields:

CONTIG_ADDR unsigned int

address of the TLB first read, with which this entry is expected to be contiguous.

CONTIG_DATA unsigned int

expected contents based on the entry at CONTIG_ADDR.

ENTRY_ADDR unsigned int

address of conflicting TLB entry.

ENTRY_DATA unsigned int

contents of conflicting TLB entry.

ArchMsg.Warning.warning_contiguous_bit_error_D128

TAGS **UNPREDICTABLE** DISPLAY TLB entry at %{ENTRY_ADDR} was not contiguous with entry at %{CONTIG_ADDR} Contents are %{ENTRY_DATA_HI} %{ENTRY_DATA_LO} but expected %{CONTIG_DATA}. Fields:

CONTIG_ADDR unsigned int

address of the TLB first read, with which this entry is expected to be contiguous.

CONTIG_DATA unsigned int

expected contents based on the entry at CONTIG_ADDR.

ENTRY_ADDR unsigned int

address of conflicting TLB entry.

ENTRY_DATA_HI unsigned int

Higher bit contents of conflicting TLB entry.

ENTRY_DATA_LO unsigned int

Lower bit contents of conflicting TLB entry.

ArchMsg.Warning.warning_cp10_cp11_mismatch

TAGS **UNPREDICTABLE** DISPLAY Unpredictable configuration of CP10 and CP11 controls at EL
%{EL:d}. Fields:

EL unsigned int

EL owning the control.

ArchMsg.Warning.warning_cp10_cp11_reserved_value

TAGS **UNPREDICTABLE** DISPLAY Unpredictable configuration of CPACR CP10 (%{CP10}) or
CP11 (%{CP11}). Fields:

CP10 unsigned int

CP10 access.

CP11 unsigned int

CP11 access.

ArchMsg.Warning.warning_csselr_level_out_of_range

TAGS **UNPREDICTABLE** DISPLAY CSSELR.Level out of range, written value %{LEVEL} with only
%{IMPLEMENTED:d} levels implemented. Fields:

IMPLEMENTED unsigned int

number of cache levels implemented.

LEVEL unsigned int

written value of CSSELR.Level.

TYPE enum

The type of cache selected by CSSELR.InD and TnD.

ArchMsg.Warning.warning_debug_flow_control_bits_not_obeyed

TAGS **UNPREDICTABLE** DISPLAY Flow control bits for %{REGISTER} were not obeyed, causing
%{ERROR}. Fields:

ERROR enum

type of error resulting.

REGISTER enum

register under consideration.

ArchMsg.Warning.warning_debug_register_access_during_reset

TAGS **UNPREDICTABLE** DISPLAY Unpredictable %{IS_WRITE:(read|write)} access to debug
register offset %{OFFSET} during reset. Fields:

IS_WRITE unsigned int

Write Not Read.

OFFSET unsigned int

Register Offset.

ArchMsg.Warning.warning_decode_cps_inconsistent_fields

TAGS **UNPREDICTABLE** DISPLAY The mode setting fields in the CPS instruction are inconsistent.

ArchMsg.Warning.warning_decode_invalid_state

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable in current exception-level/security state.

ArchMsg.Warning.warning_default_cacheable_mmu_on

TAGS **UNPREDICTABLE** DISPLAY Default cacheable is enabled (HCR.DC = 1) while MMU is enabled.

ArchMsg.Warning.warning_default_cacheable_vmmu_off

TAGS **UNPREDICTABLE** DISPLAY Default cacheable is enabled (HCR.DC = 1) while HCR.VM = 0.

ArchMsg.Warning.warning_deprecated_wvr_bit_2_set

TAGS **UNPREDICTABLE** DISPLAY DBGWVR%{N:d}_EL1=%{WVR:h} has bit 2 set, deprecated in ARMv8. Fields:

N unsigned int
index.

WVR unsigned int
DBGWVR.

ArchMsg.Warning.warning_eret_while_software_step_active_pending

Missing ISB between setting MDSCR_EL1.SS and ERET TAGS **UNPREDICTABLE**.

ArchMsg.Warning.warning_exclusive_to_non_normal

TAGS **UNPREDICTABLE** DISPLAY Exclusive Access to Non-Normal Memory %{ADDR}. Fields:

ADDR unsigned int
Address.

ArchMsg.Warning.warning_exclusive_to_non_writeback

TAGS **UNPREDICTABLE** DISPLAY Exclusive Access to Non-Writeback-Cacheable Memory %{ADDR}. Fields:

ADDR unsigned int
Address.

ArchMsg.Warning.warning_execute_from_device_memory

TAGS **UNPREDICTABLE** DISPLAY An attempt was made to execute from Device memory.

ArchMsg.Warning.warning_illegal_cpsr_mode

TAGS **UNPREDICTABLE** DISPLAY Writing an illegal or unimplement mode (%{NEW_MODE:h}) to CPSR was unpredictable. Fields:

NEW_MODE unsigned int
New value of CPSR.M.

ArchMsg.Warning.warning_illegal_srs_mode

TAGS **UNPREDICTABLE** DISPLAY Illegal or **UNPREDICTABLE** mode (%{MODE:h}) used for SRS instruction. Fields:

MODE unsigned int

mode for Banked SP.

ArchMsg.Warning.warning_implementation_defined_read_debug_register_in_SCS

Read Access to Debug registers from the processor is IMPLEMENTATION DEFINED TAGS IMP_DEF. Fields:

PPB_OFFSET unsigned int

debug register offset.

ArchMsg.Warning.warning_implementation_defined_sequential_security_transitions_supported

The behavior of sequential instruction fetches that cross from non-secure to secure memory and contain an SG instruction is **CONSTRAINED UNPREDICTABLE** TAGS **UNPREDICTABLE**.

ArchMsg.Warning.warning_implementation_defined_stack_limit_check_supported

It is **IMPLEMENTATION DEFINED** whether stack pointer limit checking is performed for this instruction TAGS IMP_DEF. Fields:

ADDRESS unsigned int

Address of instruction.

OPCODE unsigned int

instruction opcode.

ArchMsg.Warning.warning_implementation_defined_taggednonsharedmemorytype

TAGS IMP_DEF DISPLAY Memory attribute has Tagged, non shared memory type; it is **IMPLEMENTATION DEFINED** whether the memory location is treated as Tagged or non-Tagged.

ArchMsg.Warning.warning_implementation_defined_write_debug_register_in_SCS

Write Access to Debug registers from the processor is IMPLEMENTATION DEFINED TAGS IMP_DEF. Fields:

DATA unsigned int

data attempted to be written.

PPB_OFFSET unsigned int

debug register offset.

ArchMsg.Warning.warning_invalid_tcr_granule

TAGS **UNPREDICTABLE** DISPLAY TCR.TG{%TG_ID:(0|1)} bits have been set to a granule size {%REQUEST:d}K not implemented - using {%SUBSTITUTE:d}K. Fields:

REQUEST signed int

page size requested (or 0 for reserved).

SUBSTITUTE signed int

best guess available page size.

TG_ID bool

bits TG0 or TG1.

ArchMsg.Warning.warning_load_multiple_user_registers_from_user_mode

TAGS UNPREDICTABLE DISPLAY An LDM(user registers) instruction executed from user mode.

ArchMsg.Warning.warning_mte_in_mem_attr_not_supported

TAGS UNPREDICTABLE DISPLAY Memory attribute %{ATTR} has Tagged memory type but Armv8.5-MTE is not implemented. Fields:

ATTR unsigned int

Memory attribute.

ArchMsg.Warning.warning_reserved_breakpoint_state_match

TAGS UNPREDICTABLE DISPLAY %{IS_BREAKPOINT:(Watchpoint|Breakpoint)} programmed with a reserved combination of HMC, SSC and %{IS_BREAKPOINT:(PAC|PMC)}. Fields:

IS_BREAKPOINT unsigned int

Is a breakpoint, not a watchpoint.

ArchMsg.Warning.warning_shareability

TAGS UNPREDICTABLE DISPLAY Unpredictable: combination of the 1st and 2nd stages of translation is Normal Inner Non-Cacheable, Outer Non-Cacheable.

ArchMsg.Warning.warning_software_step_set_while_enabled

TAGS UNPREDICTABLE DISPLAY MDSCR_EL1.SS set to 1 while software step debug exceptions are enabled.

ArchMsg.Warning.warning_thumb_instruction_wraps_around_memory

TAGS UNPREDICTABLE DISPLAY Thumb instruction wraps around memory.

ArchMsg.Warning.warning_ttbr_sbz_bits_are_not_zero

TAGS UNPREDICTABLE DISPLAY %{TTBR}[%{MSB:d}:0] should be zero, but are not. Fields:

MSB unsigned int

x-1.

TTBR enum

which TTBR register.

ArchMsg.Warning.warning_unaligned_address_dbgdtrrx_write

TAGS UNPREDICTABLE DISPLAY Unaligned address %{ADDR} in X0/R0 when DBGDTRRX written in memory access mode. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unaligned_address_dbgdtrtx_read

TAGS UNPREDICTABLE DISPLAY Unaligned address %{ADDR} in X0/R0 when DBGDTRTX read in memory access mode. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unaligned_to_device

TAGS **UNPREDICTABLE** DISPLAY Unaligned Access to Device Memory %{ADDR}. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unaligned_to_strongly_ordered

TAGS **UNPREDICTABLE** DISPLAY Unaligned Access to Strongly Ordered Memory %{ADDR}. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unknown_sau_rnr

TAGS **UNPREDICTABLE** DISPLAY SAU_RNR was set to an unsupported value.

ArchMsg.Warning.warning_unpredictable_AIRCR_PRIS_and_BFHFNMINs

TAGS **UNPREDICTABLE** DISPLAY The effect of setting both AIRCR.BFHFNMINs and AIRCR.PRIS to 1 is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_AIRCR_VECTCLRACTIVE_when_not_in_debug

TAGS **UNPREDICTABLE** DISPLAY The effect of writing a 1 to AIRCR.VECTCLRACTIVE if the processor is not halted in Debug state is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_AIRCR_VECTRESET_and_SYSRESETREQ

TAGS **UNPREDICTABLE** DISPLAY When the processor is halted in Debug state, if a write to the register writes a 1 to both VECTRESET and SYSRESETREQ, the behavior is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_AIRCR_VECTRESET_when_not_in_debug

TAGS **UNPREDICTABLE** DISPLAY The effect of writing a 1 to AIRCR.VECTRESET if the processor is not halted in Debug state is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_AIRCR_incorrect_VKEY

TAGS **UNPREDICTABLE** DISPLAY The value 0x05FA must be written to AIRCR.VKEY, otherwise the register write is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_EXC_RETURN_Reserved_Bit

EXC_RETURN[23:7] are reserved with the special condition that all bits should be written as one. Values other than all 1s are **UNPREDICTABLE**. TAGS **UNPREDICTABLE**. Fields:

EXC_RETURN unsigned int

Special PC value.

SBO_MASK unsigned int

Which bits Should Be One to not UNPRED.

ArchMsg.Warning.warning_unpredictable_PMEVTYPEPn_ELO_VS_reserved_value

TAGS **UNPREDICTABLE** DISPLAY Unpredictable reserved value of PMEVTYPEn<n>_ELO.VS.

ArchMsg.Warning.warning_unpredictable_change_to_DEMCR_MON_STEP_at_insufficient_priority

TAGS UNPREDICTABLE DISPLAY The effect of changing DEMCR.MON_STEP at an execution priority that is lower than the priority of the DebugMonitor exception is **CONSTRAINED UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_change_to_priority_of_active_exception

TAGS UNPREDICTABLE DISPLAY Changing the priority of an active exception is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_clear_lspact

TAGS UNPREDICTABLE DISPLAY **UNPREDICTABLE** state on context switch. Fields:

DRVEXC_TAKEN unsigned int

Derived exception taken, not pended.

REACHED_LAST_STORE unsigned int

Only the last faulted.

ArchMsg.Warning.warning_unpredictable_dbg_exit_unaligned_dlr

TAGS UNPREDICTABLE DISPLAY Debug exit to AArch32 with DLR[0] set to 1. PC[0] can be set to 0 or to DLR[0].

ArchMsg.Warning.warning_unpredictable_exception_catch

TAGS UNPREDICTABLE DISPLAY Generation of an exception catch event is unpredictable.

ArchMsg.Warning.warning_unpredictable_exception_return_inconsistent_state

TAGS UNPREDICTABLE DISPLAY state on exception return is unpredictable.

ArchMsg.Warning.warning_unpredictable_exception_return_instruction

TAGS UNPREDICTABLE DISPLAY exception return instruction is unpredictable.

ArchMsg.Warning.warning_unpredictable_in_debug_state

TAGS UNPREDICTABLE DISPLAY Instruction is **UNPREDICTABLE** when executed in debug state.

ArchMsg.Warning.warning_unpredictable_mair_encoding

TAGS UNPREDICTABLE DISPLAY Memory attribute %{ATTR} has invalid MAIR encoding; this may lead to unpredictable results. Fields:

ATTR unsigned int

Memory attribute.

ArchMsg.Warning.warning_unpredictable_pmu_counter_access

TAGS UNPREDICTABLE DISPLAY Access to PMU counters from non-secure EL0 or EL1 is **UNPREDICTABLE** with MDCR_EL2.HPMN set to 0.

ArchMsg.Warning.warning_unpredictable_prioritization_breakpoint_match_vector_catch

TAGS UNPREDICTABLE DISPLAY Two events with the same priority occurred at the same instruction: Address Matching Vector Catch debug event, and Breakpoint debug event. It is constrained unpredictable which is taken.

ArchMsg.Warning.warning_unpredictable_stack_selection

TAGS UNPREDICTABLE DISPLAY **UNPREDICTABLE** state on context switch.

ArchMsg.Warning.warning_unpredictable_tsize_out_of_range

TAGS **UNPREDICTABLE** DISPLAY **%{TCR}.T%{TTBR:d}SZ (%{TSIZE:d} bits)** is out of range. Must be between 25 and **%{TSIZE_MAX:d}**. Fields:

TCR enum

Which TCR.

TSIZE signed int

Value of TSize.

TSIZE_MAX signed int

The maximum allowed value.

TTBR unsigned int

Which TTBR region.

ArchMsg.Warning.warning_unpredictable_unaligned_pc_as_base_register

TAGS **UNPREDICTABLE** DISPLAY The PC value must be word-aligned, otherwise the behavior of the instruction is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_unaligned_pop_stack

TAGS **UNPREDICTABLE** DISPLAY **CONSTRAINED UNPREDICTABLE** unaligned stack frame **%{FRAME_ADDR:x}** in exception-return. Fields:

FRAME_ADDR unsigned int

unaligned stack frame address.

ArchMsg.Warning.warning_unpredictable_unaligned_to_non_writeback

TAGS **UNPREDICTABLE** DISPLAY Unaligned Access to Non-Writeback-Cacheable Memory.

ArchMsg.Warning.warning_unpredictable_vmsa_memattrib

TAGS **UNPREDICTABLE** DISPLAY Unpredictable vmsa memory attribute with texcb value programmed to **%{TEXCB}** when tex remap is **%{TEX_REMAP}**. Fields:

TEXCB unsigned int

texcb value.

TEX_REMAP bool

use tex remap.

ArchMsg.Warning.warning_unpredictable_vtcr_t0sz_sl0

TAGS **UNPREDICTABLE** DISPLAY The combination of VTCR.TOSZ=**%{TOSZ:d}** and VTCR.SL0=**%{SL0:d}** is unpredictable. Fields:

SL0 signed int

VTCR.SL0, starting level for stage 2 translation table walks.

TOSZ signed int

VTCR.TOSZ, required input address range.

ArchMsg.Warning.warning_unpredictable_write_DHCSR

TAGS **UNPREDICTABLE** DISPLAY **UNPREDICTABLE** set of bit-changes in DHCSR. Fields:

NEW unsigned int

data attempted to be written.

OLD unsigned int

previous value.

ArchMsg.Warning.warning_unpredictable_write_SHCSR

TAGS UNPREDICTABLE DISPLAY UNPREDICTABLE setting a pending bit to 1 for an exception with priority >= execution priority.

ArchMsg.Warning.warning_unsupported_access_to_memory_mapped_register

TAGS UNPREDICTABLE DISPLAY Access to memory mapped register is unsupported.

ArchMsg.Warning.warning_user_jmcr_access

TAGS UNPREDICTABLE DISPLAY User %{WRITE:(write|read)} access to JMCR is unpredictable. Fields:

WRITE unsigned int

access is write.

ArchMsg.Warning.warning_wcr_mask_and_bas

TAGS UNPREDICTABLE DISPLAY DBGWCR_EL1.MASK is non-zero and BAS is not 0xff.

ArchMsg.Warning.warning_wcr_mask_reserved

TAGS UNPREDICTABLE DISPLAY DBGWCR_ELn.MASK is set to a reserved value.

ArchMsg.Warning.warning_wcr_non_configuous_bas

TAGS UNPREDICTABLE DISPLAY DBGWCR_EL1.BAS has non-contiguous set of ones.

ArchMsg.Warning.warning_wvr_masked_bit_not_zero

TAGS UNPREDICTABLE DISPLAY DBGWVRn_EL1.VA contains a bit masked by DBGWCRn_EL1.MASK that is not zero.

ArchMsg.Why.why_branch_target_exception

A branch target exception has occurred. DISPLAY branch target exception: target pc=%{PC}, target opcode=%{OPCODE}. PRIMARY KEY PC. Fields:

BTYPE unsigned int

pstate.BTYPE at time branch taken.

OPCODE unsigned int

opcode at branch target.

PC unsigned int

Virtual Address of branch target.

ArchMsg.Why.why_illegal_state

An illegal mode change has occurred. DISPLAY illegal mode change: %{MESSAGE}. Fields:

MESSAGE string

The message.

BRANCH_MISPREDICT

Simulating branch mispredict. Fields:

PC unsigned int

Origin address (or 0 if unavailable).

BRA_DIR

Direct branches, to immediate address. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISSET enum

The instruction set of the branch instruction.

IS_COND bool

Indicates if this is a conditional branch.

IS_HINTED bool

Whether the conditional instruction is hinted.

IS_ISB bool

Whether the instruction is ISB.

IS_LINK bool

Indicates if this is a branch with link.

PC unsigned int

The address of the branch instruction.

TARGET_ISSET enum

The instruction set after the branch.

TARGET_PC unsigned int

The address the instruction branches to.

BRA_INDIR

Indirect branches, perhaps to a register. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISSET enum

The instruction set of the branch instruction.

IS_COND bool

Indicates if this is a conditional branch.

IS_HINTED bool

Whether the conditional instruction is hinted.

IS_ISB bool

Whether the instruction is ISB.

IS_LINK bool

Indicates if this is a branch with link.

PC unsigned int

The address of the branch instruction.

TARGET_ISET enum

The instruction set after the branch.

TARGET_PC unsigned int

The address the instruction branches to.

BRA_INDIR_CCFAIL

Indirect branches fail, perhaps to a register.

BRA_RET_CCFAIL

Branch return fail, perhaps to a register.

CACHE_MAINTENANCE_OP

Cache Maintenance Operation. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

DATA unsigned int

Specified MVA or Set/way.

FUNCTION enum

Clean / Invalidate.

SCOPE enum

Affected region.

SIDE enum

Inst / Data.

CCFAIL

Conditional instruction condition check fail. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CCFAIL_UNC

Unconditional instruction condition check fail. Fields:

COND *enum*

The condition of the conditional instruction.

CORE_NUM *unsigned int*

Core number in a multi processor.

PC *unsigned int*

The address of the conditional instruction.

CCPASS

Conditional instruction condition check pass. Fields:

COND *enum*

The condition of the conditional instruction.

CORE_NUM *unsigned int*

Core number in a multi processor.

PC *unsigned int*

The address of the conditional instruction.

CCPASS_UNC

Unconditional instruction condition check pass. Fields:

COND *enum*

The condition of the conditional instruction.

CORE_NUM *unsigned int*

Core number in a multi processor.

PC *unsigned int*

The address of the conditional instruction.

CODE_CACHE_MAINT

Code cache maintenance. Fields:

END_INCL *unsigned int*

Inclusive end address for VA or PA requests.

NS *bool*

Non-secure world for PA.

START *unsigned int*

Start address for VA or PA requests.

TYPE *enum*

Request type.

COMPILE_BLOCK_END

Last instruction of basic block translated. Fields:

VADDR unsigned int

Address of next instruction after this basic block.

COMPILE_BLOCK_START

First instruction of basic block translated. Fields:

VADDR unsigned int

Address of first instruction in basic block.

COMPILE_INST

ARM instruction compiled. Fields:

DISASS string

Disassembly of the instruction.

ISSET enum

The instruction set of this instruction.

ITSTATE unsigned int

The ITSTATE current for the instruction.

OPCODE unsigned int

The opcode of the instruction.

PC unsigned int

The address of the instruction.

SIZE unsigned int

The size of the instruction in bytes.

CONTEXTIDR

Context ID Register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

NS bool

Secure or nonsecure banked register is accessed.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The new value written.

CONTEXT_SYNC

Called for every context synchronization event. Fields:

ARCH_CSE bool

Architectural Context Synchronization Event.

PC unsigned int

The address of the synchronization instruction.

SYSREG_SYNC bool

System register synchronization event.

CORE_ENDIAN

Core BE8 Big-Endian state changed. Fields:

BE8 bool

Core BE8 Big-Endian state.

CORE_INFO

Static processor attributes. Only triggered by a call to DumpState(). Fields:

ARCH_PROFILE enum

The architecture profile of the core.

CLUSTER_ID unsigned int

The cluster ID of this processor.

CORE_NUM unsigned int

The number of this core in an MP processor.

DETAILED_CORE_FEATURES enum

Detailed core features as a bitfield. Enum values are bitmasks to extract the info from the field.

FPU_VERSION enum

The VFP version implemented by the core.

MEM_ARCH enum

The memory architecture of the core.

MIDR_EL1 unsigned int

MIDR_EL1 value.

NUM_CORES unsigned int

The number of cores in this MP processor.

QUANTUM_SIZE unsigned int

The default quantum size of the core (in ticks).

SECURITY_FEATURES bool

Does the core have security features?.

CORE_LOADS

Processor load accesses. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ACQREL enum

Is this an acquire/release.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DATA unsigned int

The data read or written.

ELEMENT_SIZE unsigned int

Width of each element.

ENDIANNESS enum

Byte ordering of this access.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

LOCK enum

Normal, exclusive or locked access.

MECID unsigned int

Memory Encryption Context ID, if applicable.

MECID2 unsigned int

Memory Encryption Context ID of the second page, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PAGE_SIZE unsigned int

Page size as $\log_2(\text{size in bytes})$.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK, 2=Exclusive Failed, 3=Unsupported.

SIZE unsigned int

Width of the access in bytes. Only required if DATA is not traced.

TRANS bool

Is this a translated access.

VADDR unsigned int

The virtual address of the access.

CORE_REGS

Changes of the core registers R0 to R14. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number, 0 to 14.

MODE enum

Bank of the register accessed.

OLD_VALUE unsigned int

The old value overwritten.

PHYS_ID enum

The physical register accessed.

VALUE unsigned int

The new value written to the register.

CORE_REGS64

Changes of the core registers X0..X30, SP_ELn. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ID enum

The register number, 0-30 for X0-X30, ≥ 32 for SP_ELn.

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

CORE_REGS64_READ

General purpose AArch64 register has been read X0..X30, SP_ELn. Fields:

ID enum

The register number, 0-30 for X0-X30, >=32 for SP_ELn.

VALUE unsigned int

The value read from the register.

CORE_STALLED_TRANSITION

Detect when a core is prevented from progressing PChannel states. Fields:

CURRENT_STATE enum

Current power mode of core.

REASON string

The reason a transition was stalled, and what must be done to unblock it.

TARGET_STATE enum

Desired power mode of core after transition.

CORE_STORES

Processor store accesses. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ACQREL enum

Is this an acquire/release.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DATA unsigned int

The data read or written.

ELEMENT_SIZE unsigned int

Width of each element.

ENDIANNESS enum

Byte ordering of this access.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

LOCK enum

Normal, exclusive or locked access.

MECID unsigned int

Memory Encryption Context ID, if applicable.

MECID2 unsigned int

Memory Encryption Context ID of the second page, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PAGE_SIZE unsigned int

Page size as $\log_2(\text{size in bytes})$.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK, 2=Exclusive Failed, 3=Unsupported.

SIZE unsigned int

Width of the access in bytes. Only required if DATA is not traced.

TRANS bool

Is this a translated access.

VADDR unsigned int

The virtual address of the access.

CP14_READ

System Coprocessor register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP14 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc1 unsigned int

opcode 1.

opc2 unsigned int

opcode 2.

CP14_WRITE

System Coprocessor register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP14 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc1 unsigned int

opcode 1.

opc2 unsigned int

opcode 2.

CP15_READ

System Coprocessor register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP15 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc1 unsigned int

opcode 1.

opc2 unsigned int

opcode 2.

CP15_READ64

System Coprocessor register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP15 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc unsigned int

opcode 1.

CP15_WRITE

System Control Coprocessor register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP15 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc1 unsigned int

opcode 1.

opc2 unsigned int

opcode 2.

CP15_WRITE64

System Coprocessor register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP15 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc unsigned int

opcode 1.

CPSR

CPSR change. (In AArch64 PSTATE represented in SPSR format). Fields:

CORE_NUM unsigned int

Core number in a multi processor.

OLD_VALUE unsigned int

The old CPSR value.

UNKNOWN unsigned int

Bits within the register that have unknown value.

VALUE unsigned int

The new CPSR value.

CRYPTO_SPEC

Every crypto instruction speculatively executed.

CorePowerStateChange

Core power-up state has changed. DISPLAY Core powered %{PU:(up|down)}. Fields:

PU bool

Whether the core powered up.

DATA_CACHE_SET_ALLOC_TAGS

Set allocation tags from DC GVA instruction. Fields:

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

NSDESC unsigned int

The security state of the access.

PADDR unsigned int

The physical address of the access.

PAS enum

The physical address space of the page.

RESPONSE enum

0=Aborted, 1=OK.

SIZE unsigned int

The size of memory in bytes.

TAG unsigned int

The value of the tag being set.

TAGGING_TYPE enum

Type of Memory tagging (Physical or Virtual).

TAG_VADDR unsigned int

The virtual address of the Virtual Tag if Virtual tagging is used; otherwise 0.

VADDR unsigned int

Virtual address used for setting allocation tags.

DATA_CACHE_ZERO

Zero and invalidate cache line from DC ZVA instruction. Fields:

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

NSDESC unsigned int

The security state of the access.

PADDR unsigned int

The physical address of the access.

PADDR_TAG unsigned int

If tag checked, the physical address tag for the access.

PAS enum

The physical address space of the page.

RESPONSE enum

0=Aborted, 1=OK.

SIZE unsigned int

The size of zero'd memory in bytes.

TAG_CHECKED bool

Is this access tag checked.

VADDR unsigned int

Virtual address of the zero'd cache line.

DEBUG_EVENT

Hardware debug support event. Fields:

EVENT enum

Description of event.

VALUE unsigned int

data value.

DEBUG_TARGET_ISET

Hardware debug target ISA. Fields:

TARGET_ISET enum

The target instruction set of the debug state.

DMI_ALLOCATE

DMI allocated. Fields:

ACCESS_DIRECTION enum

READ / WRITE.

PADDR unsigned int

The physical address of the DMI.

PTR unsigned int

DMI range pointer.

SIDE enum

DSIDE / ISIDE.

SIZE unsigned int

The size of DMI.

VADDR unsigned int

The virtual address of the DMI..

DMI_HIT

DMI hit, access used DMI (only fire for small DMI regions i.e. less than a full page). Fields:

ACCESS_DIRECTION enum

READ / WRITE.

PADDR unsigned int

The physical address of the DMI.

SIDE enum

DSIDE / ISIDE.

SIZE unsigned int

The size of the DMI.

VADDR unsigned int

The virtual address of the DMI..

DMI_REVOKE

DMI revoked. Fields:

ACCESS_DIRECTION enum

READ / WRITE.

PADDR unsigned int

The physical address of the DMI.

SIDE enum

DSIDE / ISIDE.

SIZE unsigned int

The size of DMI.

VADDR unsigned int

The virtual address of the DMI..

END_COMPILE

Compilation end. Fields:

END_OF_PAGE_COUNT unsigned int

Number of basic blocks exited due to page boundary since START_COMPILE.

FETCHFAIL_COUNT unsigned int

Number of basic blocks exited due to ifetch failure START_COMPILE.

INST_COUNT unsigned int

Number of instructions compiled since START_COMPILE.

NONSEQ_COUNT unsigned int

Number of basic blocks exited due to non-sequential instructions since START_COMPILE.

PAGE_STRADDLE_COUNT unsigned int

Number of basic blocks exited due to unaligned instructions crossing page since START_COMPILE.

ESB

Error Synchronization Barrier hit. Fields:

ASYNC bool

ESB is async.

IMPLICIT bool

ESB is implicit.

EXCEPTION

Exception taken. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ESR unsigned int

The value assigned to ESR when exception is taken.

ESR_EC unsigned int

The Value assigned to ESR.EC when exception is taken.

IS_PHYSICAL bool

Physical or Virtual exception.

IS_TRAP bool

Exception coming from a trap.

LR unsigned int

The value assigned to the link register.

PC unsigned int

The location where the exception occurred.

PREFERRED_RETURN unsigned int

The preferred return address for the exception.

TAKEN_LOCALLY bool

Whether exception is taken locally(same exception level).

TARGET_ISET enum

The instruction set of the exception handler code.

TARGET_PC unsigned int

The address the exception branches to.

VECTOR enum

The exception vector.

VECTOR_OFFSET unsigned int

The offset of an exception vector from the base address.

EXCEPTION_END

Every exception completed.

EXCEPTION_INFO

Trace human-readable reason of the current exception for data/instruction aborts, floating point exceptions and unknown exceptions. Fields:

DESCRIPTION string

Reason for current exception.

ESR_EC unsigned int

The value of ESR.EC when exception is taken.

EXCEPTION_RAISE

Every exception raised.

EXCEPTION_RETURN

Branches on leaving exception. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISET enum

The instruction set of the branch instruction.

IS_COND bool

Indicates if this is a conditional branch.

IS_HINTED bool

Whether the conditional instruction is hinted.

IS_ISB bool

Whether the instruction is ISB.

IS_LINK bool

Indicates if this is a branch with link.

PC unsigned int

The address of the branch instruction.

TARGET_ISET enum

The instruction set after the branch.

TARGET_PC unsigned int

The address the instruction branches to.

EXCEPTION_RETURN_PREBRANCH

Exception return event that must occur before PC and processor state have updated.

EXCEPTION_START

Every exception started.

FIQ_TAKEN

FIQ taken exception.

FP_EXCEPTIONS_SIGNALLED

Trace which FP exceptions were signalled as a part of the FP operations performed by an instruction. Fields:

FP_EXCEPTIONS_SIGNALLED_MASK unsigned int

Indicates which FP exceptions were signalled by an instruction. This uses the same encoding as the cumulative floating-point exception bitfields in FPSR.

FP_STATE

Floating point state. Fields:

FPSR unsigned int

FPSR register value.

MASK unsigned int

The mask indicating updated cumulative bits.

FREQ_CHANGED

Core frequency changed. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The instruction count of this core.

NEW_FREQ signed int

The new frequency of this core (expressed in Hz).

OLD_FREQ signed int

The old frequency of this core (expressed in Hz).

HLT

HLT instruction occurred. Fields:

IMM16 unsigned int

The 16-bit immediate value encoded in HLT instruction.

INST

Every instruction executed. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DEBUG_STATE bool

The instruction is executed in debug state.

DISASS string

Disassembly of instruction.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISSET enum

The current instruction set.

ITSTATE unsigned int

The current ITSTATE.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

MODE enum

The mode the core is in.

NS unsigned int

The core's non-secure bit.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

OPCODE unsigned int

The opcode of the instruction.

PADDR unsigned int

The physical address of the instruction.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the instruction.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the page.

PC unsigned int

The address of the instruction.

SECURITY_STATE enum

The core's security state.

SIZE unsigned int

The size of the instruction in bytes.

INST_END

Every instruction completed.

INST_START

Every instruction started. Fields:

DEBUG_STATE bool

The instruction is executed in debug state.

ISET enum

The current instruction set.

MODE enum

The mode the core is in.

NS enum

The current Secure State.

PC unsigned int

The address of the conditional instruction.

SECURITY_STATE enum

The current Security State.

INST_STRADDLE

Instruction straddles boundary. Fields:

DEBUG_STATE bool

The instruction is executed in debug state.

ISET enum

The current instruction set.

MODE enum

The mode the core is in.

NS enum

The current Secure State.

PC unsigned int

The address of the conditional instruction.

SECURITY_STATE enum

The current Security State.

IRQ_TAKEN

IRQ taken exception.

ISB

ISB instruction occurred. Fields:

PC unsigned int

The address of the ISB instruction.

LOAD_MULTIPLE_REGS_LIST

List of destination registers (not source) of a multiple registers load instruction. Fields:

REG string

Register to add into the list.

LOCAL_MONITOR

Local monitor activity. Fields:

PADDR unsigned int

Local Monitor Address.

State enum

State of the monitor (Open/Exclusive).

MAPPING_CREATE

A memory mapping has been created. This is an implementation detail of the model and so may change without notice. Fields:

ASID unsigned int

The ASID used during translation, or 0 if no ASID was used.

IS_VALID bool

Whether the mapping is valid or not.

PADDR unsigned int

Base physical address of region, or 0 if not known.

SIDE enum

I-side or D-side.

SIZE unsigned int

Size of the page, or 0 if not known.

VADDR unsigned int

Base virtual address of region.

VMID unsigned int

The VMID used during translation, or 0 if no VMID was used.

MAPPING_DESTROY

A memory mapping has been destroyed. This is an implementation detail of the model and so may change without notice. Fields:

ASID unsigned int

The ASID used during translation.

CAUSE enum

The cause of invalidation.

PADDR unsigned int

Base physical address of page.

SIDE enum

I-side or D-side.

SIZE unsigned int

Size of the page.

VADDR unsigned int

Base virtual address of page.

VMID unsigned int

The VMID used during translation.

MEMMAP_DEBUG_READ

Memory mapped reads to the debug registers. Fields:

ADDR unsigned int

address.

CORE_NUM unsigned int

Core number in a multi processor.

EXT bool

Whether access is from an external device (such as the DAP).

NS enum

Secure state of the access.

REG_NAME string

Name of the debug register accessed.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

MEMMAP_DEBUG_WRITE

Memory mapped writes to the debug registers. Fields:

ADDR unsigned int

address.

CORE_NUM unsigned int

Core number in a multi processor.

EXT bool

Whether access is from an external device (such as the DAP).

NS enum

Secure state of the access.

REG_NAME string

Name of the debug register accessed.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The new value written.

MMU_D128_TTB_READ

This event is triggered by reads caused by a D128 translation table walk. Fields:

ABORTED bool

Was the walk successful.

DATA_HIGH unsigned int

The data upper bits read.

DATA_LOW unsigned int

The data lower bits read.

IPA unsigned int

For stage 1, the IPA of the read.

LEVEL unsigned int

Translation table level.

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

OUTADDR unsigned int

The output physical address.

PADDR unsigned int

The physical address of the read.

PASmem enum

Physical Address Space of the accessed memory.

PASreq enum

Physical Address Space of the requested lookup.

REGIME_EL enum

Entry matches in this translation regime.

REQUESTER enum

The requester of this table walk.

SIDE enum

Inst / Data.

STAGE unsigned int

Translation stage.

MMU_D128_TTB_WRITE

This event is triggered by writes caused by a D128 translation table walk. Fields:

ABORTED bool

Was the walk successful.

DATA_HIGH unsigned int

The data upper bits written.

DATA_LOW unsigned int

The data lower bits written.

IPA unsigned int

For stage 1, the IPA of the write.

IS_HACDBS_ACCESS bool

If the access was due to HACDBS_CLEAN.

LEVEL unsigned int

Translation table level.

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

OUTADDR unsigned int

The output physical address.

PADDR unsigned int

The physical address of the write.

PASmem enum

Physical Address Space of the accessed memory.

PASreq enum

Physical Address Space of the requested lookup.

REGIME_EL enum

Entry matches in this translation regime.

SIDE enum

Inst / Data.

STAGE unsigned int

Translation stage.

MMU_TRANS

Address translation information. Fields:

ASID unsigned int

Address space identifier.

CORE_NUM unsigned int

Core number in a multi processor.

Hyp bool

Entry matches in Hyp state only.

INNERCACHE_RA bool

Is the inner cache allocate on read.

INNERCACHE_TRANSIENT bool

Is the inner write-through transient.

INNERCACHE_TYPE enum

Inner Caching scheme (NC/MB/WA).

INNERCACHE_WA bool

Is the inner cache allocate on write.

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MEMTYPE enum

Memory type.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC enum

Is secure side supposed to access secure or nonsecure memory.

OUTERCACHE_RA bool

Is the outer cache allocate on read.

OUTERCACHE_TRANSIENT bool

Is the outer write-through transient.

OUTERCACHE_TYPE enum

Outer Caching scheme (NC/MB/WA).

OUTERCACHE_WA bool

Is the outer cache allocate on write.

PADDR unsigned int

Physical address of the access.

PAGESIZE unsigned int

Page size as $\log_2(\text{size})$.

PAS enum

PAS supposed to be accessed is secure or nonsecure memory or Root/Realm if RME is implemented.

SH enum

Shareability.

SIDE enum

Inst / Data.

STAGE1_PERM unsigned int

Stage 1 permissions, [5:3] Privileged access (XWR) [2:0] User access (XWR).

STAGE2_PERM unsigned int

Stage 2 permission mask [5:3] Privileged access (XWR) [2:0] User access (XWR).

VADDR unsigned int

Virtual address of the access.

VMID unsigned int

Virtual machine identifier.

XS bool

XS Attribute.

nG enum

Flag indicating whether ASID will be matched.

MMU_TTB_READ

This event is triggered by reads caused by a translation table walk. Fields:

ABORTED bool

Was the walk successful.

DATA unsigned int

The data read.

DESC_KIND enum

The encoding scheme of the descriptors used for this walk.

IPA unsigned int

For stage 1, the IPA of the read.

LEVEL unsigned int

Translation table level.

LPAE bool

Is this for an LPAE translation (*DEPRECATED*, use DESC_KIND).

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSmem bool

Non secure memory access.

NSreq bool

Non secure request.

OUTADDR unsigned int

The output physical address.

PADDR unsigned int

The physical address of the read.

PASmem enum

Physical Address Space of the accessed memory.

PASreq enum

Physical Address Space of the requested lookup.

REGIME_EL enum

Entry matches in this translation regime.

REQUESTER enum

The requester of this table walk.

SIDE enum

Inst / Data.

STAGE unsigned int

Translation stage.

MMU_TTB_WRITE

This event is triggered by writes caused by a translation table walk. Fields:

ABORTED bool

Was the walk successful.

DATA unsigned int

The data written.

DESC_KIND enum

The encoding scheme of the descriptors used for this walk.

IPA unsigned int

For stage 1, the IPA of the write.

IS_HACDBS_ACCESS bool

If the access was due to HACDBS_CLEAN.

LEVEL unsigned int

Translation table level.

LPAE bool

Is this for an LPAE translation (*DEPRECATED*, use DESC_KIND).

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSmem bool

Non secure memory access.

NSreq bool

Non secure request.

OUTADDR unsigned int

The output physical address.

PADDR unsigned int

The physical address of the write.

PASmem enum

Physical Address Space of the accessed memory.

PASreq enum

Physical Address Space of the requested lookup.

REGIME_EL enum

Entry matches in this translation regime.

SIDE enum

Inst / Data.

STAGE unsigned int

Translation stage.

MODE_CHANGE

Mode change. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

MODE enum

The new mode.

NON_SECURE enum

The core's new non-secure bit.

OLD_MODE enum

The old mode.

SECURITY_STATE enum

The new security state.

NOT_TAG_CHECKED_INST

MTE Tag not checked for current instruction.

PERIODIC

Called for every quantum. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The instruction count of this CPU.

PC unsigned int

The address of the next instruction to be executed on this CPU.

PMU_COUNTER_OVERFLOW

PMU counter overflow. Fields:

COUNTER_GROUP enum

Counter group.

COUNTER_RANGE enum

Counter range.

EVENT_ID unsigned int

Event Identifier, field is valid only for Event counter group.

INDEX unsigned int

Counter index (as selected by PMSELR), field is valid only for Event/Cycle counter groups.

INTERRUPT bool

Is interrupt enabled on overflow for this counter.

POTINDEX_UPDATE

Value of POT index for current pc changed. Fields:

NEW_POTINDEX unsigned int

New POTIndex.

OLD_POTINDEX unsigned int

Old POTIndex.

PREFETCH_MEMORY64

Prefetch from PRFM or PRFUM instructions. Fields:

PRFOP enum

Prefetch hint.

VADDR unsigned int

Virtual address of the location that should be prefetched.

PRELOAD64_REQUEST

Preload requests from PRFM or PRFUM instructions. Fields:

PRFOP enum

Prefetch hint.

VADDR unsigned int

Virtual address of the location that should be prefetched.

PRELOAD_DATA

Data preload from PLD instruction. Fields:

VADDR unsigned int

Virtual address of the data that should be preloaded.

PRELOAD_INST

Instruction preload from PLI instruction. Fields:

VADDR unsigned int

Virtual address of the instruction that should be preloaded.

PRE_CORE_LOAD

Trace just before a core load. Fields:

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

PRE_CORE_STORE

Trace just before a core store. Fields:

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

PRE_TTB_READ

This event is triggered before reads caused by a translation table walk. Fields:

ADDR unsigned int

The physical address of the read.

PSTATE

PSTATE fields update. Fields:

ALLINT unsigned int

All interrupt mask bit.

BTYPE unsigned int

Branch target identification bit.

CORE_NUM unsigned int

Core number in a multi processor.

DAIF unsigned int

Exception mask bits.

DIT unsigned int

Data Independent Timing bit.

EL unsigned int

Current Exception level field.

EXLOCK unsigned int

Exception return state lock bit.

IL unsigned int

Illegal Execution state bit.

NZCV unsigned int

Condition flags.

PACM unsigned int

PACM bit.

PAN unsigned int

Privileged Access Never state bit.

PM unsigned int

PMU exception mask bit.

PPEND unsigned int

PMU exception pending bit.

SM unsigned int

StreamingMode status bit.

SP unsigned int

Stack pointer register selection bit.

SS unsigned int

Software Step bit.

SSBS unsigned int

Speculative Store Bypass Safe bit.

TCO unsigned int

Tag Check Override bit.

TOUCHED unsigned int

Mask of the fields that have been touched since this trace has last been emitted. Note that touched does not necessarily mean the value changed. The index of the field is the bit index in the mask plus the index of this field plus one.

UAO unsigned int

User Access Override bit.

UINJ unsigned int

Undefined Instruction exceptions bit.

ZA unsigned int

ZA storage and ZT0 register enable bit.

nRW unsigned int

Current Execution state bit.

PSTATE_PACTIVE_OPMODE_TRANSITION

Trace pactive and Opmode mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Operating mode after the transition.

OLD_STATE enum

Operating mode before the transition.

PSTATE_PACTIVE_OPMODE_TRANSITION_DENIED

Trace the denied pactive and Opmode mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Operating mode after the transition.

OLD_STATE enum

Operating mode before the transition.

REASON string

Reason for the transition being denied.

PSTATE_PACTIVE_POWER_TRANSITION

Trace pactive and power mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Power mode after the transition.

OLD_STATE enum

Power mode before the transition.

PSTATE_PACTIVE_POWER_TRANSITION_DENIED

Trace the denied pactive and power mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Power mode after the transition.

OLD_STATE enum

Power mode before the transition.

REASON string

Reason for the transition being denied.

PSTATE_PACTIVE_THREAD_OPMODE_TRANSITION

Trace pactive and Opmode mode transitions for threads on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Operating mode after the transition.

OLD_STATE enum

Operating mode before the transition.

QUANTUM_START

Called at the start of every quantum.

RANGE_PRELOAD_REQUEST

Preload requests from RPRFM. Fields:

COUNT unsigned int

Number of blocks of data to access.

LENGTH unsigned int

Number of contiguous bytes to be accessed in each block.

OP unsigned int

Range prefetch operation .

REUSE unsigned int

Maximum number of bytes to be accessed before next RPRFM .

STRIDE signed int

Stride to advance after accessing LENGTH bytes.

VADDR unsigned int

Base virtual address of range prefetch request.

RUN_STATE

Run state transition. Fields:

INST_COUNT unsigned int

Ticks count at point of transition.

NEW enum

New run state.

OLD enum

Old run state.

SEMIHOSTING_CALL

Call of a semihost function occurred. Fields:

PC unsigned int

The program counter after the semihosting call.

SEMIHOSTING_PRECALL

About to call semihost. Fields:

EL2 bool

Use EL2 translation regime.

NS bool

Is Non-Secure.

REG_WIDTH unsigned int

The current register width in bytes.

SIGNAL

External signal state change. Fields:

SIGNAL enum

Signal that changed.

STATE bool

Signal asserted state.

SOFTWARE_STEP

Return the Debug Software Step state. Fields:

STATE **enum**

Software Step state.

SPSR

SPSR change. Fields:

CORE_NUM **unsigned int**

Core number in a multi processor.

MODE **enum**

Which of the banked SPSR registers is written.

OLD_VALUE **unsigned int**

The old SPSR value.

VALUE **unsigned int**

The new SPSR value.

START_COMPILE

Compilation started. Fields:

VADDR **unsigned int**

Instruction where compilation begins.

STG_MULTIPLE_MICRO_OPS_ADDRESS_RANGE

Address range for MTE stores with multiple uops. Fields:

ADDR_END **unsigned int**

Final address.

ADDR_START **unsigned int**

Starting address.

STORE_MULTIPLE_REGS_ADDRESS_RANGE

Address range for stores with multiple registers. Fields:

ADDR_END **unsigned int**

Final address.

ADDR_START **unsigned int**

Starting address.

SUBOPTIMAL_LDST_RETIRED

suboptimal load/store. Fields:

ACCESS_SIZE **unsigned int**

Log2 of the access width in bytes used for alignment checking (1: half-word, 2: word ...).

OPTIMAL_ALIGN_SELECTOR unsigned int

Selector to decide the optimal alignment size (1: access width, 2: 4-bytes, 3: 8-bytes, ..., 12: 4-kbytes).

VADDR unsigned int

The virtual address of the access.

SYNC

Called for every synchronization, that is every quantum break. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The instruction count of this CPU.

LOCAL_QUANTUM unsigned int

The local quantum of this CPU.

LOCAL_TIME unsigned int

The local time of this CPU.

SYNC_REASON enum

Reason for sync.

SYSCALL

System call instruction executed. Fields:

IMM unsigned int

Immediate value of the system call instruction.

TYPE enum

System call type.

VADDR unsigned int

Instruction that caused the system call.

SYSREG_READ128

System Coprocessor 128 bit register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

REGNUM enum

Internal register number.

REGNUM_PHYS enum

Internal register number physically accessed (e.g. in presense of VHE redirection).

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_READ64

System Coprocessor register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REGNUM enum

Internal register number.

REGNUM_OPERAND enum

Register number corresponding to the instruction operand.

REGNUM_PHYS enum

Internal register number physically accessed (e.g. in presense of VHE redirection).

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_UPDATE128

Triggers when the system updates a register. Fields:

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG enum

Register number.

UNKNOWN unsigned int

Bits of the register which became unknown.

VALUE unsigned int

Value written to the register.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_UPDATE32

Triggers when the system updates a register. Fields:

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG enum

Register number.

UNKNOWN unsigned int

Bits of the register which became unknown.

VALUE unsigned int

Value written to the register.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_UPDATE64

Triggers when the system updates a register. Fields:

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG enum

Register number.

UNKNOWN unsigned int

Bits of the register which became unknown.

VALUE unsigned int

Value written to the register.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_WRITE128

System Coprocessor 128 bit register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

REGNUM enum

Internal register number.

REGNUM_PHYS enum

Internal register number physically accessed (e.g. in presense of VHE redirection).

UNDEF bool

The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_WRITE64

System Coprocessor register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REGNUM enum

Internal register number.

REGNUM_OPERAND enum

Register number corresponding to the instruction operand.

REGNUM_PHYS enum

Internal register number physically accessed (e.g. in presense of VHE redirection).

UNDEF bool

The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

UNALIGNED_LDST_RETIRED

Processor unaligned load/store. Fields:

ACCESS_SIZE unsigned int

Log2 of the access width in bytes used for alignment checking (1=>half-word, 2->word ...).

NUMBER_OF_BEATS unsigned int

Number of accesses (beats) in this burst (Total size in bytes is ACCESS_SIZE * NUMBER_OF_BEATS).

VADDR unsigned int

The virtual address of the access.

VFP_D_REGS

VFP/NEON D 64 bit register write. Fields:

ALIASING enum

Type of register aliasing used.

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a write to the corresponding bit in the register. For example, 0x1 would indicate a write to VALUE[0].

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

VFP_D_REGS_READ

VFP/NEON D 64 bit register read. Fields:

ID unsigned int

The register number.

VALUE unsigned int

The current value of the register.

VFP_Q_REGS

VFP/NEON Q 128 bit register write. Fields:

ALIASING enum

Type of register aliasing used.

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a write to the corresponding byte in the register. For example, 0x1 would indicate a write to VALUE[7:0].

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

VFP_Q_REGS_READ

VFP/NEON Q 128 bit register read. Fields:

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a read to the corresponding byte in the register. For example, 0x1 would indicate a read to VALUE[7:0].

VALUE unsigned int

The current value of the register.

VFP_SYS_REGS

Writes to the VFP/NEON units system registers. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ID enum

Which VFP system register is written.

MASK unsigned int

The mask indicating updated cumulative bits.

OLD_VALUE unsigned int

The register's old value overwritten.

VALUE unsigned int

The new value written to the VFP system register.

VFP_S_REGS

VFP/NEON S 32 bit register write. Fields:

ALIASING enum

Type of register aliasing used.

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a write to the corresponding bit in the register. For example, 0x1 would indicate a write to VALUE[0].

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

VFP_S_REGS_READ

VFP/NEON S 32 bit register read. Fields:

ID unsigned int

The register number.

VALUE unsigned int

The current value of the register.

VIRTUAL_TAG_ALLOCATED

Virtual tag information when allocated. Fields:

TAG_VADDR unsigned int

Virtual tag virtual address.

VADDR unsigned int

Data Virtual address.

VIRTUAL_TAG unsigned int

Virtual tag value.

WAYPOINT

Signals end of basic block execution. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ISSET enum

Origin instruction set.

IS_COND bool

Indicates if this is a conditional waypoint.

PC unsigned int

Origin address (or 0 if unavailable).

TAKEN bool

Indicates if this waypoint was taken.

TARGET unsigned int

Destination address.

TARGET_ISET enum

Destination instruction set.

WFE_END

WFE ended. Fields:

INST_COUNT unsigned int

Ticks count when leaving WFE.

WFE_EVENT_REGISTER

WFE event register status: set/clear, reason. Fields:

INST_COUNT unsigned int

Ticks count.

REASON enum

Reason for set/clear. Only REASON==0 (Cleared by WFE) clears the bit, all other reasons set it.

WFE_IGNORED

WFE ignored. Fields:

EVENT bool

This WFE was ignored because the event register was set.

INST_COUNT unsigned int

Ticks count when ignoring WFE.

TRAPPED bool

This WFE was trapped.

WFE_START

WFE entered. Fields:

INST_COUNT unsigned int

Ticks count when entering WFE.

WFI_END

WFI ended. Fields:

INST_COUNT unsigned int

Ticks count when leaving WFI.

WFI_IGNORED

WFI ignored. Fields:

DISABLED bool

This WFI was ignored because WFI is disabled.

INST_COUNT unsigned int

Ticks count when ignoring WFI.

REASON enum

specifies reason why WFI trace was ignored.

TRAPPED bool

This WFI was trapped.

WFI_START

WFI entered. Fields:

INST_COUNT unsigned int

Ticks count when entering WFI.

WFI_WAKEUP

WFI wakeup. Fields:

INST_COUNT unsigned int

Ticks count when WFI wakeup occurred.

REASON enum

Reason for wakeup.

2.32 ARM_Cortex-A76

This section describes the trace sources.

AA64_ASE_SVE_REGS

Modification of AArch64 Advanced SIMD or SVE Vector Registers (via SIMD or Scalar operations), SVE Predicate Registers and ZT0. Fields:

ID enum

The register identifier.

MASK unsigned int

Bitmask of the register to signal the modified bits in the VALUE field.

SM bool

Whether the PE is in Streaming Mode.

VALUE unsigned int

Value of AArch64 vector register.

ASYNC_MEMORY_FAULT

Context ID Register write. Fields:

FAULT unsigned int

Fault status present in ESR.ISS encoding for Serror in ARCH64 and present in DFSR encoding for external abort format in ARCH32.

PADDR unsigned int

Physical Address (or 0 if unavailable).

VADDR signed int

Virtual Address (or 0 if unavailable).

ATOMIC_END_ACCESS

Bus trace access for atomics. Fields:

ACCESS_FAIL bool

Memory access failed.

ADDR unsigned int

The virtual address of the access.

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

COMPARE_VALUE unsigned int

Compare value for CAS.

LOAD_VALUE unsigned int

Loaded values.

NSDESC unsigned int

The security state of the access.

OPERAND_VALUE unsigned int

Operation's operand.

OPERATION enum

Operation type.

PADDR unsigned int

The physical address of the access.

PRIV bool

Is this a privileged access?.

ATOMIC_START_ACCESS

Bus trace access for atomics. Fields:

ADDR unsigned int

The virtual address of the access.

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

COMPARE_VALUE unsigned int

Compare value for CAS.

NSDESC unsigned int

The security state of the access.

OPERAND_VALUE unsigned int

Operation's operand.

OPERATION enum

Operation type.

PADDR unsigned int

The physical address of the access.

PRIV bool

Is this a privileged access?.

ArchMsg.Error.exit_code

DISPLAY exit code: %{EXITCODE} component kind: %{KIND} reason: %{REASON}. Fields:

EXITCODE signed int

Exit code to be returned.

KIND string

Component kind that invoked the exit code trace.

REASON string

Reason why the exit code trace was invoked (optional).

ArchMsg.Warning.IgnoredWarmResetRequest

Event fires when software attempts to request warm reset through write to deprecated DBGPRCR.CWRR.

ArchMsg.Warning.RAS_ErrselrUnpredictableState

DISPLAY ERRSELR is in unpredictable state due to value greater than number_of_error_records written to it.

ArchMsg.Warning.RAS_UnpredictableStateWrite

DISPLAY %{REG_NAME} write ignored due to unpredictable ERRSELR state. Fields:

REG_NAME string

Name of register being written.

ArchMsg.Warning.ReportedRevisionRegisterRead

DISPLAY Reported revision and patch do not match that of the cpu. Fields:

ORIGINAL_VALUE unsigned int

The value the register would have based on the modelled revision.

REGISTER string

The name of the register that has been read.

REPORTED_VALUE unsigned int

The value the register has been overridden to.

ArchMsg.Warning.Unpredictably Indexed PM Event Register

TAGS **UNPREDICTABLE** DISPLAY Event %{IsDirect:(Selector value|Register)} %{SEL} is >= %{N}.
Fields:

IsDirect unsigned int

Direct Access rather than indirect.

N unsigned int

Number of Accessible Registers.

SEL unsigned int

Selector.

ArchMsg.Warning.cache_contents_unknown

A cache has been activated without being invalidated and may contain **UNKNOWN** content.
DISPLAY %{SIDE:(D|I)}-Cache was enabled in %{REGIME:(SECURE|NON_SECURE|HYP)} regime but was not invalidated since power-on: cache lines could contain **UNKNOWN** content
%{SIDE:} (Note: D-side caches must invalidate SECURE lines even if the cache is only used by NON_SECURE code in order to ensure that unexpected dirty lines tagged SECURE are not naturally evicted). Fields:

REGIME unsigned int

Current translation regime.

SIDE unsigned int

0 d-side cache or unified cache, 1 i-side.

ArchMsg.Warning.dcimvac_matches_watchpoint

TAGS **UNPREDICTABLE** DISPLAY Watchpoints matching an AArch32 DCIMVA are implementation defined.

ArchMsg.Warning.decode_fieldmismatch

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with different fields %{FIELD1} and %{FIELD2}. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_frachitsnegative

TAGS **UNPREDICTABLE** DISPLAY VCVT Instruction is unpredictable with a negative number of fraction bits.

ArchMsg.Warning.decode_initblock

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable in an IT block.

ArchMsg.Warning.decode_initblocknotlast

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable in an IT block when not the last.

ArchMsg.Warning.decode_invalid_condition

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable when the condition is not AL.

ArchMsg.Warning.decode_invalidfieldcombination

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with the values in %{FIELD1} and %{FIELD2}. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_invalidvalue

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with the value in %{FIELD}. Fields:

FIELD enum

Field name.

ArchMsg.Warning.decode_mem_hint_unallocated

TAGS **UNPREDICTABLE** DISPLAY There is no memory hint allocated to this bit pattern.

ArchMsg.Warning.decode_registermatch

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with %{FIELD1} and %{FIELD2} as the same register. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_registermismatch

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with %{FIELD1} and %{FIELD2} as different registers. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_registeroutofrange

TAGS **UNPREDICTABLE** DISPLAY Floating-point instruction is unpredictable indexing beyond the end of the register bank.

ArchMsg.Warning.decode_sbzsbo

TAGS **UNPREDICTABLE** DISPLAY Reserved bits in the instruction are not canonical.

ArchMsg.Warning.decode_transactiontoobig

TAGS **UNPREDICTABLE** DISPLAY Attempt to perform a memory transaction of more than 128 bytes.

ArchMsg.Warning.decode_unpred_other

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable.

ArchMsg.Warning.decode_unpreduseofpc

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable when %{FIELD} is PC. Fields:

FIELD enum

Field name.

ArchMsg.Warning.decode_unpreduseofr13

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable when %{FIELD} is R13. Fields:

FIELD enum

Field name.

ArchMsg.Warning.decode_unpreduseofr13inreglist

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with R13 in register-list.

ArchMsg.Warning.decode_useofd16orhigher

TAGS **UNPREDICTABLE** DISPLAY Attempt to use register D16 or higher where it is not allowed.

ArchMsg.Warning.decode_writebackandbaseinlist

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with write back when the base register is in the transfer list.

ArchMsg.Warning.decode_zeroregistersinlist

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable when no registers are to be transferred.

ArchMsg.Warning.recursive_branch

An instruction is performing a branch that targets the same instruction. If this was intended then a WFI might be more effective DISPLAY recursive branch detected at PC=%{PC}. PRIMARY KEY PC. Fields:

PC unsigned int

Address of instruction causing the branch.

ArchMsg.Warning.recursive_exception

An instruction has generated an exception that targets the same instruction. DISPLAY recursive %{TYPE} exception detected at PC=%{PC} (CPSR=%{CPSR}). PRIMARY KEY PC. Fields:

CPSR unsigned int

Processor state.

PC unsigned int

Address of instruction causing the exception.

TYPE enum

Type of exception.

ArchMsg.Warning.reentrant_vector_catch

TAGS **UNPREDICTABLE** DISPLAY Vector catch debug event on Prefetch or Data abort vector %**{ADDR}**. Fields:

ADDR unsigned int

vector address.

ArchMsg.Warning.reserved_it_state

TAGS **UNPREDICTABLE** DISPLAY Execution with reserved IT state %**{ITSTATE}**. Fields:

ITSTATE unsigned int

reserved IT state.

ArchMsg.Warning.tlb_contents_unknown

Translation has been enabled without invalidating the TLB. DISPLAY Translation was enabled in the %**{PAS:(Secure|Non-secure|Root|Realm)}** EL%**{EL:d}** translation regime (vmid=%**{VMID:x}**) without invalidating %**{INVALIDITY:(|d-side |i-side |)}**entries since power-on: %**{INVALIDITY:(|D||)}**TLB could contain **UNKNOWN** content. Fields:

EL unsigned int

Exception level of translation regime.

INVALIDITY unsigned int

Which TLB may be invalid (if TLBs are separate).

NON_SECURE bool

Whether the translation regime is non-secure.

PAS unsigned int

Physical address space of translation regime.

VMID unsigned int

Current VMID, if applicable.

ArchMsg.Warning.unknown_DLR_DSPSR

Exiting debug state with unknown DLR or DSPSR.

ArchMsg.Warning.unknown_ELR_SPSR

Returning from debug exception with unknown ELR or SPSR.

ArchMsg.Warning.unpred_FAR_on_LRCPC3_non_atomic_mem_access

TAGS **UNPREDICTABLE** Unpredictable FAR/FSTATUS on LRCPC3 pair instructions resulting in two separate memory accessesDISPLAY Unpredictable FAR on LRCPC3 non atomic

memory access at %{ADDR1} with FSTATUS %{FSTATUS1} or %{ADDR2} with FSTATUS %{FSTATUS2}. Fields:

ADDR1 unsigned int

Possible FAR address 1.

ADDR2 unsigned int

Possible FAR address 2.

FSTATUS1 unsigned int

Possible fault status value 1.

FSTATUS2 unsigned int

Possible fault status value 2.

ArchMsg.Warning.unpred_async_tag_check_fail_on_sve_speculative_load_do_update_ffr

Unpredictable set to FALSE of the FFR element associated to SVE speculative load with an async tag check fail TAGS **UNPREDICTABLE**.

ArchMsg.Warning.unpredictable_a32_breakpoint

TAGS **UNPREDICTABLE** DISPLAY Unpredictable breakpoint on a A32 instruction at %{ADDR1} Offset %{OFFSET} BAS %{BAS} for a %{IS_ADDRESS_MISMATCH_BREAKPOINT:(Mismatch|Match)} Breakpoint number %{N:d} Breakpoint address %{ADDR2}. Fields:

ADDR1 unsigned int

Address.

ADDR2 unsigned int

Breakpoint address.

BAS unsigned int

BAS field.

IS_ADDRESS_MISMATCH_BREAKPOINT bool

Is a Mismatch breakpoint, not a Match breakpoint.

N unsigned int

Breakpoint number.

OFFSET unsigned int

Always has a value of zero.

ArchMsg.Warning.unpredictable_banked_register_access

TAGS **UNPREDICTABLE** DISPLAY Access to banked register (R=%{R:d} SYSm=%{SYSM:2x}) is unpredictable. Fields:

R unsigned int

R bit.

SYSM unsigned int

SYSm.

ArchMsg.Warning.unpredictable_ccfail_undef

TAGS **UNPREDICTABLE** DISPLAY An instruction %{OPCODE} at VA=%{VADDR} failed its condition codes check but would otherwise have caused a non-data-dependent trap or undefined exception. Behaviour is implementation-defined choice of exception or nop. Fields:

OPCODE unsigned int

opcode of the instruction.

VADDR unsigned int

Virtual address of the instruction being executed.

ArchMsg.Warning.unpredictable_ici_bits

TAGS **UNPREDICTABLE** DISPLAY ICI bits do not apply or are architecturally unpredictable for instruction.

ArchMsg.Warning.unpredictable_t32_breakpoint

TAGS **UNPREDICTABLE** DISPLAY Unpredictable breakpoint on a T32 instruction at %{ADDR1} Offset %{OFFSET} BAS %{BAS} for a %{IS_ADDRESS_MISMATCH_BREAKPOINT:(Mismatch|Match)} Breakpoint number %{N:d} Breakpoint address %{ADDR2}. Fields:

ADDR1 unsigned int

Address.

ADDR2 unsigned int

Breakpoint address.

BAS unsigned int

BAS field.

IS_ADDRESS_MISMATCH_BREAKPOINT bool

Is a Mismatch breakpoint, not a Match breakpoint.

N unsigned int

Breakpoint number.

OFFSET unsigned int

Indicates which halfword of a 32-bit Thumb instruction the breakpoint is set on. 0 for the first halfword or if the instruction is 16 bits wide, or 1 for the second halfword.

ArchMsg.Warning.unpredictable_t32_vector_catch

TAGS **UNPREDICTABLE** DISPLAY Unpredictable Vector catch exception on a T32 instruction at %{ADDR} Offset %{OFFSET}. Fields:

ADDR unsigned int

Address.

OFFSET unsigned int

Indicates which halfword of a 32-bit Thumb instruction the vector catch exception is set on. 0 for the first halfword or if the instruction is 16 bits wide, or 1 for the second halfword.

ArchMsg.Warning.unpredictable_watchpoint_far

TAGS **UNPREDICTABLE** DISPLAY The value for FAR/EDWAR for hitting this watchpoint may be anywhere in the range %{LOWER_BOUND} to %{UPPER_BOUND}. Fields:

LOWER_BOUND unsigned int

the lowest address accessed by the instruction that triggered the watchpoint.

UPPER_BOUND unsigned int

the highest watchpointed address accessed by that instruction.

ArchMsg.Warning.unpredictable_watchpoint_far_isscmp

DISPLAY The value for FAR/EDWAR for hitting this watchpoint may be anywhere in the range %{LOWER_BOUND} to %{UPPER_BOUND}. Fields:

LOWER_BOUND unsigned int

the lowest address accessed by the instruction that triggered the watchpoint.

UPPER_BOUND unsigned int

the highest watchpointed address accessed by that instruction.

ArchMsg.Warning.warning_AdvSIMDExpandImmUnexpectedZero

TAGS **UNPREDICTABLE** DISPLAY AdvSIMDExpandImm may treat this immediate value as **UNPREDICTABLE**.

ArchMsg.Warning.warning_ConditionalSMC

TAGS **UNPREDICTABLE** DISPLAY SMC instruction has **UNPREDICTABLE** effects when conditional and when combined with traps.

ArchMsg.Warning.warning_TLBID_out_of_range

TLBID out of range in TLBI according to the configuration of NIS, NVIS, NOS and NVOS. Fields:

NIS unsigned int

NIS.

NOS unsigned int

NOS.

NVIS unsigned int

NVIS.

NVOS unsigned int

NVOS.

TLBID unsigned int

The TLBID value used in the TLBI instruction.

ArchMsg.Warning.warning_TTL_or_TG_mismatches_current_translation_granule

TTL or TG mismatches current translation granule in TLBI VA/RVA/IPA/RIPA. Fields:

CURRENT_TG unsigned int

Current TG as indicated by TCR_ELx or VTCR_EL2.

TLBI_TG unsigned int

TG (or TTL[3:2]) used in TLBI instruction.

ArchMsg.Warning.warning_access_crosses_page_boundary

TAGS **UNPREDICTABLE** DISPLAY Access crosses page boundary.

ArchMsg.Warning.warning_access_crosses_page_boundary_has_fault

TAGS **UNPREDICTABLE** DISPLAY Access that crosses page boundary has fault(s).

ArchMsg.Warning.warning_access_crosses_page_boundary_has_fault_on_both_pages

TAGS **UNPREDICTABLE** DISPLAY Access that crosses page boundary has a fault on each page.

Fields:

ADDR1 unsigned int

FAR address on first page.

ADDR2 unsigned int

FAR address on second page.

FSTATUS1 unsigned int

Fault status on first page.

FSTATUS2 unsigned int

Fault status on second page.

ArchMsg.Warning.warning_access_crosses_page_boundary_spanning_different_memory

TAGS **UNPREDICTABLE** DISPLAY Access at address %{ADDR} crosses page from MEM_TYPE %{MEMTYPE_PAGE1} to %{MEMTYPE_PAGE2}. Fields:

ADDR unsigned int

address of access crossing page.

MEMTYPE_PAGE1 signed int

First Page's memory type (0-Normal, 1-Non-Normal).

MEMTYPE_PAGE2 signed int

Second Page's memory type (0-Normal, 1-Non-Normal).

ArchMsg.Warning.warning_access_wraps_around_memory

TAGS **UNPREDICTABLE** DISPLAY Access wraps around memory in %{WIDTH:d}bit mode.

Fields:

WIDTH unsigned int

Address width.

ArchMsg.Warning.warning_bcr_linking_status

TAGS **UNPREDICTABLE** DISPLAY DBGBCR.{BT2,BT} combination gives unpredictable behaviour.

ArchMsg.Warning.warning_bcr_mask_reserved

TAGS **UNPREDICTABLE** DISPLAY DBGBCR_ELn.MASK is set to a reserved value.

ArchMsg.Warning.warning_bvr_masked_bit_not_zero

TAGS **UNPREDICTABLE** DISPLAY DBGBVRn_EL1.VA contains a bit masked by DBGBCRn_EL1.MASK that is not zero.

ArchMsg.Warning.warning_ccsidr_unimplemented_level

TAGS **UNPREDICTABLE** DISPLAY CCSIDR read while CSSELR %{CSSELR:x} points to an unimplemented cache level. Fields:

CSSELR unsigned int

current effective value of CSSELR.

ArchMsg.Warning.warning_change_to_ns_when_tge_set

TAGS **UNPREDICTABLE** DISPLAY Attempting to change NS to 1 when HCR.TGE = 1.

ArchMsg.Warning.warning_contiguous_bit_check_abort

TAGS **UNPREDICTABLE** DISPLAY An abort occurred whilst attempting to check TLB entry is contiguous at %{ENTRY_ADDR}. Fields:

ENTRY_ADDR unsigned int

address of conflicting TLB entry.

ArchMsg.Warning.warning_contiguous_bit_error

TAGS **UNPREDICTABLE** DISPLAY TLB entry at %{ENTRY_ADDR} was not contiguous with entry at %{CONTIG_ADDR} Contents are %{ENTRY_DATA} but expected %{CONTIG_DATA}. Fields:

CONTIG_ADDR unsigned int

address of the TLB first read, with which this entry is expected to be contiguous.

CONTIG_DATA unsigned int

expected contents based on the entry at CONTIG_ADDR.

ENTRY_ADDR unsigned int

address of conflicting TLB entry.

ENTRY_DATA unsigned int

contents of conflicting TLB entry.

ArchMsg.Warning.warning_contiguous_bit_error_D128

TAGS **UNPREDICTABLE** DISPLAY TLB entry at %{ENTRY_ADDR} was not contiguous with entry at %{CONTIG_ADDR} Contents are %{ENTRY_DATA_HI} %{ENTRY_DATA_LO} but expected %{CONTIG_DATA}. Fields:

CONTIG_ADDR unsigned int

address of the TLB first read, with which this entry is expected to be contiguous.

CONTIG_DATA unsigned int

expected contents based on the entry at CONTIG_ADDR.

ENTRY_ADDR unsigned int

address of conflicting TLB entry.

ENTRY_DATA_HI unsigned int

Higher bit contents of conflicting TLB entry.

ENTRY_DATA_LO unsigned int

Lower bit contents of conflicting TLB entry.

ArchMsg.Warning.warning_cp10_cp11_mismatch

TAGS **UNPREDICTABLE** DISPLAY Unpredictable configuration of CP10 and CP11 controls at EL
%{EL:d}. Fields:

EL unsigned int

EL owning the control.

ArchMsg.Warning.warning_cp10_cp11_reserved_value

TAGS **UNPREDICTABLE** DISPLAY Unpredictable configuration of CPACR CP10 (%{CP10}) or
CP11 (%{CP11}). Fields:

CP10 unsigned int

CP10 access.

CP11 unsigned int

CP11 access.

ArchMsg.Warning.warning_csselr_level_out_of_range

TAGS **UNPREDICTABLE** DISPLAY CSSELR.Level out of range, written value %{LEVEL} with only
%{IMPLEMENTED:d} levels implemented. Fields:

IMPLEMENTED unsigned int

number of cache levels implemented.

LEVEL unsigned int

written value of CSSELR.Level.

TYPE enum

The type of cache selected by CSSELR.InD and TnD.

ArchMsg.Warning.warning_debug_flow_control_bits_not_obeyed

TAGS **UNPREDICTABLE** DISPLAY Flow control bits for %{REGISTER} were not obeyed, causing
%{ERROR}. Fields:

ERROR enum

type of error resulting.

REGISTER enum

register under consideration.

ArchMsg.Warning.warning_debug_register_access_during_reset

TAGS **UNPREDICTABLE** DISPLAY Unpredictable %{IS_WRITE:(read|write)} access to debug
register offset %{OFFSET} during reset. Fields:

IS_WRITE unsigned int

Write Not Read.

OFFSET unsigned int

Register Offset.

ArchMsg.Warning.warning_decode_cps_inconsistent_fields

TAGS **UNPREDICTABLE** DISPLAY The mode setting fields in the CPS instruction are inconsistent.

ArchMsg.Warning.warning_decode_invalid_state

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable in current exception-level/security state.

ArchMsg.Warning.warning_default_cacheable_mmu_on

TAGS **UNPREDICTABLE** DISPLAY Default cacheable is enabled (HCR.DC = 1) while MMU is enabled.

ArchMsg.Warning.warning_default_cacheable_vmmu_off

TAGS **UNPREDICTABLE** DISPLAY Default cacheable is enabled (HCR.DC = 1) while HCR.VM = 0.

ArchMsg.Warning.warning_deprecated_wvr_bit_2_set

TAGS **UNPREDICTABLE** DISPLAY DBGWVR%{N:d}_EL1=%{WVR:h} has bit 2 set, deprecated in ARMv8. Fields:

N unsigned int

index.

WVR unsigned int

DBGWVR.

ArchMsg.Warning.warning_eret_while_software_step_active_pending

Missing ISB between setting MDSCR_EL1.SS and ERET TAGS **UNPREDICTABLE**.

ArchMsg.Warning.warning_exclusive_to_non_normal

TAGS **UNPREDICTABLE** DISPLAY Exclusive Access to Non-Normal Memory %{ADDR}. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_exclusive_to_non_writeback

TAGS **UNPREDICTABLE** DISPLAY Exclusive Access to Non-Writeback-Cacheable Memory %{ADDR}. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_execute_from_device_memory

TAGS **UNPREDICTABLE** DISPLAY An attempt was made to execute from Device memory.

ArchMsg.Warning.warning_illegal_cpsr_mode

TAGS **UNPREDICTABLE** DISPLAY Writing an illegal or unimplement mode (%{NEW_MODE:h}) to CPSR was unpredictable. Fields:

NEW_MODE unsigned int

New value of CPSR.M.

ArchMsg.Warning.warning_illegal_srs_mode

TAGS **UNPREDICTABLE** DISPLAY Illegal or **UNPREDICTABLE** mode (%{MODE:h}) used for SRS instruction. Fields:

MODE unsigned int

mode for Banked SP.

ArchMsg.Warning.warning_implementation_defined_read_debug_register_in_SCS

Read Access to Debug registers from the processor is **IMPLEMENTATION DEFINED** TAGS **IMP_DEF**. Fields:

PPB_OFFSET unsigned int

debug register offset.

ArchMsg.Warning.warning_implementation_defined_sequential_security_transitions_supported

The behavior of sequential instruction fetches that cross from non-secure to secure memory and contain an SG instruction is **CONSTRAINED UNPREDICTABLE** TAGS **UNPREDICTABLE**.

ArchMsg.Warning.warning_implementation_defined_stack_limit_check_supported

It is **IMPLEMENTATION DEFINED** whether stack pointer limit checking is performed for this instruction TAGS **IMP_DEF**. Fields:

ADDRESS unsigned int

Address of instruction.

OPCODE unsigned int

instruction opcode.

ArchMsg.Warning.warning_implementation_defined_taggednonsharedmemorytype

TAGS **IMP_DEF** DISPLAY Memory attribute has Tagged, non shared memory type; it is **IMPLEMENTATION DEFINED** whether the memory location is treated as Tagged or non-Tagged.

ArchMsg.Warning.warning_implementation_defined_write_debug_register_in_SCS

Write Access to Debug registers from the processor is **IMPLEMENTATION DEFINED** TAGS **IMP_DEF**. Fields:

DATA unsigned int

data attempted to be written.

PPB_OFFSET unsigned int

debug register offset.

ArchMsg.Warning.warning_invalid_tcr_granule

TAGS **UNPREDICTABLE** DISPLAY TCR.TG%{TG_ID:(0|1)} bits have been set to a granule size %{REQUEST:d}K not implemented - using %{SUBSTITUTE:d}K. Fields:

REQUEST signed int

page size requested (or 0 for reserved).

SUBSTITUTE signed int

best guess available page size.

TG_ID bool

bits TG0 or TG1.

ArchMsg.Warning.warning_load_multiple_user_registers_from_user_mode

TAGS UNPREDICTABLE DISPLAY An LDM(user registers) instruction executed from user mode.

ArchMsg.Warning.warning_mte_in_mem_attr_not_supported

TAGS UNPREDICTABLE DISPLAY Memory attribute %{ATTR} has Tagged memory type but Armv8.5-MTE is not implemented. Fields:

ATTR unsigned int

Memory attribute.

ArchMsg.Warning.warning_reserved_breakpoint_state_match

TAGS UNPREDICTABLE DISPLAY %{IS_BREAKPOINT:(Watchpoint|Breakpoint)} programmed with a reserved combination of HMC, SSC and %{IS_BREAKPOINT:(PAC|PMC)}. Fields:

IS_BREAKPOINT unsigned int

Is a breakpoint, not a watchpoint.

ArchMsg.Warning.warning_shareability

TAGS UNPREDICTABLE DISPLAY Unpredictable: combination of the 1st and 2nd stages of translation is Normal Inner Non-Cacheable, Outer Non-Cacheable.

ArchMsg.Warning.warning_software_step_set_while_enabled

TAGS UNPREDICTABLE DISPLAY MDSCR_EL1.SS set to 1 while software step debug exceptions are enabled.

ArchMsg.Warning.warning_thumb_instruction_wraps_around_memory

TAGS UNPREDICTABLE DISPLAY Thumb instruction wraps around memory.

ArchMsg.Warning.warning_ttbr_sbz_bits_are_not_zero

TAGS UNPREDICTABLE DISPLAY %{TTBR}[%{MSB:d};0] should be zero, but are not. Fields:

MSB unsigned int

x-1.

TTBR enum

which TTBR register.

ArchMsg.Warning.warning_unaligned_address_dbgdtrrx_write

TAGS UNPREDICTABLE DISPLAY Unaligned address %{ADDR} in X0/R0 when DBGDTRRX written in memory access mode. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unaligned_address_dbgdtrtx_read

TAGS **UNPREDICTABLE** DISPLAY Unaligned address %{ADDR} in XO/RO when DBGDTRTX read in memory access mode. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unaligned_to_device

TAGS **UNPREDICTABLE** DISPLAY Unaligned Access to Device Memory %{ADDR}. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unaligned_to_strongly_ordered

TAGS **UNPREDICTABLE** DISPLAY Unaligned Access to Strongly Ordered Memory %{ADDR}. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unknown_sau_rnr

TAGS **UNPREDICTABLE** DISPLAY SAU_RNR was set to an unsupported value.

ArchMsg.Warning.warning_unpredictable_AIRCR_PRIS_and_BFHFNMINs

TAGS **UNPREDICTABLE** DISPLAY The effect of setting both AIRCR.BFHFNMINs and AIRCR.PRIS to 1 is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_AIRCR_VECTCLRACTIVE_when_not_in_debug

TAGS **UNPREDICTABLE** DISPLAY The effect of writing a 1 to AIRCR.VECTCLRACTIVE if the processor is not halted in Debug state is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_AIRCR_VECTRESET_and_SYSRESETREQ

TAGS **UNPREDICTABLE** DISPLAY When the processor is halted in Debug state, if a write to the register writes a 1 to both VECTRESET and SYSRESETREQ, the behavior is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_AIRCR_VECTRESET_when_not_in_debug

TAGS **UNPREDICTABLE** DISPLAY The effect of writing a 1 to AIRCR.VECTRESET if the processor is not halted in Debug state is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_AIRCR_incorrect_VKEY

TAGS **UNPREDICTABLE** DISPLAY The value 0x05FA must be written to AIRCR.VKEY, otherwise the register write is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_EXC_RETURN_Reserved_Bit

EXC_RETURN[23:7] are reserved with the special condition that all bits should be written as one. Values other than all 1s are **UNPREDICTABLE**. TAGS **UNPREDICTABLE**. Fields:

EXC_RETURN unsigned int

Special PC value.

SBO_MASK unsigned int

Which bits Should Be One to not UNPRED.

ArchMsg.Warning.warning_unpredictable_PMEVTYPErn_ELO_VS_reserved_value

TAGS UNPREDICTABLE DISPLAY Unpredictable reserved value of PMEVTYPER<n>_ELO.VS.

ArchMsg.Warning.warning_unpredictable_change_to_DEMCR_MON_STEP_at_insufficient_priority

TAGS UNPREDICTABLE DISPLAY The effect of changing DEMCR.MON_STEP at an execution priority that is lower than the priority of the DebugMonitor exception is **CONSTRAINED UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_change_to_priority_of_active_exception

TAGS UNPREDICTABLE DISPLAY Changing the priority of an active exception is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_clear_lspact

TAGS UNPREDICTABLE DISPLAY **UNPREDICTABLE** state on context switch. Fields:

DRVEXC_TAKEN unsigned int

Derived exception taken, not pended.

REACHED_LAST_STORE unsigned int

Only the last faulted.

ArchMsg.Warning.warning_unpredictable_dbg_exit_unaligned_dlr

TAGS UNPREDICTABLE DISPLAY Debug exit to AArch32 with DLR[0] set to 1. PC[0] can be set to 0 or to DLR[0].

ArchMsg.Warning.warning_unpredictable_exception_catch

TAGS UNPREDICTABLE DISPLAY Generation of an exception catch event is unpredictable.

ArchMsg.Warning.warning_unpredictable_exception_return_inconsistent_state

TAGS UNPREDICTABLE DISPLAY state on exception return is unpredictable.

ArchMsg.Warning.warning_unpredictable_exception_return_instruction

TAGS UNPREDICTABLE DISPLAY exception return instruction is unpredictable.

ArchMsg.Warning.warning_unpredictable_in_debug_state

TAGS UNPREDICTABLE DISPLAY Instruction is **UNPREDICTABLE** when executed in debug state.

ArchMsg.Warning.warning_unpredictable_mair_encoding

TAGS UNPREDICTABLE DISPLAY Memory attribute %{ATTR} has invalid MAIR encoding; this may lead to unpredictable results. Fields:

ATTR unsigned int

Memory attribute.

ArchMsg.Warning.warning_unpredictable_pmu_counter_access

TAGS UNPREDICTABLE DISPLAY Access to PMU counters from non-secure ELO or EL1 is **UNPREDICTABLE** with MDCR_EL2.HPMN set to 0.

ArchMsg.Warning.warning_unpredictable_prioritization_breakpoint_match_vector_catch

TAGS UNPREDICTABLE DISPLAY Two events with the same priority occurred at the same instruction: Address Matching Vector Catch debug event, and Breakpoint debug event. It is constrained unpredictable which is taken.

ArchMsg.Warning.warning_unpredictable_stack_selection

TAGS **UNPREDICTABLE** DISPLAY **UNPREDICTABLE** state on context switch.

ArchMsg.Warning.warning_unpredictable_tsize_out_of_range

TAGS **UNPREDICTABLE** DISPLAY **%{TCR}.T%{TTBR:d}SZ (%{TSIZE:d} bits)** is out of range. Must be between 25 and **%{TSIZE_MAX:d}**. Fields:

TCR enum

Which TCR.

TSIZE signed int

Value of TSize.

TSIZE_MAX signed int

The maximum allowed value.

TTBR unsigned int

Which TTBR region.

ArchMsg.Warning.warning_unpredictable_unaligned_pc_as_base_register

TAGS **UNPREDICTABLE** DISPLAY The PC value must be word-aligned, otherwise the behavior of the instruction is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_unaligned_pop_stack

TAGS **UNPREDICTABLE** DISPLAY **CONSTRAINED UNPREDICTABLE** unaligned stack frame **%{FRAME_ADDR:x}** in exception-return. Fields:

FRAME_ADDR unsigned int

unaligned stack frame address.

ArchMsg.Warning.warning_unpredictable_unaligned_to_non_writeback

TAGS **UNPREDICTABLE** DISPLAY Unaligned Access to Non-Writeback-Cacheable Memory.

ArchMsg.Warning.warning_unpredictable_vmsa_memattrib

TAGS **UNPREDICTABLE** DISPLAY Unpredictable vmsa memory attribute with texcb value programmed to **%{TEXCB}** when tex remap is **%{TEX_REMAP}**. Fields:

TEXCB unsigned int

texcb value.

TEX_REMAP bool

use tex remap.

ArchMsg.Warning.warning_unpredictable_vtcr_t0sz_sl0

TAGS **UNPREDICTABLE** DISPLAY The combination of **VTCR.TOSZ=%{TOSZ:d}** and **VTCR.SL0=%{SL0:d}** is unpredictable. Fields:

SL0 signed int

VTCR.SL0, starting level for stage 2 translation table walks.

TOSZ signed int

VTCR.TOSZ, required input address range.

ArchMsg.Warning.warning_unpredictable_write_DHCSR

TAGS UNPREDICTABLE DISPLAY UNPREDICTABLE set of bit-changes in DHCSR. Fields:

NEW unsigned int

data attempted to be written.

OLD unsigned int

previous value.

ArchMsg.Warning.warning_unpredictable_write_SHCSR

TAGS UNPREDICTABLE DISPLAY UNPREDICTABLE setting a pending bit to 1 for an exception with priority >= execution priority.

ArchMsg.Warning.warning_unsupported_access_to_memory_mapped_register

TAGS UNPREDICTABLE DISPLAY Access to memory mapped register is unsupported.

ArchMsg.Warning.warning_user_jmcr_access

TAGS UNPREDICTABLE DISPLAY User %{WRITE:(write|read)} access to JMCR is unpredictable. Fields:

WRITE unsigned int

access is write.

ArchMsg.Warning.warning_wcr_mask_and_bas

TAGS UNPREDICTABLE DISPLAY DBGWCR_EL1.MASK is non-zero and BAS is not 0xff.

ArchMsg.Warning.warning_wcr_mask_reserved

TAGS UNPREDICTABLE DISPLAY DBGWCR_EL1.MASK is set to a reserved value.

ArchMsg.Warning.warning_wcr_non_configuous_bas

TAGS UNPREDICTABLE DISPLAY DBGWCR_EL1.BAS has non-contiguous set of ones.

ArchMsg.Warning.warning_wvr_masked_bit_not_zero

TAGS UNPREDICTABLE DISPLAY DBGWVRn_EL1.VA contains a bit masked by DBGWCRn_EL1.MASK that is not zero.

ArchMsg.Why.why_branch_target_exception

A branch target exception has occurred. DISPLAY branch target exception: target pc=%{PC}, target opcode=%{OPCODE}. PRIMARY KEY PC. Fields:

BTYPE unsigned int

pstate.BTYPE at time branch taken.

OPCODE unsigned int

opcode at branch target.

PC unsigned int

Virtual Address of branch target.

ArchMsg.Why.why_illegal_state

An illegal mode change has occurred. DISPLAY illegal mode change: %{MESSAGE}. Fields:

MESSAGE string

The message.

ArchMsg.warning_invalid_interval_counter_reload_value

Invalid interval counter reload value in PMSIRR_EL1.INTERVAL: software should set this to a value greater than or equal to the minimum indicated by PMSIDR_EL1.Interval.

ArchMsg.warning_invalid_spe_buffer_write_pointer

Invalid value programmed in PMBPTR_EL1.PTR or PMBLIMITR_EL1.LIMIT: Current write pointer must be at least one sample record below the write limit pointer and PMBPTR_EL1.PTR[63:56] must be equal to PMBLIMITR_EL1.LIMIT[63:56].

BRANCH_MISPREDICT

Simulating branch mispredict. Fields:

PC unsigned int

Origin address (or 0 if unavailable).

BRA_DIR

Direct branches, to immediate address. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISET enum

The instruction set of the branch instruction.

IS_COND bool

Indicates if this is a conditional branch.

IS_HINTED bool

Whether the conditional instruction is hinted.

IS_ISB bool

Whether the instruction is ISB.

IS_LINK bool

Indicates if this is a branch with link.

PC unsigned int

The address of the branch instruction.

TARGET_ISET enum

The instruction set after the branch.

TARGET_PC unsigned int

The address the instruction branches to.

BRA_INDIR

Indirect branches, perhaps to a register. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISET enum

The instruction set of the branch instruction.

IS_COND bool

Indicates if this is a conditional branch.

IS_HINTED bool

Whether the conditional instruction is hinted.

IS_ISB bool

Whether the instruction is ISB.

IS_LINK bool

Indicates if this is a branch with link.

PC unsigned int

The address of the branch instruction.

TARGET_ISET enum

The instruction set after the branch.

TARGET_PC unsigned int

The address the instruction branches to.

BRA_INDIR_CCFAIL

Indirect branches fail, perhaps to a register.

BRA_RET_CCFAIL

Branch return fail, perhaps to a register.

CACHE_MAINTENANCE_OP

Cache Maintenance Operation. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

DATA unsigned int

Specified MVA or Set/way.

FUNCTION enum

Clean / Invalidate.

SCOPE enum

Affected region.

SIDE enum

Inst / Data.

CCFAIL

Conditional instruction condition check fail. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CCFAIL_UNC

Unconditional instruction condition check fail. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CCPASS

Conditional instruction condition check pass. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CCPASS_UNC

Unconditional instruction condition check pass. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CODE_CACHE_MAINT

Code cache maintenance. Fields:

END_INCL unsigned int

Inclusive end address for VA or PA requests.

NS bool

Non-secure world for PA.

START unsigned int

Start address for VA or PA requests.

TYPE enum

Request type.

COMPILE_BLOCK_END

Last instruction of basic block translated. Fields:

VADDR unsigned int

Address of next instruction after this basic block.

COMPILE_BLOCK_START

First instruction of basic block translated. Fields:

VADDR unsigned int

Address of first instruction in basic block.

COMPILE_INST

ARM instruction compiled. Fields:

DISASS string

Disassembly of the instruction.

ISSET enum

The instruction set of this instruction.

ITSTATE unsigned int

The ITSTATE current for the instruction.

OPCODE unsigned int

The opcode of the instruction.

PC unsigned int

The address of the instruction.

SIZE unsigned int

The size of the instruction in bytes.

CONTEXTIDR

Context ID Register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

NS bool

Secure or nonsecure banked register is accessed.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The new value written.

CONTEXT_SYNC

Called for every context synchronization event. Fields:

ARCH_CSE bool

Architectural Context Synchronization Event.

PC unsigned int

The address of the synchronization instruction.

SYSREG_SYNC bool

System register synchronization event.

CORE_ENDIAN

Core BE8 Big-Endian state changed. Fields:

BE8 bool

Core BE8 Big-Endian state.

CORE_INFO

Static processor attributes. Only triggered by a call to DumpState(). Fields:

ARCH_PROFILE enum

The architecture profile of the core.

CLUSTER_ID unsigned int

The cluster ID of this processor.

CORE_NUM unsigned int

The number of this core in an MP processor.

DETAILED_CORE_FEATURES enum

Detailed core features as a bitfield. Enum values are bitmasks to extract the info from the field.

FPU_VERSION enum

The VFP version implemented by the core.

MEM_ARCH enum

The memory architecture of the core.

MIDR_EL1 unsigned int

MIDR_EL1 value.

NUM_CORES unsigned int

The number of cores in this MP processor.

QUANTUM_SIZE unsigned int

The default quantum size of the core (in ticks).

SECURITY_FEATURES bool

Does the core have security features?.

CORE_LOADS

Processor load accesses. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ACQREL enum

Is this an acquire/release.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DATA unsigned int

The data read or written.

ELEMENT_SIZE unsigned int

Width of each element.

ENDIANNESS enum

Byte ordering of this access.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

LOCK enum

Normal, exclusive or locked access.

MECID unsigned int

Memory Encryption Context ID, if applicable.

MECID2 unsigned int

Memory Encryption Context ID of the second page, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PAGE_SIZE unsigned int

Page size as $\log_2(\text{size in bytes})$.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK, 2=Exclusive Failed, 3=Unsupported.

SIZE unsigned int

Width of the access in bytes. Only required if DATA is not traced.

TRANS bool

Is this a translated access.

VADDR unsigned int

The virtual address of the access.

CORE_REGS

Changes of the core registers R0 to R14. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number, 0 to 14.

MODE enum

Bank of the register accessed.

OLD_VALUE unsigned int

The old value overwritten.

PHYS_ID enum

The physical register accessed.

VALUE unsigned int

The new value written to the register.

CORE_REGS64

Changes of the core registers X0..X30, SP_ELn. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ID enum

The register number, 0-30 for X0-X30, >=32 for SP_ELn.

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

CORE_REGS64_READ

General purpose AArch64 register has been read X0..X30, SP_ELn. Fields:

ID enum

The register number, 0-30 for X0-X30, >=32 for SP_ELn.

VALUE unsigned int

The value read from the register.

CORE_STALLED_TRANSITION

Detect when a core is prevented from progressing PChannel states. Fields:

CURRENT_STATE enum

Current power mode of core.

REASON string

The reason a transition was stalled, and what must be done to unblock it.

TARGET_STATE enum

Desired power mode of core after transition.

CORE_STORES

Processor store accesses. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ACQREL enum

Is this an acquire/release.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DATA unsigned int

The data read or written.

ELEMENT_SIZE unsigned int

Width of each element.

ENDIANNESS enum

Byte ordering of this access.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

LOCK enum

Normal, exclusive or locked access.

MECID unsigned int

Memory Encryption Context ID, if applicable.

MECID2 unsigned int

Memory Encryption Context ID of the second page, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PAGE_SIZE unsigned int

Page size as $\log_2(\text{size in bytes})$.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK, 2=Exclusive Failed, 3=Unsupported.

SIZE unsigned int

Width of the access in bytes. Only required if DATA is not traced.

TRANS bool

Is this a translated access.

VADDR unsigned int

The virtual address of the access.

CP14_READ

System Coprocessor register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP14 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc1 unsigned int

opcode 1.

opc2 unsigned int

opcode 2.

CP14_WRITE

System Coprocessor register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP14 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc1 unsigned int

opcode 1.

opc2 unsigned int

opcode 2.

CP15_READ

System Coprocessor register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP15 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc1 unsigned int

opcode 1.

opc2 unsigned int

opcode 2.

CP15_READ64

System Coprocessor register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP15 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc unsigned int

opcode 1.

CP15_WRITE

System Control Coprocessor register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP15 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc1 unsigned int

opcode 1.

opc2 unsigned int

opcode 2.

CP15_WRITE64

System Coprocessor register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP15 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc unsigned int

opcode 1.

CPSR

CPSR change. (In AArch64 PSTATE represented in SPSR format). Fields:

CORE_NUM unsigned int

Core number in a multi processor.

OLD_VALUE unsigned int

The old CPSR value.

UNKNOWN unsigned int

Bits within the register that have unknown value.

VALUE unsigned int

The new CPSR value.

CRYPTO_SPEC

Every crypto instruction speculatively executed.

CorePowerStateChange

Core power-up state has changed. DISPLAY Core powered %{PU:(up|down)}. Fields:

PU bool

Whether the core powered up.

DATA_CACHE_SET_ALLOC_TAGS

Set allocation tags from DC GVA instruction. Fields:

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

NSDESC unsigned int

The security state of the access.

PADDR unsigned int

The physical address of the access.

PAS enum

The physical address space of the page.

RESPONSE enum

0=Aborted, 1=OK.

SIZE unsigned int

The size of memory in bytes.

TAG unsigned int

The value of the tag being set.

TAGGING_TYPE enum

Type of Memory tagging (Physical or Virtual).

TAG_VADDR unsigned int

The virtual address of the Virtual Tag if Virtual tagging is used; otherwise 0.

VADDR unsigned int

Virtual address used for setting allocation tags.

DATA_CACHE_ZERO

Zero and invalidate cache line from DC ZVA instruction. Fields:

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

NSDESC unsigned int

The security state of the access.

PADDR unsigned int

The physical address of the access.

PADDR_TAG unsigned int

If tag checked, the physical address tag for the access.

PAS enum

The physical address space of the page.

RESPONSE enum

0=Aborted, 1=OK.

SIZE unsigned int

The size of zero'd memory in bytes.

TAG_CHECKED bool

Is this access tag checked.

VADDR unsigned int

Virtual address of the zero'd cache line.

DEBUG_EVENT

Hardware debug support event. Fields:

EVENT enum

Description of event.

VALUE unsigned int

data value.

DEBUG_TARGET_ISET

Hardware debug target ISA. Fields:

TARGET_ISET enum

The target instruction set of the debug state.

DMI_ALLOCATE

DMI allocated. Fields:

ACCESS_DIRECTION enum

READ / WRITE.

PADDR unsigned int

The physical address of the DMI.

PTR unsigned int

DMI range pointer.

SIDE enum

DSIDE / ISIDE.

SIZE unsigned int

The size of DMI.

VADDR unsigned int

The virtual address of the DMI..

DMI_HIT

DMI hit, access used DMI (only fire for small DMI regions i.e. less than a full page). Fields:

ACCESS_DIRECTION enum

READ / WRITE.

PADDR unsigned int

The physical address of the DMI.

SIDE enum

DSIDE / ISIDE.

SIZE unsigned int

The size of the DMI.

VADDR unsigned int

The virtual address of the DMI..

DMI_REVOKE

DMI revoked. Fields:

ACCESS_DIRECTION enum

READ / WRITE.

PADDR unsigned int

The physical address of the DMI.

SIDE enum

DSIDE / ISIDE.

SIZE unsigned int

The size of DMI.

VADDR unsigned int

The virtual address of the DMI..

END_COMPILE

Compilation end. Fields:

END_OF_PAGE_COUNT unsigned int

Number of basic blocks exited due to page boundary since START_COMPILE.

FETCHFAIL_COUNT unsigned int

Number of basic blocks exited due to ifetch failure START_COMPILE.

INST_COUNT unsigned int

Number of instructions compiled since START_COMPILE.

NONSEQ_COUNT unsigned int

Number of basic blocks exited due to non-sequential instructions since START_COMPILE.

PAGE_STRADDLE_COUNT unsigned int

Number of basic blocks exited due to unaligned instructions crossing page since START_COMPILE.

ESB

Error Synchronization Barrier hit. Fields:

ASYNC bool

ESB is async.

IMPLICIT bool

ESB is implicit.

EXCEPTION

Exception taken. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ESR unsigned int

The value assigned to ESR when exception is taken.

ESR_EC unsigned int

The Value assigned to ESR.EC when exception is taken.

IS_PHYSICAL bool

Physical or Virtual exception.

IS_TRAP bool

Exception coming from a trap.

LR unsigned int

The value assigned to the link register.

PC unsigned int

The location where the exception occurred.

PREFERRED_RETURN unsigned int

The preferred return address for the exception.

TAKEN_LOCALLY bool

Whether exception is taken locally(same exception level).

TARGET_ISET enum

The instruction set of the exception handler code.

TARGET_PC unsigned int

The address the exception branches to.

VECTOR enum

The exception vector.

VECTOR_OFFSET unsigned int

The offset of an exception vector from the base address.

EXCEPTION_END

Every exception completed.

EXCEPTION_INFO

Trace human-readable reason of the current exception for data/instruction aborts, floating point exceptions and unknown exceptions. Fields:

DESCRIPTION string

Reason for current exception.

ESR_EC unsigned int

The value of ESR.EC when exception is taken.

EXCEPTION_RAISE

Every exception raised.

EXCEPTION_RETURN

Branches on leaving exception. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISET enum

The instruction set of the branch instruction.

IS_COND bool

Indicates if this is a conditional branch.

IS_HINTED bool

Whether the conditional instruction is hinted.

IS_ISB bool

Whether the instruction is ISB.

IS_LINK bool

Indicates if this is a branch with link.

PC unsigned int

The address of the branch instruction.

TARGET_ISET enum

The instruction set after the branch.

TARGET_PC unsigned int

The address the instruction branches to.

EXCEPTION_RETURN_PREBRANCH

Exception return event that must occur before PC and processor state have updated.

EXCEPTION_START

Every exception started.

FIQ_TAKEN

FIQ taken exception.

FP_EXCEPTIONS_SIGNALLED

Trace which FP exceptions were signalled as a part of the FP operations performed by an instruction. Fields:

FP_EXCEPTIONS_SIGNALLED_MASK unsigned int

Indicates which FP exceptions were signalled by an instruction. This uses the same encoding as the cumulative floating-point exception bitfields in FPSR.

FP_STATE

Floating point state. Fields:

FPSR unsigned int

FPSR register value.

MASK unsigned int

The mask indicating updated cumulative bits.

FREQ_CHANGED

Core frequency changed. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The instruction count of this core.

NEW_FREQ signed int

The new frequency of this core (expressed in Hz).

OLD_FREQ signed int

The old frequency of this core (expressed in Hz).

HLT

HLT instruction occurred. Fields:

IMM16 unsigned int

The 16-bit immediate value encoded in HLT instruction.

INST

Every instruction executed. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DEBUG_STATE bool

The instruction is executed in debug state.

DISASS string

Disassembly of instruction.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISSET enum

The current instruction set.

ITSTATE unsigned int

The current ITSTATE.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

MODE enum

The mode the core is in.

NS unsigned int

The core's non-secure bit.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

OPCODE unsigned int

The opcode of the instruction.

PADDR unsigned int

The physical address of the instruction.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the instruction.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the page.

PC unsigned int

The address of the instruction.

SECURITY_STATE enum

The core's security state.

SIZE unsigned int

The size of the instruction in bytes.

INST_END

Every instruction completed.

INST_START

Every instruction started. Fields:

DEBUG_STATE bool

The instruction is executed in debug state.

ISET enum

The current instruction set.

MODE enum

The mode the core is in.

NS enum

The current Secure State.

PC unsigned int

The address of the conditional instruction.

SECURITY_STATE enum

The current Security State.

INST_STRADDLE

Instruction straddles boundary. Fields:

DEBUG_STATE bool

The instruction is executed in debug state.

ISET enum

The current instruction set.

MODE enum

The mode the core is in.

NS enum

The current Secure State.

PC unsigned int

The address of the conditional instruction.

SECURITY_STATE enum

The current Security State.

IRQ_TAKEN

IRQ taken exception.

ISB

ISB instruction occurred. Fields:

PC unsigned int

The address of the ISB instruction.

LOAD_MULTIPLE_REGS_LIST

List of destination registers (not source) of a multiple registers load instruction. Fields:

REG string

Register to add into the list.

LOCAL_MONITOR

Local monitor activity. Fields:

PADDR unsigned int

Local Monitor Address.

State enum

State of the monitor (Open/Exclusive).

MAPPING_CREATE

A memory mapping has been created. This is an implementation detail of the model and so may change without notice. Fields:

ASID unsigned int

The ASID used during translation, or 0 if no ASID was used.

IS_VALID bool

Whether the mapping is valid or not.

PADDR unsigned int

Base physical address of region, or 0 if not known.

SIDE enum

I-side or D-side.

SIZE unsigned int

Size of the page, or 0 if not known.

VADDR unsigned int

Base virtual address of region.

VMID unsigned int

The VMID used during translation, or 0 if no VMID was used.

MAPPING_DESTROY

A memory mapping has been destroyed. This is an implementation detail of the model and so may change without notice. Fields:

ASID unsigned int

The ASID used during translation.

CAUSE enum

The cause of invalidation.

PADDR unsigned int

Base physical address of page.

SIDE enum

I-side or D-side.

SIZE unsigned int

Size of the page.

VADDR unsigned int

Base virtual address of page.

VMID unsigned int

The VMID used during translation.

MEMMAP_DEBUG_READ

Memory mapped reads to the debug registers. Fields:

ADDR unsigned int

address.

CORE_NUM unsigned int

Core number in a multi processor.

EXT bool

Whether access is from an external device (such as the DAP).

NS enum

Secure state of the access.

REG_NAME string

Name of the debug register accessed.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

MEMMAP_DEBUG_WRITE

Memory mapped writes to the debug registers. Fields:

ADDR **unsigned int**

address.

CORE_NUM **unsigned int**

Core number in a multi processor.

EXT **bool**

Whether access is from an external device (such as the DAP).

NS **enum**

Secure state of the access.

REG_NAME **string**

Name of the debug register accessed.

UNDEF **bool**

The register accessed is undefined.

VALUE **unsigned int**

The new value written.

MMU_D128_TTB_READ

This event is triggered by reads caused by a D128 translation table walk. Fields:

ABORTED **bool**

Was the walk successful.

DATA_HIGH **unsigned int**

The data upper bits read.

DATA_LOW **unsigned int**

The data lower bits read.

IPA **unsigned int**

For stage 1, the IPA of the read.

LEVEL **unsigned int**

Translation table level.

MECID **unsigned int**

Memory Encryption Context(MEC) ID, if applicable.

MPAM_PARTID **unsigned int**

MPAM Partition ID.

MPAM_PMG **unsigned int**

MPAM Performance Monitoring Group.

MPAM_SP **unsigned int**

MPAM Partition ID Space.

OUTADDR unsigned int

The output physical address.

PADDR unsigned int

The physical address of the read.

PASmem enum

Physical Address Space of the accessed memory.

PASreq enum

Physical Address Space of the requested lookup.

REGIME_EL enum

Entry matches in this translation regime.

REQUESTER enum

The requester of this table walk.

SIDE enum

Inst / Data.

STAGE unsigned int

Translation stage.

MMU_D128_TTB_WRITE

This event is triggered by writes caused by a D128 translation table walk. Fields:

ABORTED bool

Was the walk successful.

DATA_HIGH unsigned int

The data upper bits written.

DATA_LOW unsigned int

The data lower bits written.

IPA unsigned int

For stage 1, the IPA of the write.

IS_HACDBS_ACCESS bool

If the access was due to HACDBS_CLEAN.

LEVEL unsigned int

Translation table level.

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

OUTADDR unsigned int

The output physical address.

PADDR unsigned int

The physical address of the write.

PASmem enum

Physical Address Space of the accessed memory.

PASreq enum

Physical Address Space of the requested lookup.

REGIME_EL enum

Entry matches in this translation regime.

SIDE enum

Inst / Data.

STAGE unsigned int

Translation stage.

MMU_TRANS

Address translation information. Fields:

ASID unsigned int

Address space identifier.

CORE_NUM unsigned int

Core number in a multi processor.

Hyp bool

Entry matches in Hyp state only.

INNERCACHE_RA bool

Is the inner cache allocate on read.

INNERCACHE_TRANSIENT bool

Is the inner write-through transient.

INNERCACHE_TYPE enum

Inner Caching scheme (NC/MB/WA).

INNERCACHE_WA bool

Is the inner cache allocate on write.

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MEMTYPE enum

Memory type.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC enum

Is secure side supposed to access secure or nonsecure memory.

OUTERCACHE_RA bool

Is the outer cache allocate on read.

OUTERCACHE_TRANSIENT bool

Is the outer write-through transient.

OUTERCACHE_TYPE enum

Outer Caching scheme (NC/MB/WA).

OUTERCACHE_WA bool

Is the outer cache allocate on write.

PADDR unsigned int

Physical address of the access.

PAGESIZE unsigned int

Page size as log2(size).

PAS enum

PAS supposed to be accessed is secure or nonsecure memory or Root/Realm if RME is implemented.

SH enum

Shareability.

SIDE enum

Inst / Data.

STAGE1_PERM unsigned int

Stage 1 permissions, [5:3] Privileged access (XWR) [2:0] User access (XWR).

STAGE2_PERM unsigned int

Stage 2 permission mask [5:3] Privileged access (XWR) [2:0] User access (XWR).

VADDR unsigned int

Virtual address of the access.

VMID unsigned int

Virtual machine identifier.

XS bool

XS Attribute.

nG enum

Flag indicating whether ASID will be matched.

MMU_TTB_READ

This event is triggered by reads caused by a translation table walk. Fields:

ABORTED bool

Was the walk successful.

DATA unsigned int

The data read.

DESC_KIND enum

The encoding scheme of the descriptors used for this walk.

IPA unsigned int

For stage 1, the IPA of the read.

LEVEL unsigned int

Translation table level.

LPAE bool

Is this for an LPAE translation (*DEPRECATED*, use DESC_KIND).

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSmem bool

Non secure memory access.

NSreq bool

Non secure request.

OUTADDR unsigned int

The output physical address.

PADDR unsigned int

The physical address of the read.

PASmem enum

Physical Address Space of the accessed memory.

PASreq enum

Physical Address Space of the requested lookup.

REGIME_EL enum

Entry matches in this translation regime.

REQUESTER enum

The requester of this table walk.

SIDE enum

Inst / Data.

STAGE unsigned int

Translation stage.

MMU_TTB_WRITE

This event is triggered by writes caused by a translation table walk. Fields:

ABORTED bool

Was the walk successful.

DATA unsigned int

The data written.

DESC_KIND enum

The encoding scheme of the descriptors used for this walk.

IPA unsigned int

For stage 1, the IPA of the write.

IS_HACDBS_ACCESS bool

If the access was due to HACDBS_CLEAN.

LEVEL unsigned int

Translation table level.

LPAE bool

Is this for an LPAE translation (*DEPRECATED*, use DESC_KIND).

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSmem bool

Non secure memory access.

NSreq bool

Non secure request.

OUTADDR unsigned int

The output physical address.

PADDR unsigned int

The physical address of the write.

PASmem enum

Physical Address Space of the accessed memory.

PASreq enum

Physical Address Space of the requested lookup.

REGIME_EL enum

Entry matches in this translation regime.

SIDE enum

Inst / Data.

STAGE unsigned int

Translation stage.

MODE_CHANGE

Mode change. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

MODE enum

The new mode.

NON_SECURE enum

The core's new non-secure bit.

OLD_MODE enum

The old mode.

SECURITY_STATE enum

The new security state.

NOT_TAG_CHECKED_INST

MTE Tag not checked for current instruction.

PERIODIC

Called for every quantum. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The instruction count of this CPU.

PC unsigned int

The address of the next instruction to be executed on this CPU.

PMU_COUNTER_OVERFLOW

PMU counter overflow. Fields:

COUNTER_GROUP enum

Counter group.

COUNTER_RANGE enum

Counter range.

EVENT_ID unsigned int

Event Identifier, field is valid only for Event counter group.

INDEX unsigned int

Counter index (as selected by PMSELR), field is valid only for Event/Cycle counter groups.

INTERRUPT bool

Is interrupt enabled on overflow for this counter.

POTINDEX_UPDATE

Value of POT index for current pc changed. Fields:

NEW_POTINDEX unsigned int

New POTIndex.

OLD_POTINDEX unsigned int

Old POTIndex.

PREFETCH_MEMORY64

Prefetch from PRFM or PRFUM instructions. Fields:

PRFOP enum

Prefetch hint.

VADDR unsigned int

Virtual address of the location that should be prefetched.

PRELOAD64_REQUEST

Preload requests from PRFM or PRFUM instructions. Fields:

PRFOP enum

Prefetch hint.

VADDR unsigned int

Virtual address of the location that should be prefetched.

PRELOAD_DATA

Data preload from PLD instruction. Fields:

VADDR unsigned int

Virtual address of the data that should be preloaded.

PRELOAD_INST

Instruction preload from PLI instruction. Fields:

VADDR unsigned int

Virtual address of the instruction that should be preloaded.

PRE_CORE_LOAD

Trace just before a core load. Fields:

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

PRE_CORE_STORE

Trace just before a core store. Fields:

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

PRE_TTB_READ

This event is triggered before reads caused by a translation table walk. Fields:

ADDR unsigned int

The physical address of the read.

PSTATE

PSTATE fields update. Fields:

ALLINT unsigned int

All interrupt mask bit.

BTYPE unsigned int

Branch target identification bit.

CORE_NUM unsigned int

Core number in a multi processor.

DAIF unsigned int

Exception mask bits.

DIT unsigned int

Data Independent Timing bit.

EL unsigned int

Current Exception level field.

EXLOCK unsigned int

Exception return state lock bit.

IL unsigned int

Illegal Execution state bit.

NZCV unsigned int

Condition flags.

PACM unsigned int

PACM bit.

PAN unsigned int

Privileged Access Never state bit.

PM unsigned int

PMU exception mask bit.

PPEND unsigned int

PMU exception pending bit.

SM unsigned int

StreamingMode status bit.

SP unsigned int

Stack pointer register selection bit.

SS unsigned int

Software Step bit.

SSBS unsigned int

Speculative Store Bypass Safe bit.

TCO unsigned int

Tag Check Override bit.

TOUCHED unsigned int

Mask of the fields that have been touched since this trace has last been emitted. Note that touched does not necessarily mean the value changed. The index of the field is the bit index in the mask plus the index of this field plus one.

UAO unsigned int

User Access Override bit.

UINJ unsigned int

Undefined Instruction exceptions bit.

ZA unsigned int

ZA storage and ZT0 register enable bit.

nRW unsigned int

Current Execution state bit.

PSTATE_PACTIVE_OPMODE_TRANSITION

Trace pactive and Opmode mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Operating mode after the transition.

OLD_STATE enum

Operating mode before the transition.

PSTATE_PACTIVE_OPMODE_TRANSITION_DENIED

Trace the denied pactive and Opmode mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Operating mode after the transition.

OLD_STATE enum

Operating mode before the transition.

REASON string

Reason for the transition being denied.

PSTATE_PACTIVE_POWER_TRANSITION

Trace pactive and power mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Power mode after the transition.

OLD_STATE enum

Power mode before the transition.

PSTATE_PACTIVE_POWER_TRANSITION_DENIED

Trace the denied pactive and power mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Power mode after the transition.

OLD_STATE enum

Power mode before the transition.

REASON string

Reason for the transition being denied.

PSTATE_PACTIVE_THREAD_OPMODE_TRANSITION

Trace pactive and Opmode mode transitions for threads on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Operating mode after the transition.

OLD_STATE enum

Operating mode before the transition.

QUANTUM_START

Called at the start of every quantum.

RANGE_PRELOAD_REQUEST

Preload requests from RPRFM. Fields:

COUNT unsigned int

Number of blocks of data to access.

LENGTH unsigned int

Number of contiguous bytes to be accessed in each block.

OP unsigned int

Range prefetch operation .

REUSE unsigned int

Maximum number of bytes to be accessed before next RPRFM .

STRIDE signed int

Stride to advance after accessing LENGTH bytes.

VADDR unsigned int

Base virtual address of range prefetch request.

RUN_STATE

Run state transition. Fields:

INST_COUNT unsigned int

Ticks count at point of transition.

NEW enum

New run state.

OLD enum

Old run state.

SBP_SAMPLE_COLLISION

DISPLAY Statistical profiling samples collided .

SBP_SAMPLE_DISCARDED

DISPLAY Operation discarded by statistical profiling unit when configured in discard mode.
Fields:

BUFFER_POINTER unsigned int

Value of buffer pointer.

DATA unsigned int

Data discarded.

SIZE unsigned int

Size of sample, in bytes.

SBP_SAMPLE_END

DISPLAY Statistical profiling of an operation ended.

SBP_SAMPLE_POP

DISPLAY Operation which might be sampled by statistical profiling.

SBP_SAMPLE_RECORDED

DISPLAY Operation sampled by statistical profiling has been written to the buffer at %{BUFFER_ADDRESS} with size %{SIZE} and data %{DATA}. Fields:

BUFFER_ADDRESS unsigned int

Virtual Address of start of buffer write.

DATA unsigned int

Data written to the buffer.

SIZE unsigned int

Size of sample written to buffer, in bytes.

SBP_SAMPLE_START

DISPLAY Statistical profiling of an operation started.

SBP_SAMPLE_TAKEN

DISPLAY Operation sampled by statistical profiling after filtering.

SEMIHOSTING_CALL

Call of a semihost function occurred. Fields:

PC unsigned int

The program counter after the semihosting call.

SEMIHOSTING_PRECALL

About to call semihost. Fields:

EL2 bool

Use EL2 translation regime.

NS bool

Is Non-Secure.

REG_WIDTH unsigned int

The current register width in bytes.

SIGNAL

External signal state change. Fields:

SIGNAL enum

Signal that changed.

STATE bool

Signal asserted state.

SOFTWARE_STEP

Return the Debug Software Step state. Fields:

STATE enum

Software Step state.

SPSR

SPSR change. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

MODE enum

Which of the banked SPSR registers is written.

OLD_VALUE unsigned int

The old SPSR value.

VALUE unsigned int

The new SPSR value.

START_COMPILE

Compilation started. Fields:

VADDR unsigned int

Instruction where compilation begins.

STG_MULTIPLE_MICRO_OPS_ADDRESS_RANGE

Address range for MTE stores with multiple uops. Fields:

ADDR_END unsigned int

Final address.

ADDR_START unsigned int

Starting address.

STORE_MULTIPLE_REGS_ADDRESS_RANGE

Address range for stores with multiple registers. Fields:

ADDR_END unsigned int

Final address.

ADDR_START unsigned int

Starting address.

SUBOPTIMAL_LDST_RETIRED

suboptimal load/store. Fields:

ACCESS_SIZE unsigned int

Log2 of the access width in bytes used for alignment checking (1: half-word, 2: word ...).

OPTIMAL_ALIGN_SELECTOR unsigned int

Selector to decide the optimal alignment size (1: access width, 2: 4-bytes, 3: 8-bytes, ..., 12: 4-kbytes).

VADDR unsigned int

The virtual address of the access.

SYNC

Called for every synchronization, that is every quantum break. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The instruction count of this CPU.

LOCAL_QUANTUM unsigned int

The local quantum of this CPU.

LOCAL_TIME unsigned int

The local time of this CPU.

SYNC_REASON enum

Reason for sync.

SYSCALL

System call instruction executed. Fields:

IMM unsigned int

Immediate value of the system call instruction.

TYPE enum

System call type.

VADDR unsigned int

Instruction that caused the system call.

SYSREG_READ128

System Coprocessor 128 bit register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

REGNUM enum

Internal register number.

REGNUM_PHYS enum

Internal register number physically accessed (e.g. in presense of VHE redirection).

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_READ64

System Coprocessor register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REGNUM enum

Internal register number.

REGNUM_OPERAND enum

Register number corresponding to the instruction operand.

REGNUM_PHYS enum

Internal register number physically accessed (e.g. in presense of VHE redirection).

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_UPDATE128

Triggers when the system updates a register. Fields:

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG enum

Register number.

UNKNOWN unsigned int

Bits of the register which became unknown.

VALUE unsigned int

Value written to the register.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_UPDATE32

Triggers when the system updates a register. Fields:

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG enum

Register number.

UNKNOWN unsigned int

Bits of the register which became unknown.

VALUE unsigned int

Value written to the register.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_UPDATE64

Triggers when the system updates a register. Fields:

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG enum

Register number.

UNKNOWN unsigned int

Bits of the register which became unknown.

VALUE unsigned int

Value written to the register.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_WRITE128

System Coprocessor 128 bit register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

REGNUM enum

Internal register number.

REGNUM_PHYS enum

Internal register number physically accessed (e.g. in presense of VHE redirection).

UNDEF bool

The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_WRITE64

System Coprocessor register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REGNUM enum

Internal register number.

REGNUM_OPERAND enum

Register number corresponding to the instruction operand.

REGNUM_PHYS enum

Internal register number physically accessed (e.g. in presense of VHE redirection).

UNDEF bool

The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

UNALIGNED_LDST_RETIRE

Processor unaligned load/store. Fields:

ACCESS_SIZE unsigned int

Log2 of the access width in bytes used for alignment checking (1=>half-word, 2->word ...).

NUMBER_OF_BEATS unsigned int

Number of accesses (beats) in this burst (Total size in bytes is ACCESS_SIZE * NUMBER_OF_BEATS).

VADDR unsigned int

The virtual address of the access.

VFP_D_REGS

VFP/NEON D 64 bit register write. Fields:

ALIASING enum

Type of register aliasing used.

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a write to the corresponding bit in the register. For example, 0x1 would indicate a write to VALUE[0].

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

VFP_D_REGS_READ

VFP/NEON D 64 bit register read. Fields:

ID unsigned int

The register number.

VALUE **unsigned int**

The current value of the register.

VFP_Q_REGS

VFP/NEON Q 128 bit register write. Fields:

ALIASING **enum**

Type of register aliasing used.

CORE_NUM **unsigned int**

Core number in a multi processor.

ID **unsigned int**

The register number.

MASK **unsigned int**

Each bit indicates a write to the corresponding byte in the register. For example, 0x1 would indicate a write to VALUE[7:0].

OLD_VALUE **unsigned int**

The old value overwritten.

VALUE **unsigned int**

The new value written to the register.

VFP_Q_REGS_READ

VFP/NEON Q 128 bit register read. Fields:

ID **unsigned int**

The register number.

MASK **unsigned int**

Each bit indicates a read to the corresponding byte in the register. For example, 0x1 would indicate a read to VALUE[7:0].

VALUE **unsigned int**

The current value of the register.

VFP_SYS_REGS

Writes to the VFP/NEON units system registers. Fields:

CORE_NUM **unsigned int**

Core number in a multi processor.

ID **enum**

Which VFP system register is written.

MASK **unsigned int**

The mask indicating updated cumulative bits.

OLD_VALUE **unsigned int**

The register's old value overwritten.

VALUE unsigned int

The new value written to the VFP system register.

VFP_S_REGS

VFP/NEON S 32 bit register write. Fields:

ALIASING enum

Type of register aliasing used.

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a write to the corresponding bit in the register. For example, 0x1 would indicate a write to VALUE[0].

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

VFP_S_REGS_READ

VFP/NEON S 32 bit register read. Fields:

ID unsigned int

The register number.

VALUE unsigned int

The current value of the register.

VIRTUAL_TAG_ALLOCATED

Virtual tag information when allocated. Fields:

TAG_VADDR unsigned int

Virtual tag virtual address.

VADDR unsigned int

Data Virtual address.

VIRTUAL_TAG unsigned int

Virtual tag value.

WAYPOINT

Signals end of basic block execution. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ISET enum

Origin instruction set.

IS_COND bool

Indicates if this is a conditional waypoint.

PC unsigned int

Origin address (or 0 if unavailable).

TAKEN bool

Indicates if this waypoint was taken.

TARGET unsigned int

Destination address.

TARGET_ISET enum

Destination instruction set.

WFE_END

WFE ended. Fields:

INST_COUNT unsigned int

Ticks count when leaving WFE.

WFE_EVENT_REGISTER

WFE event register status: set/clear, reason. Fields:

INST_COUNT unsigned int

Ticks count.

REASON enum

Reason for set/clear. Only REASON==0 (Cleared by WFE) clears the bit, all other reasons set it.

WFE_IGNORED

WFE ignored. Fields:

EVENT bool

This WFE was ignored because the event register was set.

INST_COUNT unsigned int

Ticks count when ignoring WFE.

TRAPPED bool

This WFE was trapped.

WFE_START

WFE entered. Fields:

INST_COUNT unsigned int

Ticks count when entering WFE.

WFI_END

WFI ended. Fields:

INST_COUNT **unsigned int**

Ticks count when leaving WFI.

WFI_IGNORED

WFI ignored. Fields:

DISABLED **bool**

This WFI was ignored because WFI is disabled.

INST_COUNT **unsigned int**

Ticks count when ignoring WFI.

REASON **enum**

specifies reason why WFI trace was ignored.

TRAPPED **bool**

This WFI was trapped.

WFI_START

WFI entered. Fields:

INST_COUNT **unsigned int**

Ticks count when entering WFI.

WFI_WAKEUP

WFI wakeup. Fields:

INST_COUNT **unsigned int**

Ticks count when WFI wakeup occurred.

REASON **enum**

Reason for wakeup.

2.33 ARM_Cortex-A76AE

This section describes the trace sources.

AA64_ASE_SVE_REGS

Modification of AArch64 Advanced SIMD or SVE Vector Registers (via SIMD or Scalar operations), SVE Predicate Registers and ZT0. Fields:

ID **enum**

The register identifier.

MASK **unsigned int**

Bitmask of the register to signal the modified bits in the VALUE field.

SM bool

Whether the PE is in Streaming Mode.

VALUE unsigned int

Value of AArch64 vector register.

ASYNC_MEMORY_FAULT

Context ID Register write. Fields:

FAULT unsigned int

Fault status present in ESR.ISS encoding for Serror in ARCH64 and present in DFSR encoding for external abort format in ARCH32.

PADDR unsigned int

Physical Address (or 0 if unavailable).

VADDR signed int

Virtual Address (or 0 if unavailable).

ATOMIC_END_ACCESS

Bus trace access for atomics. Fields:

ACCESS_FAIL bool

Memory access failed.

ADDR unsigned int

The virtual address of the access.

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

COMPARE_VALUE unsigned int

Compare value for CAS.

LOAD_VALUE unsigned int

Loaded values.

NSDESC unsigned int

The security state of the access.

OPERAND_VALUE unsigned int

Operation's operand.

OPERATION enum

Operation type.

PADDR unsigned int

The physical address of the access.

PRIV bool

Is this a privileged access?.

ATOMIC_START_ACCESS

Bus trace access for atomics. Fields:

ADDR unsigned int

The virtual address of the access.

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

COMPARE_VALUE unsigned int

Compare value for CAS.

NSDESC unsigned int

The security state of the access.

OPERAND_VALUE unsigned int

Operation's operand.

OPERATION enum

Operation type.

PADDR unsigned int

The physical address of the access.

PRIV bool

Is this a privileged access?.

ArchMsg.Error.exit_code

DISPLAY exit code: %{EXITCODE} component kind: %{KIND} reason: %{REASON}. Fields:

EXITCODE signed int

Exit code to be returned.

KIND string

Component kind that invoked the exit code trace.

REASON string

Reason why the exit code trace was invoked (optional).

ArchMsg.Warning.IgnoredWarmResetRequest

Event fires when software attempts to request warm reset through write to deprecated DBGPRCR.CWRR.

ArchMsg.Warning.RAS_ErrselrUnpredictableState

DISPLAY ERRSELR is in unpredictable state due to value greater than number_of_error_records written to it.

ArchMsg.Warning.RAS_UnpredictableStateWrite

DISPLAY %{REG_NAME} write ignored due to unpredictable ERRSEL state. Fields:

REG_NAME string

Name of register being written.

ArchMsg.Warning.ReportedRevisionRegisterRead

DISPLAY Reported revision and patch do not match that of the cpu. Fields:

ORIGINAL_VALUE unsigned int

The value the register would have based on the modelled revision.

REGISTER string

The name of the register that has ben read.

REPORTED_VALUE unsigned int

The value the register has been overridden to.

ArchMsg.Warning.Unpredictably Indexed PM Event Register

TAGS **UNPREDICTABLE** DISPLAY Event %{IsDirect:(Selector value|Register)} %{SEL} is >= %{N}. Fields:

IsDirect unsigned int

Direct Access rather than indirect.

N unsigned int

Number of Accessible Registers.

SEL unsigned int

Selector.

ArchMsg.Warning.cache_contents_unknown

A cache has been activated without being invalidated and may contain **UNKNOWN** content. DISPLAY %{SIDE:(D|I)}-Cache was enabled in %{REGIME:(SECURE|NON_SECURE|HYP)} regime but was not invalidated since power-on: cache lines could contain **UNKNOWN** content %{SIDE:(Note: D-side caches must invalidate SECURE lines even if the cache is only used by NON_SECURE code in order to ensure that unexpected dirty lines tagged SECURE are not naturally evicted)}. Fields:

REGIME unsigned int

Current translation regime.

SIDE unsigned int

0 d-side cache or unified cache, 1 i-side.

ArchMsg.Warning.dcimvac_matches_watchpoint

TAGS **UNPREDICTABLE** DISPLAY Watchpoints matching an AArch32 DCIMVA are implementation defined.

ArchMsg.Warning.decode_fieldmismatch

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with different fields %{FIELD1} and %{FIELD2}. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_fracbitsnegative

TAGS **UNPREDICTABLE** DISPLAY VCVT Instruction is unpredictable with a negative number of fraction bits.

ArchMsg.Warning.decode_initblock

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable in an IT block.

ArchMsg.Warning.decode_initblocknotlast

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable in an IT block when not the last.

ArchMsg.Warning.decode_invalid_condition

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable when the condition is not AL.

ArchMsg.Warning.decode_invalidfieldcombination

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with the values in %{FIELD1} and %{FIELD2}. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_invalidvalue

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with the value in %{FIELD}. Fields:

FIELD enum

Field name.

ArchMsg.Warning.decode_mem_hint_unallocated

TAGS **UNPREDICTABLE** DISPLAY There is no memory hint allocated to this bit pattern.

ArchMsg.Warning.decode_registermatch

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with %{FIELD1} and %{FIELD2} as the same register. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_registermismatch

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with %{FIELD1} and %{FIELD2} as different registers. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_registeroutofrange

TAGS UNPREDICTABLE DISPLAY Floating-point instruction is unpredictable indexing beyond the end of the register bank.

ArchMsg.Warning.decode_sbzsbo

TAGS UNPREDICTABLE DISPLAY Reserved bits in the instruction are not canonical.

ArchMsg.Warning.decode_transactiontoobig

TAGS UNPREDICTABLE DISPLAY Attempt to perform a memory transaction of more than 128 bytes.

ArchMsg.Warning.decode_unpred_other

TAGS UNPREDICTABLE DISPLAY Instruction is unpredictable.

ArchMsg.Warning.decode_unpreduseofpc

TAGS UNPREDICTABLE DISPLAY Instruction is unpredictable when %{FIELD} is PC. Fields:

FIELD enum

Field name.

ArchMsg.Warning.decode_unpreduseofr13

TAGS UNPREDICTABLE DISPLAY Instruction is unpredictable when %{FIELD} is R13. Fields:

FIELD enum

Field name.

ArchMsg.Warning.decode_unpreduseofr13inreglist

TAGS UNPREDICTABLE DISPLAY Instruction is unpredictable with R13 in register-list.

ArchMsg.Warning.decode_useofd16orhigher

TAGS UNPREDICTABLE DISPLAY Attempt to use register D16 or higher where it is not allowed.

ArchMsg.Warning.decode_writebackandbaseinlist

TAGS UNPREDICTABLE DISPLAY Instruction is unpredictable with write back when the base register is in the transfer list.

ArchMsg.Warning.decode_zeroregistersinlist

TAGS UNPREDICTABLE DISPLAY Instruction is unpredictable when no registers are to be transferred.

ArchMsg.Warning.recursive_branch

An instruction is performing a branch that targets the same instruction. If this was intended then a WFI might be more effective DISPLAY recursive branch detected at PC=%{PC}.

PRIMARY KEY PC. Fields:

PC unsigned int

Address of instruction causing the branch.

ArchMsg.Warning.recursive_exception

An instruction has generated an exception that targets the same instruction. DISPLAY recursive %{TYPE} exception detected at PC=%{PC} (CPSR=%{CPSR}). PRIMARY KEY PC. Fields:

CPSR unsigned int

Processor state.

PC unsigned int

Address of instruction causing the exception.

TYPE enum

Type of exception.

ArchMsg.Warning.reentrant_vector_catch

TAGS **UNPREDICTABLE** DISPLAY Vector catch debug event on Prefetch or Data abort vector %{ADDR}. Fields:

ADDR unsigned int

vector address.

ArchMsg.Warning.reserved_it_state

TAGS **UNPREDICTABLE** DISPLAY Execution with reserved IT state %{ITSTATE}. Fields:

ITSTATE unsigned int

reserved IT state.

ArchMsg.Warning.tlb_contents_unknown

Translation has been enabled without invalidating the TLB. DISPLAY Translation was enabled in the %{PAS:(Secure|Non-secure|Root|Realm)} EL%{EL:d} translation regime (vmid=%{VMID:x}) without invalidating %{INVALIDITY:(|d-side |i-side |)}entries since power-on: %{INVALIDITY:(|D||)}TLB could contain **UNKNOWN** content. Fields:

EL unsigned int

Exception level of translation regime.

INVALIDITY unsigned int

Which TLB may be invalid (if TLBs are separate).

NON_SECURE bool

Whether the translation regime is non-secure.

PAS unsigned int

Physical address space of translation regime.

VMID unsigned int

Current VMID, if applicable.

ArchMsg.Warning.unknown_DLR_DSPSR

Exiting debug state with unknown DLR or DSPSR.

ArchMsg.Warning.unknown_ELR_SPSR

Returning from debug exception with unknown ELR or SPSR.

ArchMsg.Warning.unpred_FAR_on_LRCPC3_non_atomic_mem_access

TAGS **UNPREDICTABLE** Unpredictable FAR/FSTATUS on LRCPC3 pair instructions resulting in two separate memory accessesDISPLAY Unpredictable FAR on LRCPC3 non atomic memory access at %{ADDR1} with FSTATUS %{FSTATUS1} or %{ADDR2} with FSTATUS %{FSTATUS2}. Fields:

ADDR1 unsigned int

Possible FAR address 1.

ADDR2 unsigned int

Possible FAR address 2.

FSTATUS1 unsigned int

Possible fault status value 1.

FSTATUS2 unsigned int

Possible fault status value 2.

ArchMsg.Warning.unpred_async_tag_check_fail_on_sve_speculative_load_do_update_ffr

Unpredictable set to FALSE of the FFR element associated to SVE speculative load with an async tag check fail TAGS **UNPREDICTABLE**.

ArchMsg.Warning.unpredictable_a32_breakpoint

TAGS **UNPREDICTABLE** DISPLAY Unpredictable breakpoint on a A32 instruction at %{ADDR1} Offset %{OFFSET} BAS %{BAS} for a %{IS_ADDRESS_MISMATCH_BREAKPOINT:(Mismatch|Match)} Breakpoint number %{N:d} Breakpoint address %{ADDR2}. Fields:

ADDR1 unsigned int

Address.

ADDR2 unsigned int

Breakpoint address.

BAS unsigned int

BAS field.

IS_ADDRESS_MISMATCH_BREAKPOINT bool

Is a Mismatch breakpoint, not a Match breakpoint.

N unsigned int

Breakpoint number.

OFFSET unsigned int

Always has a value of zero.

ArchMsg.Warning.unpredictable_banked_register_access

TAGS **UNPREDICTABLE** DISPLAY Access to banked register (R=%{R:d} SYSm=%{SYSM:2x}) is unpredictable. Fields:

R unsigned int

R bit.

SYSM unsigned int

SYSm.

ArchMsg.Warning.unpredictable_ccfail_undef

TAGS **UNPREDICTABLE** DISPLAY An instruction %{OPCODE} at VA=%{VADDR} failed its condition codes check but would otherwise have caused a non-data-dependent trap or undefined exception. Behaviour is implementation-defined choice of exception or nop. Fields:

OPCODE unsigned int

opcode of the instruction.

VADDR unsigned int

Virtual address of the instruction being executed.

ArchMsg.Warning.unpredictable_ici_bits

TAGS **UNPREDICTABLE** DISPLAY ICI bits do not apply or are architecturally unpredictable for instruction.

ArchMsg.Warning.unpredictable_t32_breakpoint

TAGS **UNPREDICTABLE** DISPLAY Unpredictable breakpoint on a T32 instruction at %{ADDR1} Offset %{OFFSET} BAS %{BAS} for a %{IS_ADDRESS_MISMATCH_BREAKPOINT:(Mismatch|Match)} Breakpoint number %{N:d} Breakpoint address %{ADDR2}. Fields:

ADDR1 unsigned int

Address.

ADDR2 unsigned int

Breakpoint address.

BAS unsigned int

BAS field.

IS_ADDRESS_MISMATCH_BREAKPOINT bool

Is a Mismatch breakpoint, not a Match breakpoint.

N unsigned int

Breakpoint number.

OFFSET unsigned int

Indicates which halfword of a 32-bit Thumb instruction the breakpoint is set on. 0 for the first halfword or if the instruction is 16 bits wide, or 1 for the second halfword.

ArchMsg.Warning.unpredictable_t32_vector_catch

TAGS **UNPREDICTABLE** DISPLAY Unpredictable Vector catch exception on a T32 instruction at %{ADDR} Offset %{OFFSET}. Fields:

ADDR unsigned int

Address.

OFFSET unsigned int

Indicates which halfword of a 32-bit Thumb instruction the vector catch exception is set on. 0 for the first halfword or if the instruction is 16 bits wide, or 1 for the second halfword.

ArchMsg.Warning.unpredictable_watchpoint_far

TAGS **UNPREDICTABLE** DISPLAY The value for FAR/EDWAR for hitting this watchpoint may be anywhere in the range %{LOWER_BOUND} to %{UPPER_BOUND}. Fields:

LOWER_BOUND unsigned int

the lowest address accessed by the instruction that triggered the watchpoint.

UPPER_BOUND unsigned int

the highest watchpointed address accessed by that instruction.

ArchMsg.Warning.unpredictable_watchpoint_far_isncmp

DISPLAY The value for FAR/EDWAR for hitting this watchpoint may be anywhere in the range %{LOWER_BOUND} to %{UPPER_BOUND}. Fields:

LOWER_BOUND unsigned int

the lowest address accessed by the instruction that triggered the watchpoint.

UPPER_BOUND unsigned int

the highest watchpointed address accessed by that instruction.

ArchMsg.Warning.warning_AdvSIMDExpandImmUnexpectedZero

TAGS **UNPREDICTABLE** DISPLAY AdvSIMDExpandImm may treat this immediate value as **UNPREDICTABLE**.

ArchMsg.Warning.warning_ConditionalSMC

TAGS **UNPREDICTABLE** DISPLAY SMC instruction has **UNPREDICTABLE** effects when conditional and when combined with traps.

ArchMsg.Warning.warning_TLBID_out_of_range

TLBID out of range in TLBI according to the configuration of NIS, NVIS, NOS and NVOS. Fields:

NIS unsigned int

NIS.

NOS unsigned int

NOS.

NVIS unsigned int

NVIS.

NVOS unsigned int

NVOS.

TLBID unsigned int

The TLBID value used in the TLBI instruction.

ArchMsg.Warning.warning_TTL_or_TG_mismatches_current_translation_granule

TTL or TG mismatches current translation granule in TLBI VA/RVA/IPA/RIPA. Fields:

CURRENT_TG unsigned int

Current TG as indicated by TCR_ELx or VTCR_EL2.

TLBI_TG unsigned int

TG (or TTL[3:2]) used in TLBI instruction.

ArchMsg.Warning.warning_access_crosses_page_boundary

TAGS **UNPREDICTABLE** DISPLAY Access crosses page boundary.

ArchMsg.Warning.warning_access_crosses_page_boundary_has_fault

TAGS **UNPREDICTABLE** DISPLAY Access that crosses page boundary has fault(s).

ArchMsg.Warning.warning_access_crosses_page_boundary_has_fault_on_both_pages

TAGS **UNPREDICTABLE** DISPLAY Access that crosses page boundary has a fault on each page.
Fields:

ADDR1 unsigned int

FAR address on first page.

ADDR2 unsigned int

FAR address on second page.

FSTATUS1 unsigned int

Fault status on first page.

FSTATUS2 unsigned int

Fault status on second page.

ArchMsg.Warning.warning_access_crosses_page_boundary_spanning_different_memory

TAGS **UNPREDICTABLE** DISPLAY Access at address %{ADDR} crosses page from MEM_TYPE %{MEMTYPE_PAGE1} to %{MEMTYPE_PAGE2}. Fields:

ADDR unsigned int

address of access crossing page.

MEMTYPE_PAGE1 signed int

First Page's memory type (0-Normal, 1-Non-Normal).

MEMTYPE_PAGE2 signed int

Second Page's memory type (0-Normal, 1-Non-Normal).

ArchMsg.Warning.warning_access_wraps_around_memory

TAGS **UNPREDICTABLE** DISPLAY Access wraps around memory in %{WIDTH:d}bit mode.
Fields:

WIDTH unsigned int

Address width.

ArchMsg.Warning.warning_bcr_linking_status

TAGS **UNPREDICTABLE** DISPLAY DBGBCR.{BT2,BT} combination gives unpredictable behaviour.

ArchMsg.Warning.warning_bcr_mask_reserved

TAGS **UNPREDICTABLE** DISPLAY DBGBCR_ELn.MASK is set to a reserved value.

ArchMsg.Warning.warning_bvr_masked_bit_not_zero

TAGS **UNPREDICTABLE** DISPLAY DBGBVRn_EL1.VA contains a bit masked by DBGBCRn_EL1.MASK that is not zero.

ArchMsg.Warning.warning_ccsidr_unimplemented_level

TAGS **UNPREDICTABLE** DISPLAY CCSIDR read while CSSELR %{CSSELR:x} points to an unimplemented cache level. Fields:

CSSELR unsigned int

current effective value of CSSELR.

ArchMsg.Warning.warning_change_to_ns_when_tge_set

TAGS **UNPREDICTABLE** DISPLAY Attempting to change NS to 1 when HCR.TGE = 1.

ArchMsg.Warning.warning_contiguous_bit_check_abort

TAGS **UNPREDICTABLE** DISPLAY An abort occurred whilst attempting to check TLB entry is contiguous at %{ENTRY_ADDR}. Fields:

ENTRY_ADDR unsigned int

address of conflicting TLB entry.

ArchMsg.Warning.warning_contiguous_bit_error

TAGS **UNPREDICTABLE** DISPLAY TLB entry at %{ENTRY_ADDR} was not contiguous with entry at %{CONTIG_ADDR} Contents are %{ENTRY_DATA} but expected %{CONTIG_DATA}. Fields:

CONTIG_ADDR unsigned int

address of the TLB first read, with which this entry is expected to be contiguous.

CONTIG_DATA unsigned int

expected contents based on the entry at CONTIG_ADDR.

ENTRY_ADDR unsigned int

address of conflicting TLB entry.

ENTRY_DATA unsigned int

contents of conflicting TLB entry.

ArchMsg.Warning.warning_contiguous_bit_error_D128

TAGS **UNPREDICTABLE** DISPLAY TLB entry at %{ENTRY_ADDR} was not contiguous with entry at %{CONTIG_ADDR} Contents are %{ENTRY_DATA_HI} %{ENTRY_DATA_LO} but expected %{CONTIG_DATA}. Fields:

CONTIG_ADDR unsigned int

address of the TLB first read, with which this entry is expected to be contiguous.

CONTIG_DATA unsigned int

expected contents based on the entry at CONTIG_ADDR.

ENTRY_ADDR unsigned int

address of conflicting TLB entry.

ENTRY_DATA_HI unsigned int

Higher bit contents of conflicting TLB entry.

ENTRY_DATA_LO unsigned int

Lower bit contents of conflicting TLB entry.

ArchMsg.Warning.warning_cp10_cp11_mismatch

TAGS **UNPREDICTABLE** DISPLAY Unpredictable configuration of CP10 and CP11 controls at EL
%{EL:d}. Fields:

EL unsigned int

EL owning the control.

ArchMsg.Warning.warning_cp10_cp11_reserved_value

TAGS **UNPREDICTABLE** DISPLAY Unpredictable configuration of CPACR CP10 (%{CP10}) or
CP11 (%{CP11}). Fields:

CP10 unsigned int

CP10 access.

CP11 unsigned int

CP11 access.

ArchMsg.Warning.warning_csselr_level_out_of_range

TAGS **UNPREDICTABLE** DISPLAY CSSELR.Level out of range, written value %{LEVEL} with only
%{IMPLEMENTED:d} levels implemented. Fields:

IMPLEMENTED unsigned int

number of cache levels implemented.

LEVEL unsigned int

written value of CSSELR.Level.

TYPE enum

The type of cache selected by CSSELR.InD and TnD.

ArchMsg.Warning.warning_debug_flow_control_bits_not_obeyed

TAGS **UNPREDICTABLE** DISPLAY Flow control bits for %{REGISTER} were not obeyed, causing
%{ERROR}. Fields:

ERROR enum

type of error resulting.

REGISTER enum

register under consideration.

ArchMsg.Warning.warning_debug_register_access_during_reset

TAGS **UNPREDICTABLE** DISPLAY Unpredictable %{IS_WRITE:(read|write)} access to debug register offset %{OFFSET} during reset. Fields:

IS_WRITE unsigned int

Write Not Read.

OFFSET unsigned int

Register Offset.

ArchMsg.Warning.warning_decode_cps_inconsistent_fields

TAGS **UNPREDICTABLE** DISPLAY The mode setting fields in the CPS instruction are inconsistent.

ArchMsg.Warning.warning_decode_invalid_state

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable in current exception-level/security state.

ArchMsg.Warning.warning_default_cacheable_mmu_on

TAGS **UNPREDICTABLE** DISPLAY Default cacheable is enabled (HCR.DC = 1) while MMU is enabled.

ArchMsg.Warning.warning_default_cacheable_vmmu_off

TAGS **UNPREDICTABLE** DISPLAY Default cacheable is enabled (HCR.DC = 1) while HCR.VM = 0.

ArchMsg.Warning.warning_deprecated_wvr_bit_2_set

TAGS **UNPREDICTABLE** DISPLAY DBGWVR%{N:d}_EL1=%{WVR:h} has bit 2 set, deprecated in ARMv8. Fields:

N unsigned int

index.

WVR unsigned int

DBGWVR.

ArchMsg.Warning.warning_eret_while_software_step_active_pending

Missing ISB between setting MDSCR_EL1.SS and ERET TAGS **UNPREDICTABLE**.

ArchMsg.Warning.warning_exclusive_to_non_normal

TAGS **UNPREDICTABLE** DISPLAY Exclusive Access to Non-Normal Memory %{ADDR}. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_exclusive_to_non_writeback

TAGS **UNPREDICTABLE** DISPLAY Exclusive Access to Non-Writeback-Cacheable Memory %{ADDR}. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_execute_from_device_memory

TAGS **UNPREDICTABLE** DISPLAY An attempt was made to execute from Device memory.

ArchMsg.Warning.warning_illegal_cpsr_mode

TAGS **UNPREDICTABLE** DISPLAY Writing an illegal or unimplement mode ({NEW_MODE:h}) to CPSR was unpredictable. Fields:

NEW_MODE unsigned int

New value of CPSR.M.

ArchMsg.Warning.warning_illegal_srs_mode

TAGS **UNPREDICTABLE** DISPLAY Illegal or **UNPREDICTABLE** mode ({MODE:h}) used for SRS instruction. Fields:

MODE unsigned int

mode for Banked SP.

ArchMsg.Warning.warning_implementation_defined_read_debug_register_in_SCS

Read Access to Debug registers from the processor is **IMPLEMENTATION DEFINED** TAGS **IMP_DEF**. Fields:

PPB_OFFSET unsigned int

debug register offset.

ArchMsg.Warning.warning_implementation_defined_sequential_security_transitions_supported

The behavior of sequential instruction fetches that cross from non-secure to secure memory and contain an SG instruction is **CONSTRAINED UNPREDICTABLE** TAGS **UNPREDICTABLE**.

ArchMsg.Warning.warning_implementation_defined_stack_limit_check_supported

It is **IMPLEMENTATION DEFINED** whether stack pointer limit checking is performed for this instruction TAGS **IMP_DEF**. Fields:

ADDRESS unsigned int

Address of instruction.

OPCODE unsigned int

instruction opcode.

ArchMsg.Warning.warning_implementation_defined_taggednonsharedmemorytype

TAGS **IMP_DEF** DISPLAY Memory attribute has Tagged, non shared memory type; it is **IMPLEMENTATION DEFINED** whether the memory location is treated as Tagged or non-Tagged.

ArchMsg.Warning.warning_implementation_defined_write_debug_register_in_SCS

Write Access to Debug registers from the processor is **IMPLEMENTATION DEFINED** TAGS **IMP_DEF**. Fields:

DATA unsigned int

data attempted to be written.

PPB_OFFSET unsigned int

debug register offset.

ArchMsg.Warning.warning_invalid_tcr_granule

TAGS UNPREDICTABLE DISPLAY TCR.TG{%TG_ID:(0|1)} bits have been set to a granule size {%REQUEST:d}K not implemented - using {%SUBSTITUTE:d}K. Fields:

REQUEST signed int

page size requested (or 0 for reserved).

SUBSTITUTE signed int

best guess available page size.

TG_ID bool

bits TG0 or TG1.

ArchMsg.Warning.warning_load_multiple_user_registers_from_user_mode

TAGS UNPREDICTABLE DISPLAY An LDM(user registers) instruction executed from user mode.

ArchMsg.Warning.warning_mte_in_mem_attr_not_supported

TAGS UNPREDICTABLE DISPLAY Memory attribute {%ATTR} has Tagged memory type but Armv8.5-MTE is not implemented. Fields:

ATTR unsigned int

Memory attribute.

ArchMsg.Warning.warning_reserved_breakpoint_state_match

TAGS UNPREDICTABLE DISPLAY {%IS_BREAKPOINT:(Watchpoint|Breakpoint)} programmed with a reserved combination of HMC, SSC and {%IS_BREAKPOINT:(PAC|PMC)}. Fields:

IS_BREAKPOINT unsigned int

Is a breakpoint, not a watchpoint.

ArchMsg.Warning.warning_shareability

TAGS UNPREDICTABLE DISPLAY Unpredictable: combination of the 1st and 2nd stages of translation is Normal Inner Non-Cacheable, Outer Non-Cacheable.

ArchMsg.Warning.warning_software_step_set_while_enabled

TAGS UNPREDICTABLE DISPLAY MDSCR_EL1.SS set to 1 while software step debug exceptions are enabled.

ArchMsg.Warning.warning_thumb_instruction_wraps_around_memory

TAGS UNPREDICTABLE DISPLAY Thumb instruction wraps around memory.

ArchMsg.Warning.warning_ttbr_sbz_bits_are_not_zero

TAGS UNPREDICTABLE DISPLAY {%TTBR}[{%MSB:d}:0] should be zero, but are not. Fields:

MSB unsigned int

x-1.

TTBR enum

which TTBR register.

ArchMsg.Warning.warning_unaligned_address_dbgdtrrx_write

TAGS **UNPREDICTABLE** DISPLAY Unaligned address %{ADDR} in X0/R0 when DBGDTRRX written in memory access mode. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unaligned_address_dbgdtrtx_read

TAGS **UNPREDICTABLE** DISPLAY Unaligned address %{ADDR} in X0/R0 when DBGDTRTX read in memory access mode. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unaligned_to_device

TAGS **UNPREDICTABLE** DISPLAY Unaligned Access to Device Memory %{ADDR}. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unaligned_to_strongly_ordered

TAGS **UNPREDICTABLE** DISPLAY Unaligned Access to Strongly Ordered Memory %{ADDR}. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unknown_sau_rnr

TAGS **UNPREDICTABLE** DISPLAY SAU_RNR was set to an unsupported value.

ArchMsg.Warning.warning_unpredictable_AIRCR_PRIS_and_BFHFNMINs

TAGS **UNPREDICTABLE** DISPLAY The effect of setting both AIRCR.BFHFNMINs and AIRCR.PRIS to 1 is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_AIRCR_VECTCLRACTIVE_when_not_in_debug

TAGS **UNPREDICTABLE** DISPLAY The effect of writing a 1 to AIRCR.VECTCLRACTIVE if the processor is not halted in Debug state is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_AIRCR_VECTRESET_and_SYSRESETREQ

TAGS **UNPREDICTABLE** DISPLAY When the processor is halted in Debug state, if a write to the register writes a 1 to both VECTRESET and SYSRESETREQ, the behavior is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_AIRCR_VECTRESET_when_not_in_debug

TAGS **UNPREDICTABLE** DISPLAY The effect of writing a 1 to AIRCR.VECTRESET if the processor is not halted in Debug state is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_AIRCR_incorrect_VKEY

TAGS **UNPREDICTABLE** DISPLAY The value 0x05FA must be written to AIRCR.VKEY, otherwise the register write is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_EXC_RETURN_Reserved_Bit

EXC_RETURN[23:7] are reserved with the special condition that all bits should be written as one. Values other than all 1s are **UNPREDICTABLE**. TAGS **UNPREDICTABLE**. Fields:

EXC_RETURN unsigned int

Special PC value.

SBO_MASK unsigned int

Which bits Should Be One to not UNPRED.

ArchMsg.Warning.warning_unpredictable_PMEVTYPEPn_ELO_VS_reserved_value

TAGS **UNPREDICTABLE** DISPLAY Unpredictable reserved value of PMEVTYPEN<n>_ELO.VS.

ArchMsg.Warning.warning_unpredictable_change_to_DEMCR_MON_STEP_at_insufficient_priority

TAGS **UNPREDICTABLE** DISPLAY The effect of changing DEMCR.MON_STEP at an execution priority that is lower than the priority of the DebugMonitor exception is **CONSTRAINED UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_change_to_priority_of_active_exception

TAGS **UNPREDICTABLE** DISPLAY Changing the priority of an active exception is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_clear_lspact

TAGS **UNPREDICTABLE** DISPLAY **UNPREDICTABLE** state on context switch. Fields:

DRVEXC_TAKEN unsigned int

Derived exception taken, not pended.

REACHED_LAST_STORE unsigned int

Only the last faulted.

ArchMsg.Warning.warning_unpredictable_dbg_exit_unaligned_dlr

TAGS **UNPREDICTABLE** DISPLAY Debug exit to AArch32 with DLR[0] set to 1. PC[0] can be set to 0 or to DLR[0].

ArchMsg.Warning.warning_unpredictable_exception_catch

TAGS **UNPREDICTABLE** DISPLAY Generation of an exception catch event is unpredictable.

ArchMsg.Warning.warning_unpredictable_exception_return_inconsistent_state

TAGS **UNPREDICTABLE** DISPLAY state on exception return is unpredictable.

ArchMsg.Warning.warning_unpredictable_exception_return_instruction

TAGS **UNPREDICTABLE** DISPLAY exception return instruction is unpredictable.

ArchMsg.Warning.warning_unpredictable_in_debug_state

TAGS **UNPREDICTABLE** DISPLAY Instruction is **UNPREDICTABLE** when executed in debug state.

ArchMsg.Warning.warning_unpredictable_mair_encoding

TAGS **UNPREDICTABLE** DISPLAY Memory attribute %{ATTR} has invalid MAIR encoding; this may lead to unpredictable results. Fields:

ATTR unsigned int

Memory attribute.

ArchMsg.Warning.warning_unpredictable_pmu_counter_access

TAGS **UNPREDICTABLE** DISPLAY Access to PMU counters from non-secure EL0 or EL1 is **UNPREDICTABLE** with MDCR_EL2.HPMN set to 0.

ArchMsg.Warning.warning_unpredictable_prioritization_breakpoint_match_vector_catch

TAGS **UNPREDICTABLE** DISPLAY Two events with the same priority occurred at the same instruction: Address Matching Vector Catch debug event, and Breakpoint debug event. It is constrained unpredictable which is taken.

ArchMsg.Warning.warning_unpredictable_stack_selection

TAGS **UNPREDICTABLE** DISPLAY **UNPREDICTABLE** state on context switch.

ArchMsg.Warning.warning_unpredictable_tsize_out_of_range

TAGS **UNPREDICTABLE** DISPLAY %{TCR}.T%{TTBR:d}SZ (%{TSIZE:d} bits) is out of range. Must be between 25 and %{TSIZE_MAX:d}. Fields:

TCR enum

Which TCR.

TSIZE signed int

Value of TSize.

TSIZE_MAX signed int

The maximum allowed value.

TTBR unsigned int

Which TTBR region.

ArchMsg.Warning.warning_unpredictable_unaligned_pc_as_base_register

TAGS **UNPREDICTABLE** DISPLAY The PC value must be word-aligned, otherwise the behavior of the instruction is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_unaligned_pop_stack

TAGS **UNPREDICTABLE** DISPLAY **CONSTRAINED UNPREDICTABLE** unaligned stack frame %{FRAME_ADDR:x} in exception-return. Fields:

FRAME_ADDR unsigned int

unaligned stack frame address.

ArchMsg.Warning.warning_unpredictable_unaligned_to_non_writeback

TAGS **UNPREDICTABLE** DISPLAY Unaligned Access to Non-Writeback-Cacheable Memory.

ArchMsg.Warning.warning_unpredictable_vmsa_memattrib

TAGS **UNPREDICTABLE** DISPLAY Unpredictable vmsa memory attribute with texcb value programmed to %{TEXCB} when tex remap is %{TEX_REMAP}. Fields:

TEXCB unsigned int

texcb value.

TEX_REMAP bool

use tex remap.

ArchMsg.Warning.warning_unpredictable_vtcr_t0sz_sl0

TAGS **UNPREDICTABLE** DISPLAY The combination of VTCR.TOSZ=%{TOSZ:d} and VTCR.SLO=%{SLO:d} is unpredictable. Fields:

SLO signed int

VTCR.SLO, starting level for stage 2 translation table walks.

TOSZ signed int

VTCR.TOSZ, required input address range.

ArchMsg.Warning.warning_unpredictable_write_DHCSR

TAGS **UNPREDICTABLE** DISPLAY **UNPREDICTABLE** set of bit-changes in DHCSR. Fields:

NEW unsigned int

data attempted to be written.

OLD unsigned int

previous value.

ArchMsg.Warning.warning_unpredictable_write_SHCSR

TAGS **UNPREDICTABLE** DISPLAY **UNPREDICTABLE** setting a pending bit to 1 for an exception with priority >= execution priority.

ArchMsg.Warning.warning_unsupported_access_to_memory_mapped_register

TAGS **UNPREDICTABLE** DISPLAY Access to memory mapped register is unsupported.

ArchMsg.Warning.warning_user_jmcr_access

TAGS **UNPREDICTABLE** DISPLAY User %{WRITE:(write|read)} access to JMCR is unpredictable. Fields:

WRITE unsigned int

access is write.

ArchMsg.Warning.warning_wcr_mask_and_bas

TAGS **UNPREDICTABLE** DISPLAY DBGWCR_EL1.MASK is non-zero and BAS is not 0xff.

ArchMsg.Warning.warning_wcr_mask_reserved

TAGS **UNPREDICTABLE** DISPLAY DBGWCR_ELn.MASK is set to a reserved value.

ArchMsg.Warning.warning_wcr_non_configuous_bas

TAGS **UNPREDICTABLE** DISPLAY DBGWCR_EL1.BAS has non-contiguous set of ones.

ArchMsg.Warning.warning_wvr_masked_bit_not_zero

TAGS **UNPREDICTABLE** DISPLAY DBGWVRn_EL1.VA contains a bit masked by DBGWCRn_EL1.MASK that is not zero.

ArchMsg.Why.why_branch_target_exception

A branch target exception has occurred. DISPLAY branch target exception: target pc=%{PC}, target opcode=%{OPCODE}. PRIMARY KEY PC. Fields:

BTYPE unsigned int

pstate.BTYPE at time branch taken.

OPCODE unsigned int

opcode at branch target.

PC unsigned int

Virtual Address of branch target.

ArchMsg.Why.why_illegal_state

An illegal mode change has occurred. DISPLAY illegal mode change: %{MESSAGE}. Fields:

MESSAGE string

The message.

ArchMsg.warning_invalid_interval_counter_reload_value

Invalid interval counter reload value in PMSIRR_EL1.INTERVAL: software should set this to a value greater than or equal to the minimum indicated by PMSIDR_EL1.Interval.

ArchMsg.warning_invalid_spe_buffer_write_pointer

Invalid value programmed in PMBPTR_EL1.PTR or PMBLIMITR_EL1.LIMIT: Current write pointer must be at least one sample record below the write limit pointer and PMBPTR_EL1.PTR[63:56] must be equal to PMBLIMITR_EL1.LIMIT[63:56].

BRANCH_MISPREDICT

Simulating branch mispredict. Fields:

PC unsigned int

Origin address (or 0 if unavailable).

BRA_DIR

Direct branches, to immediate address. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISSET enum

The instruction set of the branch instruction.

IS_COND bool

Indicates if this is a conditional branch.

IS_HINTED bool

Whether the conditional instruction is hinted.

IS_ISB bool

Whether the instruction is ISB.

IS_LINK bool

Indicates if this is a branch with link.

PC unsigned int

The address of the branch instruction.

TARGET_ISET enum

The instruction set after the branch.

TARGET_PC unsigned int

The address the instruction branches to.

BRA_INDIR

Indirect branches, perhaps to a register. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISET enum

The instruction set of the branch instruction.

IS_COND bool

Indicates if this is a conditional branch.

IS_HINTED bool

Whether the conditional instruction is hinted.

IS_ISB bool

Whether the instruction is ISB.

IS_LINK bool

Indicates if this is a branch with link.

PC unsigned int

The address of the branch instruction.

TARGET_ISET enum

The instruction set after the branch.

TARGET_PC unsigned int

The address the instruction branches to.

BRA_INDIR_CCFAIL

Indirect branches fail, perhaps to a register.

BRA_RET_CCFAIL

Branch return fail, perhaps to a register.

CACHE_MAINTENANCE_OP

Cache Maintenance Operation. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

DATA unsigned int

Specified MVA or Set/way.

FUNCTION enum

Clean / Invalidate.

SCOPE enum

Affected region.

SIDE enum

Inst / Data.

CCFAIL

Conditional instruction condition check fail. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CCFAIL_UNC

Unconditional instruction condition check fail. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CCPASS

Conditional instruction condition check pass. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CCPASS_UNC

Unconditional instruction condition check pass. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CODE_CACHE_MAINT

Code cache maintenance. Fields:

END_INCL unsigned int

Inclusive end address for VA or PA requests.

NS bool

Non-secure world for PA.

START unsigned int

Start address for VA or PA requests.

TYPE enum

Request type.

COMPILE_BLOCK_END

Last instruction of basic block translated. Fields:

VADDR unsigned int

Address of next instruction after this basic block.

COMPILE_BLOCK_START

First instruction of basic block translated. Fields:

VADDR unsigned int

Address of first instruction in basic block.

COMPILE_INST

ARM instruction compiled. Fields:

DISASS string

Disassembly of the instruction.

ISSET enum

The instruction set of this instruction.

ITSTATE unsigned int

The ITSTATE current for the instruction.

OPCODE unsigned int

The opcode of the instruction.

PC unsigned int

The address of the instruction.

SIZE unsigned int

The size of the instruction in bytes.

CONTEXTIDR

Context ID Register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

NS bool

Secure or nonsecure banked register is accessed.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The new value written.

CONTEXT_SYNC

Called for every context synchronization event. Fields:

ARCH_CSE bool

Architectural Context Synchronization Event.

PC unsigned int

The address of the synchronization instruction.

SYSREG_SYNC bool

System register synchronization event.

CORE_ENDIAN

Core BE8 Big-Endian state changed. Fields:

BE8 bool

Core BE8 Big-Endian state.

CORE_INFO

Static processor attributes. Only triggered by a call to DumpState(). Fields:

ARCH_PROFILE enum

The architecture profile of the core.

CLUSTER_ID unsigned int

The cluster ID of this processor.

CORE_NUM unsigned int

The number of this core in an MP processor.

DETAILED_CORE_FEATURES enum

Detailed core features as a bitfield. Enum values are bitmasks to extract the info from the field.

FPU_VERSION enum

The VFP version implemented by the core.

MEM_ARCH enum

The memory architecture of the core.

MIDR_EL1 unsigned int

MIDR_EL1 value.

NUM_CORES unsigned int

The number of cores in this MP processor.

QUANTUM_SIZE unsigned int

The default quantum size of the core (in ticks).

SECURITY_FEATURES bool

Does the core have security features?.

CORE_LOADS

Processor load accesses. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ACQREL enum

Is this an acquire/release.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DATA unsigned int

The data read or written.

ELEMENT_SIZE unsigned int

Width of each element.

ENDIANNESS enum

Byte ordering of this access.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

LOCK enum

Normal, exclusive or locked access.

MECID unsigned int

Memory Encryption Context ID, if applicable.

MECID2 unsigned int

Memory Encryption Context ID of the second page, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PAGE_SIZE unsigned int

Page size as $\log_2(\text{size in bytes})$.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK, 2=Exclusive Failed, 3=Unsupported.

SIZE unsigned int

Width of the access in bytes. Only required if DATA is not traced.

TRANS bool

Is this a translated access.

VADDR unsigned int

The virtual address of the access.

CORE_REGS

Changes of the core registers R0 to R14. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number, 0 to 14.

MODE enum

Bank of the register accessed.

OLD_VALUE unsigned int

The old value overwritten.

PHYS_ID enum

The physical register accessed.

VALUE unsigned int

The new value written to the register.

CORE_REGS64

Changes of the core registers X0..X30, SP_ELn. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ID enum

The register number, 0-30 for X0-X30, >=32 for SP_ELn.

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

CORE_REGS64_READ

General purpose AArch64 register has been read X0..X30, SP_ELn. Fields:

ID enum

The register number, 0-30 for X0-X30, >=32 for SP_ELn.

VALUE unsigned int

The value read from the register.

CORE_STALLED_TRANSITION

Detect when a core is prevented from progressing PChannel states. Fields:

CURRENT_STATE enum

Current power mode of core.

REASON string

The reason a transition was stalled, and what must be done to unblock it.

TARGET_STATE enum

Desired power mode of core after transition.

CORE_STORES

Processor store accesses. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ACQREL enum

Is this an acquire/release.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DATA unsigned int

The data read or written.

ELEMENT_SIZE unsigned int

Width of each element.

ENDIANNESS enum

Byte ordering of this access.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

LOCK enum

Normal, exclusive or locked access.

MECID unsigned int

Memory Encryption Context ID, if applicable.

MECID2 unsigned int

Memory Encryption Context ID of the second page, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PAGE_SIZE unsigned int

Page size as $\log_2(\text{size in bytes})$.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK, 2=Exclusive Failed, 3=Unsupported.

SIZE unsigned int

Width of the access in bytes. Only required if DATA is not traced.

TRANS bool

Is this a translated access.

VADDR unsigned int

The virtual address of the access.

CP14_READ

System Coprocessor register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP14 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc1 unsigned int

opcode 1.

opc2 unsigned int
opcode 2.

CP14_WRITE

System Coprocessor register write. Fields:

CORE_NUM unsigned int
Core number in a multi processor.

CRm unsigned int
CRm.

CRn unsigned int
CRn.

NS bool
Secure or nonsecure banked register is accessed.

REG_NAME string
Name of the CP14 register accessed.

SECURITY_STATE enum
The current security state.

UNDEF bool
The register accessed is undefined.

UPDATED_VALUE unsigned int
Updated value of the register now it has been written.

VALUE unsigned int
The new value written.

opc1 unsigned int
opcode 1.

opc2 unsigned int
opcode 2.

CP15_READ

System Coprocessor register read. Fields:

CORE_NUM unsigned int
Core number in a multi processor.

CRm unsigned int
CRm.

CRn unsigned int
CRn.

NS bool
Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP15 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc1 unsigned int

opcode 1.

opc2 unsigned int

opcode 2.

CP15_READ64

System Coprocessor register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP15 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc unsigned int

opcode 1.

CP15_WRITE

System Control Coprocessor register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP15 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc1 unsigned int

opcode 1.

opc2 unsigned int

opcode 2.

CP15_WRITE64

System Coprocessor register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP15 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc unsigned int

opcode 1.

CPSR

CPSR change. (In AArch64 PSTATE represented in SPSR format). Fields:

CORE_NUM unsigned int

Core number in a multi processor.

OLD_VALUE unsigned int

The old CPSR value.

UNKNOWN unsigned int

Bits within the register that have unknown value.

VALUE unsigned int

The new CPSR value.

CRYPTO_SPEC

Every crypto instruction speculatively executed.

CorePowerStateChange

Core power-up state has changed. DISPLAY Core powered %{PU:(up|down)}. Fields:

PU bool

Whether the core powered up.

DATA_CACHE_SET_ALLOC_TAGS

Set allocation tags from DC GVA instruction. Fields:

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

NSDESC unsigned int

The security state of the access.

PADDR unsigned int

The physical address of the access.

PAS enum

The physical address space of the page.

RESPONSE enum

0=Aborted, 1=OK.

SIZE unsigned int

The size of memory in bytes.

TAG unsigned int

The value of the tag being set.

TAGGING_TYPE enum

Type of Memory tagging (Physical or Virtual).

TAG_VADDR unsigned int

The virtual address of the Virtual Tag if Virtual tagging is used; otherwise 0.

VADDR unsigned int

Virtual address used for setting allocation tags.

DATA_CACHE_ZERO

Zero and invalidate cache line from DC ZVA instruction. Fields:

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

NSDESC unsigned int

The security state of the access.

PADDR unsigned int

The physical address of the access.

PADDR_TAG unsigned int

If tag checked, the physical address tag for the access.

PAS enum

The physical address space of the page.

RESPONSE enum

0=Aborted, 1=OK.

SIZE unsigned int

The size of zero'd memory in bytes.

TAG_CHECKED bool

Is this access tag checked.

VADDR unsigned int

Virtual address of the zero'd cache line.

DEBUG_EVENT

Hardware debug support event. Fields:

EVENT enum

Description of event.

VALUE unsigned int

data value.

DEBUG_TARGET_ISET

Hardware debug target ISA. Fields:

TARGET_ISET enum

The target instruction set of the debug state.

DMI_ALLOCATE

DMI allocated. Fields:

ACCESS_DIRECTION enum

READ / WRITE.

PADDR unsigned int

The physical address of the DMI.

PTR unsigned int

DMI range pointer.

SIDE enum

DSIDE / ISIDE.

SIZE unsigned int

The size of DMI.

VADDR unsigned int

The virtual address of the DMI..

DMI_HIT

DMI hit, access used DMI (only fire for small DMI regions i.e. less than a full page). Fields:

ACCESS_DIRECTION enum

READ / WRITE.

PADDR unsigned int

The physical address of the DMI.

SIDE enum

DSIDE / ISIDE.

SIZE unsigned int

The size of the DMI.

VADDR unsigned int

The virtual address of the DMI..

DMI_REVOKE

DMI revoked. Fields:

ACCESS_DIRECTION enum

READ / WRITE.

PADDR unsigned int

The physical address of the DMI.

SIDE enum

DSIDE / ISIDE.

SIZE unsigned int

The size of DMI.

VADDR unsigned int

The virtual address of the DMI..

END_COMPILE

Compilation end. Fields:

END_OF_PAGE_COUNT unsigned int

Number of basic blocks exited due to page boundary since START_COMPILE.

FETCHFAIL_COUNT unsigned int

Number of basic blocks exited due to ifetch failure START_COMPILE.

INST_COUNT unsigned int

Number of instructions compiled since START_COMPILE.

NONSEQ_COUNT unsigned int

Number of basic blocks exited due to non-sequential instructions since START_COMPILE.

PAGE_STRADDLE_COUNT unsigned int

Number of basic blocks exited due to unaligned instructions crossing page since START_COMPILE.

ESB

Error Synchronization Barrier hit. Fields:

ASYNC bool

ESB is async.

IMPLICIT bool

ESB is implicit.

EXCEPTION

Exception taken. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ESR unsigned int

The value assigned to ESR when exception is taken.

ESR_EC unsigned int

The Value assigned to ESR.EC when exception is taken.

IS_PHYSICAL bool

Physical or Virtual exception.

IS_TRAP bool

Exception coming from a trap.

LR unsigned int

The value assigned to the link register.

PC unsigned int

The location where the exception occurred.

PREFERRED_RETURN unsigned int

The preferred return address for the exception.

TAKEN_LOCALLY bool

Whether exception is taken locally(same exception level).

TARGET_ISET enum

The instruction set of the exception handler code.

TARGET_PC unsigned int

The address the exception branches to.

VECTOR enum

The exception vector.

VECTOR_OFFSET unsigned int

The offset of an exception vector from the base address.

EXCEPTION_END

Every exception completed.

EXCEPTION_INFO

Trace human-readable reason of the current exception for data/instruction aborts, floating point exceptions and unknown exceptions. Fields:

DESCRIPTION string

Reason for current exception.

ESR_EC unsigned int

The value of ESR.EC when exception is taken.

EXCEPTION_RAISE

Every exception raised.

EXCEPTION_RETURN

Branches on leaving exception. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISET enum

The instruction set of the branch instruction.

IS_COND bool

Indicates if this is a conditional branch.

IS_HINTED bool

Whether the conditional instruction is hinted.

IS_ISB bool

Whether the instruction is ISB.

IS_LINK bool

Indicates if this is a branch with link.

PC unsigned int

The address of the branch instruction.

TARGET_ISET enum

The instruction set after the branch.

TARGET_PC unsigned int

The address the instruction branches to.

EXCEPTION_RETURN_PREBRANCH

Exception return event that must occur before PC and processor state have updated.

EXCEPTION_START

Every exception started.

FIQ_TAKEN

FIQ taken exception.

FP_EXCEPTIONS_SIGNALLED

Trace which FP exceptions were signalled as a part of the FP operations performed by an instruction. Fields:

FP_EXCEPTIONS_SIGNALLED_MASK unsigned int

Indicates which FP exceptions were signalled by an instruction. This uses the same encoding as the cumulative floating-point exception bitfields in FPSR.

FP_STATE

Floating point state. Fields:

FPSR unsigned int

FPSR register value.

MASK unsigned int

The mask indicating updated cumulative bits.

FREQ_CHANGED

Core frequency changed. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The instruction count of this core.

NEW_FREQ signed int

The new frequency of this core (expressed in Hz).

OLD_FREQ signed int

The old frequency of this core (expressed in Hz).

HLT

HLT instruction occurred. Fields:

IMM16 unsigned int

The 16-bit immediate value encoded in HLT instruction.

INST

Every instruction executed. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DEBUG_STATE bool

The instruction is executed in debug state.

DISASS string

Disassembly of instruction.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISSET enum

The current instruction set.

ITSTATE unsigned int

The current ITSTATE.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

MODE enum

The mode the core is in.

NS unsigned int

The core's non-secure bit.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

OPCODE unsigned int

The opcode of the instruction.

PADDR unsigned int

The physical address of the instruction.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the instruction.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the page.

PC unsigned int

The address of the instruction.

SECURITY_STATE enum

The core's security state.

SIZE unsigned int

The size of the instruction in bytes.

INST_END

Every instruction completed.

INST_START

Every instruction started. Fields:

DEBUG_STATE bool

The instruction is executed in debug state.

ISSET enum

The current instruction set.

MODE enum

The mode the core is in.

NS enum

The current Secure State.

PC unsigned int

The address of the conditional instruction.

SECURITY_STATE enum

The current Security State.

INST_STRADDLE

Instruction straddles boundary. Fields:

DEBUG_STATE bool

The instruction is executed in debug state.

ISSET enum

The current instruction set.

MODE enum

The mode the core is in.

NS enum

The current Secure State.

PC unsigned int

The address of the conditional instruction.

SECURITY_STATE enum

The current Security State.

IRQ_TAKEN

IRQ taken exception.

ISB

ISB instruction occurred. Fields:

PC unsigned int

The address of the ISB instruction.

LOAD_MULTIPLE_REGS_LIST

List of destination registers (not source) of a multiple registers load instruction. Fields:

REG string

Register to add into the list.

LOCAL_MONITOR

Local monitor activity. Fields:

PADDR unsigned int

Local Monitor Address.

State enum

State of the monitor (Open/Exclusive).

MAPPING_CREATE

A memory mapping has been created. This is an implementation detail of the model and so may change without notice. Fields:

ASID unsigned int

The ASID used during translation, or 0 if no ASID was used.

IS_VALID bool

Whether the mapping is valid or not.

PADDR unsigned int

Base physical address of region, or 0 if not known.

SIDE enum

I-side or D-side.

SIZE unsigned int

Size of the page, or 0 if not known.

VADDR unsigned int

Base virtual address of region.

VMID unsigned int

The VMID used during translation, or 0 if no VMID was used.

MAPPING_DESTROY

A memory mapping has been destroyed. This is an implementation detail of the model and so may change without notice. Fields:

ASID unsigned int

The ASID used during translation.

CAUSE enum

The cause of invalidation.

PADDR unsigned int

Base physical address of page.

SIDE enum

I-side or D-side.

SIZE unsigned int

Size of the page.

VADDR unsigned int

Base virtual address of page.

VMID unsigned int

The VMID used during translation.

MEMMAP_DEBUG_READ

Memory mapped reads to the debug registers. Fields:

ADDR unsigned int

address.

CORE_NUM unsigned int

Core number in a multi processor.

EXT bool

Whether access is from an external device (such as the DAP).

NS enum

Secure state of the access.

REG_NAME string

Name of the debug register accessed.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

MEMMAP_DEBUG_WRITE

Memory mapped writes to the debug registers. Fields:

ADDR unsigned int

address.

CORE_NUM unsigned int

Core number in a multi processor.

EXT bool

Whether access is from an external device (such as the DAP).

NS enum

Secure state of the access.

REG_NAME string

Name of the debug register accessed.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The new value written.

MMU_D128_TTB_READ

This event is triggered by reads caused by a D128 translation table walk. Fields:

ABORTED bool

Was the walk successful.

DATA_HIGH unsigned int

The data upper bits read.

DATA_LOW unsigned int

The data lower bits read.

IPA unsigned int

For stage 1, the IPA of the read.

LEVEL unsigned int

Translation table level.

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

OUTADDR unsigned int

The output physical address.

PADDR unsigned int

The physical address of the read.

PASmem enum

Physical Address Space of the accessed memory.

PASreq enum

Physical Address Space of the requested lookup.

REGIME_EL enum

Entry matches in this translation regime.

REQUESTER enum

The requester of this table walk.

SIDE enum

Inst / Data.

STAGE unsigned int

Translation stage.

MMU_D128_TTB_WRITE

This event is triggered by writes caused by a D128 translation table walk. Fields:

ABORTED bool

Was the walk successful.

DATA_HIGH unsigned int

The data upper bits written.

DATA_LOW unsigned int

The data lower bits written.

IPA unsigned int

For stage 1, the IPA of the write.

IS_HACDBS_ACCESS bool

If the access was due to HACDBS_CLEAN.

LEVEL unsigned int

Translation table level.

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

OUTADDR unsigned int

The output physical address.

PADDR unsigned int

The physical address of the write.

PASmem enum

Physical Address Space of the accessed memory.

PASreq enum

Physical Address Space of the requested lookup.

REGIME_EL enum

Entry matches in this translation regime.

SIDE enum

Inst / Data.

STAGE unsigned int

Translation stage.

MMU_TRANS

Address translation information. Fields:

ASID unsigned int

Address space identifier.

CORE_NUM unsigned int

Core number in a multi processor.

Hyp bool

Entry matches in Hyp state only.

INNERCACHE_RA bool

Is the inner cache allocate on read.

INNERCACHE_TRANSIENT bool

Is the inner write-through transient.

INNERCACHE_TYPE enum

Inner Caching scheme (NC/MB/WA).

INNERCACHE_WA bool

Is the inner cache allocate on write.

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MEMTYPE enum

Memory type.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC enum

Is secure side supposed to access secure or nonsecure memory.

OUTERCACHE_RA bool

Is the outer cache allocate on read.

OUTERCACHE_TRANSIENT bool

Is the outer write-through transient.

OUTERCACHE_TYPE enum

Outer Caching scheme (NC/MB/WA).

OUTERCACHE_WA bool

Is the outer cache allocate on write.

PADDR unsigned int

Physical address of the access.

PAGESIZE unsigned int

Page size as log2(size).

PAS enum

PAS supposed to be accessed is secure or nonsecure memory or Root/Realm if RME is implemented.

SH enum

Shareability.

SIDE enum

Inst / Data.

STAGE1_PERM unsigned int

Stage 1 permissions, [5:3] Privileged access (XWR) [2:0] User access (XWR).

STAGE2_PERM unsigned int

Stage 2 permission mask [5:3] Privileged access (XWR) [2:0] User access (XWR).

VADDR unsigned int

Virtual address of the access.

VMID unsigned int

Virtual machine identifier.

XS bool

XS Attribute.

nG enum

Flag indicating whether ASID will be matched.

MMU_TTB_READ

This event is triggered by reads caused by a translation table walk. Fields:

ABORTED bool

Was the walk successful.

DATA unsigned int

The data read.

DESC_KIND enum

The encoding scheme of the descriptors used for this walk.

IPA unsigned int

For stage 1, the IPA of the read.

LEVEL unsigned int

Translation table level.

LPAAE bool

Is this for an LPAE translation (*DEPRECATED*, use DESC_KIND).

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSmem bool

Non secure memory access.

NSreq bool

Non secure request.

OUTADDR unsigned int

The output physical address.

PADDR unsigned int

The physical address of the read.

PASmem enum

Physical Address Space of the accessed memory.

PASreq enum

Physical Address Space of the requested lookup.

REGIME_EL enum

Entry matches in this translation regime.

REQUESTER enum

The requester of this table walk.

SIDE enum

Inst / Data.

STAGE unsigned int

Translation stage.

MMU_TTB_WRITE

This event is triggered by writes caused by a translation table walk. Fields:

ABORTED bool

Was the walk successful.

DATA unsigned int

The data written.

DESC_KIND enum

The encoding scheme of the descriptors used for this walk.

IPA unsigned int

For stage 1, the IPA of the write.

IS_HACDBS_ACCESS bool

If the access was due to HACDBS_CLEAN.

LEVEL unsigned int

Translation table level.

LPAE bool

Is this for an LPAE translation (*DEPRECATED*, use DESC_KIND).

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSmem bool

Non secure memory access.

NSreq bool

Non secure request.

OUTADDR unsigned int

The output physical address.

PADDR unsigned int

The physical address of the write.

PASmem enum

Physical Address Space of the accessed memory.

PASreq enum

Physical Address Space of the requested lookup.

REGIME_EL enum

Entry matches in this translation regime.

SIDE enum

Inst / Data.

STAGE unsigned int

Translation stage.

MODE_CHANGE

Mode change. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

MODE enum

The new mode.

NON_SECURE enum

The core's new non-secure bit.

OLD_MODE enum

The old mode.

SECURITY_STATE enum

The new security state.

NOT_TAG_CHECKED_INST

MTE Tag not checked for current instruction.

PERIODIC

Called for every quantum. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The instruction count of this CPU.

PC unsigned int

The address of the next instruction to be executed on this CPU.

PMU_COUNTER_OVERFLOW

PMU counter overflow. Fields:

COUNTER_GROUP enum

Counter group.

COUNTER_RANGE enum

Counter range.

EVENT_ID unsigned int

Event Identifier, field is valid only for Event counter group.

INDEX unsigned int

Counter index (as selected by PMSELR), field is valid only for Event/Cycle counter groups.

INTERRUPT bool

Is interrupt enabled on overflow for this counter.

POTINDEX_UPDATE

Value of POT index for current pc changed. Fields:

NEW_POTINDEX unsigned int

New POTIndex.

OLD_POTINDEX unsigned int

Old POTIndex.

PREFETCH_MEMORY64

Prefetch from PRFM or PRFUM instructions. Fields:

PRFOP enum

Prefetch hint.

VADDR unsigned int

Virtual address of the location that should be prefetched.

PRELOAD64_REQUEST

Preload requests from PRFM or PRFUM instructions. Fields:

PRFOP enum

Prefetch hint.

VADDR unsigned int

Virtual address of the location that should be prefetched.

PRELOAD_DATA

Data preload from PLD instruction. Fields:

VADDR unsigned int

Virtual address of the data that should be preloaded.

PRELOAD_INST

Instruction preload from PLI instruction. Fields:

VADDR unsigned int

Virtual address of the instruction that should be preloaded.

PRE_CORE_LOAD

Trace just before a core load. Fields:

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

PRE_CORE_STORE

Trace just before a core store. Fields:

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

PRE_TTB_READ

This event is triggered before reads caused by a translation table walk. Fields:

ADDR unsigned int

The physical address of the read.

PSTATE

PSTATE fields update. Fields:

ALLINT unsigned int

All interrupt mask bit.

BTYPE unsigned int

Branch target identification bit.

CORE_NUM unsigned int

Core number in a multi processor.

DAIF unsigned int

Exception mask bits.

DIT unsigned int

Data Independent Timing bit.

EL unsigned int

Current Exception level field.

EXLOCK unsigned int

Exception return state lock bit.

IL unsigned int

Illegal Execution state bit.

NZCV unsigned int

Condition flags.

PACM unsigned int

PACM bit.

PAN unsigned int

Privileged Access Never state bit.

PM unsigned int

PMU exception mask bit.

PPEND unsigned int

PMU exception pending bit.

SM unsigned int

StreamingMode status bit.

SP unsigned int

Stack pointer register selection bit.

SS unsigned int

Software Step bit.

SSBS unsigned int

Speculative Store Bypass Safe bit.

TCO unsigned int

Tag Check Override bit.

TOUCHED unsigned int

Mask of the fields that have been touched since this trace has last been emitted. Note that touched does not necessarily mean the value changed. The index of the field is the bit index in the mask plus the index of this field plus one.

UAO unsigned int

User Access Override bit.

UINJ unsigned int

Undefined Instruction exceptions bit.

ZA unsigned int

ZA storage and ZT0 register enable bit.

nRW unsigned int

Current Execution state bit.

PSTATE_PACTIVE_OPMODE_TRANSITION

Trace pactive and Opmode mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Operating mode after the transition.

OLD_STATE enum

Operating mode before the transition.

PSTATE_PACTIVE_OPMODE_TRANSITION_DENIED

Trace the denied pactive and Opmode mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Operating mode after the transition.

OLD_STATE enum

Operating mode before the transition.

REASON string

Reason for the transition being denied.

PSTATE_PACTIVE_POWER_TRANSITION

Trace pactive and power mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Power mode after the transition.

OLD_STATE enum

Power mode before the transition.

PSTATE_PACTIVE_POWER_TRANSITION_DENIED

Trace the denied pactive and power mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Power mode after the transition.

OLD_STATE enum

Power mode before the transition.

REASON string

Reason for the transition being denied.

PSTATE_PACTIVE_THREAD_OPMODE_TRANSITION

Trace pactive and Opmode mode transitions for threads on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Operating mode after the transition.

OLD_STATE enum

Operating mode before the transition.

QUANTUM_START

Called at the start of every quantum.

RANGE_PRELOAD_REQUEST

Preload requests from RPRFM. Fields:

COUNT unsigned int

Number of blocks of data to access.

LENGTH unsigned int

Number of contiguous bytes to be accessed in each block.

OP unsigned int

Range prefetch operation .

REUSE unsigned int

Maximum number of bytes to be accessed before next RPRFM .

STRIDE signed int

Stride to advance after accessing LENGTH bytes.

VADDR unsigned int

Base virtual address of range prefetch request.

RUN_STATE

Run state transition. Fields:

INST_COUNT unsigned int

Ticks count at point of transition.

NEW enum

New run state.

OLD enum

Old run state.

SBP_SAMPLE_COLLISION

DISPLAY Statistical profiling samples collided .

SBP_SAMPLE_DISCARDED

DISPLAY Operation discarded by statistical profiling unit when configured in discard mode.
Fields:

BUFFER_POINTER unsigned int

Value of buffer pointer.

DATA unsigned int

Data discarded.

SIZE unsigned int

Size of sample, in bytes.

SBP_SAMPLE_END

DISPLAY Statistical profiling of an operation ended.

SBP_SAMPLE_POP

DISPLAY Operation which might be sampled by statistical profiling.

SBP_SAMPLE_RECORDED

DISPLAY Operation sampled by statistical profiling has been written to the buffer at %{BUFFER_ADDRESS} with size %{SIZE} and data %{DATA}. Fields:

BUFFER_ADDRESS unsigned int

Virtual Address of start of buffer write.

DATA unsigned int

Data written to the buffer.

SIZE unsigned int

Size of sample written to buffer, in bytes.

SBP_SAMPLE_START

DISPLAY Statistical profiling of an operation started.

SBP_SAMPLE_TAKEN

DISPLAY Operation sampled by statistical profiling after filtering.

SEMIHOSTING_CALL

Call of a semihost function occurred. Fields:

PC unsigned int

The program counter after the semihosting call.

SEMIHOSTING_PRECALL

About to call semihost. Fields:

EL2 bool

Use EL2 translation regime.

NS bool

Is Non-Secure.

REG_WIDTH unsigned int

The current register width in bytes.

SIGNAL

External signal state change. Fields:

SIGNAL *enum*

Signal that changed.

STATE *bool*

Signal asserted state.

SOFTWARE_STEP

Return the Debug Software Step state. Fields:

STATE *enum*

Software Step state.

SPSR

SPSR change. Fields:

CORE_NUM *unsigned int*

Core number in a multi processor.

MODE *enum*

Which of the banked SPSR registers is written.

OLD_VALUE *unsigned int*

The old SPSR value.

VALUE *unsigned int*

The new SPSR value.

START_COMPILE

Compilation started. Fields:

VADDR *unsigned int*

Instruction where compilation begins.

STG_MULTIPLE_MICRO_OPS_ADDRESS_RANGE

Address range for MTE stores with multiple uops. Fields:

ADDR_END *unsigned int*

Final address.

ADDR_START *unsigned int*

Starting address.

STORE_MULTIPLE_REGS_ADDRESS_RANGE

Address range for stores with multiple registers. Fields:

ADDR_END *unsigned int*

Final address.

ADDR_START *unsigned int*

Starting address.

SUBOPTIMAL_LDST_RETIRED

suboptimal load/store. Fields:

ACCESS_SIZE unsigned int

Log2 of the access width in bytes used for alignment checking (1: half-word, 2: word ...).

OPTIMAL_ALIGN_SELECTOR unsigned int

Selector to decide the optimal alignment size (1: access width, 2: 4-bytes, 3: 8-bytes, ..., 12: 4-kbytes).

VADDR unsigned int

The virtual address of the access.

SYNC

Called for every synchronization, that is every quantum break. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The instruction count of this CPU.

LOCAL_QUANTUM unsigned int

The local quantum of this CPU.

LOCAL_TIME unsigned int

The local time of this CPU.

SYNC_REASON enum

Reason for sync.

SYSCALL

System call instruction executed. Fields:

IMM unsigned int

Immediate value of the system call instruction.

TYPE enum

System call type.

VADDR unsigned int

Instruction that caused the system call.

SYSREG_READ128

System Coprocessor 128 bit register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

REGNUM enum

Internal register number.

REGNUM_PHYS enum

Internal register number physically accessed (e.g. in presense of VHE redirection).

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_READ64

System Coprocessor register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REGNUM enum

Internal register number.

REGNUM_OPERAND enum

Register number corresponding to the instruction operand.

REGNUM_PHYS enum

Internal register number physically accessed (e.g. in presense of VHE redirection).

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_UPDATE128

Triggers when the system updates a register. Fields:

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG enum

Register number.

UNKNOWN unsigned int

Bits of the register which became unknown.

VALUE unsigned int

Value written to the register.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_UPDATE32

Triggers when the system updates a register. Fields:

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG enum

Register number.

UNKNOWN unsigned int

Bits of the register which became unknown.

VALUE unsigned int

Value written to the register.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_UPDATE64

Triggers when the system updates a register. Fields:

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG enum

Register number.

UNKNOWN unsigned int

Bits of the register which became unknown.

VALUE unsigned int

Value written to the register.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_WRITE128

System Coprocessor 128 bit register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

REGNUM enum

Internal register number.

REGNUM_PHYS enum

Internal register number physically accessed (e.g. in presense of VHE redirection).

UNDEF bool

The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_WRITE64

System Coprocessor register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REGNUM enum

Internal register number.

REGNUM_OPERAND enum

Register number corresponding to the instruction operand.

REGNUM_PHYS enum

Internal register number physically accessed (e.g. in presense of VHE redirection).

UNDEF bool

The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

UNALIGNED_LDST_RETIRE

Processor unaligned load/store. Fields:

ACCESS_SIZE unsigned int

Log2 of the access width in bytes used for alignment checking (1=>half-word, 2->word ...).

NUMBER_OF_BEATS unsigned int

Number of accesses (beats) in this burst (Total size in bytes is ACCESS_SIZE * NUMBER_OF_BEATS).

VADDR unsigned int

The virtual address of the access.

VFP_D_REGS

VFP/NEON D 64 bit register write. Fields:

ALIASING enum

Type of register aliasing used.

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a write to the corresponding bit in the register. For example, 0x1 would indicate a write to VALUE[0].

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

VFP_D_REGS_READ

VFP/NEON D 64 bit register read. Fields:

ID unsigned int

The register number.

VALUE unsigned int

The current value of the register.

VFP_Q_REGS

VFP/NEON Q 128 bit register write. Fields:

ALIASING enum

Type of register aliasing used.

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a write to the corresponding byte in the register. For example, 0x1 would indicate a write to VALUE[7:0].

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

VFP_Q_REGS_READ

VFP/NEON Q 128 bit register read. Fields:

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a read to the corresponding byte in the register. For example, 0x1 would indicate a read to VALUE[7:0].

VALUE unsigned int

The current value of the register.

VFP_SYS_REGS

Writes to the VFP/NEON units system registers. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ID enum

Which VFP system register is written.

MASK unsigned int

The mask indicating updated cumulative bits.

OLD_VALUE unsigned int

The register's old value overwritten.

VALUE unsigned int

The new value written to the VFP system register.

VFP_S_REGS

VFP/NEON S 32 bit register write. Fields:

ALIASING enum

Type of register aliasing used.

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a write to the corresponding bit in the register. For example, 0x1 would indicate a write to VALUE[0].

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

VFP_S_REGS_READ

VFP/NEON S 32 bit register read. Fields:

ID unsigned int

The register number.

VALUE unsigned int

The current value of the register.

VIRTUAL_TAG_ALLOCATED

Virtual tag information when allocated. Fields:

TAG_VADDR unsigned int

Virtual tag virtual address.

VADDR unsigned int

Data Virtual address.

VIRTUAL_TAG unsigned int

Virtual tag value.

WAYPOINT

Signals end of basic block execution. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ISET enum

Origin instruction set.

IS_COND bool

Indicates if this is a conditional waypoint.

PC unsigned int

Origin address (or 0 if unavailable).

TAKEN bool

Indicates if this waypoint was taken.

TARGET unsigned int

Destination address.

TARGET_ISET enum

Destination instruction set.

WFE_END

WFE ended. Fields:

INST_COUNT unsigned int

Ticks count when leaving WFE.

WFE_EVENT_REGISTER

WFE event register status: set/clear, reason. Fields:

INST_COUNT unsigned int

Ticks count.

REASON enum

Reason for set/clear. Only REASON==0 (Cleared by WFE) clears the bit, all other reasons set it.

WFE_IGNORED

WFE ignored. Fields:

EVENT bool

This WFE was ignored because the event register was set.

INST_COUNT unsigned int

Ticks count when ignoring WFE.

TRAPPED bool

This WFE was trapped.

WFE_START

WFE entered. Fields:

INST_COUNT unsigned int

Ticks count when entering WFE.

WFI_END

WFI ended. Fields:

INST_COUNT unsigned int

Ticks count when leaving WFI.

WFI_IGNORED

WFI ignored. Fields:

DISABLED bool

This WFI was ignored because WFI is disabled.

INST_COUNT unsigned int

Ticks count when ignoring WFI.

REASON enum

specifies reason why WFI trace was ignored.

TRAPPED bool

This WFI was trapped.

WFI_START

WFI entered. Fields:

INST_COUNT unsigned int

Ticks count when entering WFI.

WFI_WAKEUP

WFI wakeup. Fields:

INST_COUNT unsigned int

Ticks count when WFI wakeup occurred.

REASON enum

Reason for wakeup.

2.34 ARM_Cortex-A77

This section describes the trace sources.

AA64_ASE_SVE_REGS

Modification of AArch64 Advanced SIMD or SVE Vector Registers (via SIMD or Scalar operations), SVE Predicate Registers and ZTO. Fields:

ID enum

The register identifier.

MASK unsigned int

Bitmask of the register to signal the modified bits in the VALUE field.

SM bool

Whether the PE is in Streaming Mode.

VALUE unsigned int

Value of AArch64 vector register.

ASYNC_MEMORY_FAULT

Context ID Register write. Fields:

FAULT unsigned int

Fault status present in ESR.ISS encoding for Serror in ARCH64 and present in DFSR encoding for external abort format in ARCH32.

PADDR unsigned int

Physical Address (or 0 if unavailable).

VADDR signed int

Virtual Address (or 0 if unavailable).

ATOMIC_END_ACCESS

Bus trace access for atomics. Fields:

ACCESS_FAIL bool

Memory access failed.

ADDR unsigned int

The virtual address of the access.

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

COMPARE_VALUE unsigned int

Compare value for CAS.

LOAD_VALUE unsigned int

Loaded values.

NSDESC unsigned int

The security state of the access.

OPERAND_VALUE unsigned int

Operation's operand.

OPERATION enum

Operation type.

PADDR **unsigned int**

The physical address of the access.

PRIV **bool**

Is this a privileged access?.

ATOMIC_START_ACCESS

Bus trace access for atomics. Fields:

ADDR **unsigned int**

The virtual address of the access.

ATTR **unsigned int**

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

COMPARE_VALUE **unsigned int**

Compare value for CAS.

NSDESC **unsigned int**

The security state of the access.

OPERAND_VALUE **unsigned int**

Operation's operand.

OPERATION **enum**

Operation type.

PADDR **unsigned int**

The physical address of the access.

PRIV **bool**

Is this a privileged access?.

ArchMsg.Error.exit_code

DISPLAY exit code: %{EXITCODE} component kind: %{KIND} reason: %{REASON}. Fields:

EXITCODE **signed int**

Exit code to be returned.

KIND **string**

Component kind that invoked the exit code trace.

REASON **string**

Reason why the exit code trace was invoked (optional).

ArchMsg.Warning.IgnoredWarmResetRequest

Event fires when software attempts to request warm reset through write to deprecated DBGPRCR.CWRR.

ArchMsg.Warning.RAS_ErrselrUnpredictableState

DISPLAY ERRSELR is in unpredictable state due to value greater than number_of_error_records written to it.

ArchMsg.Warning.RAS_UnpredictableStateWrite

DISPLAY %{REG_NAME} write ignored due to unpredictable ERRSELR state. Fields:

REG_NAME string

Name of register being written.

ArchMsg.Warning.ReportedRevisionRegisterRead

DISPLAY Reported revision and patch do not match that of the cpu. Fields:

ORIGINAL_VALUE unsigned int

The value the register would have based on the modelled revision.

REGISTER string

The name of the register that has ben read.

REPORTED_VALUE unsigned int

The value the register has been overridden to.

ArchMsg.Warning.Unpredictably Indexed PM Event Register

TAGS **UNPREDICTABLE** DISPLAY Event %{IsDirect:(Selector value|Register)} %{SEL} is >= %{N}. Fields:

IsDirect unsigned int

Direct Access rather than indirect.

N unsigned int

Number of Accessible Registers.

SEL unsigned int

Selector.

ArchMsg.Warning.cache_contents_unknown

A cache has been activated without being invalidated and may contain **UNKNOWN** content. DISPLAY %{SIDE:(D|I)}-Cache was enabled in %{REGIME:(SECURE|NON_SECURE|HYP)} regime but was not invalidated since power-on: cache lines could contain **UNKNOWN** content %{SIDE: (Note: D-side caches must invalidate SECURE lines even if the cache is only used by NON_SECURE code in order to ensure that unexpected dirty lines tagged SECURE are not naturally evicted)}. Fields:

REGIME unsigned int

Current translation regime.

SIDE unsigned int

0 d-side cache or unified cache, 1 i-side.

ArchMsg.Warning.dcimvac_matches_watchpoint

TAGS **UNPREDICTABLE** DISPLAY Watchpoints matching an AArch32 DCIMVA are implementation defined.

ArchMsg.Warning.decode_fieldmismatch

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with different fields %{FIELD1} and %{FIELD2}. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_fracbitsnegative

TAGS **UNPREDICTABLE** DISPLAY VCVT Instruction is unpredictable with a negative number of fraction bits.

ArchMsg.Warning.decode_initblock

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable in an IT block.

ArchMsg.Warning.decode_initblocknotlast

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable in an IT block when not the last.

ArchMsg.Warning.decode_invalid_condition

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable when the condition is not AL.

ArchMsg.Warning.decode_invalidfieldcombination

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with the values in %{FIELD1} and %{FIELD2}. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_invalidvalue

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with the value in %{FIELD}. Fields:

FIELD enum

Field name.

ArchMsg.Warning.decode_mem_hint_unallocated

TAGS **UNPREDICTABLE** DISPLAY There is no memory hint allocated to this bit pattern.

ArchMsg.Warning.decode_registermatch

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with %{FIELD1} and %{FIELD2} as the same register. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_registermismatch

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with %{FIELD1} and %{FIELD2} as different registers. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_registeroutofrange

TAGS **UNPREDICTABLE** DISPLAY Floating-point instruction is unpredictable indexing beyond the end of the register bank.

ArchMsg.Warning.decode_sbzsbo

TAGS **UNPREDICTABLE** DISPLAY Reserved bits in the instruction are not canonical.

ArchMsg.Warning.decode_transactiontoobig

TAGS **UNPREDICTABLE** DISPLAY Attempt to perform a memory transaction of more than 128 bytes.

ArchMsg.Warning.decode_unpred_other

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable.

ArchMsg.Warning.decode_unpreduseofpc

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable when %{FIELD} is PC. Fields:

FIELD enum

Field name.

ArchMsg.Warning.decode_unpreduseofr13

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable when %{FIELD} is R13. Fields:

FIELD enum

Field name.

ArchMsg.Warning.decode_unpreduseofr13inreglist

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with R13 in register-list.

ArchMsg.Warning.decode_useofd16orhigher

TAGS **UNPREDICTABLE** DISPLAY Attempt to use register D16 or higher where it is not allowed.

ArchMsg.Warning.decode_writebackandbaseinlist

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with write back when the base register is in the transfer list.

ArchMsg.Warning.decode_zeroregistersinlist

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable when no registers are to be transferred.

ArchMsg.Warning.recursive_branch

An instruction is performing a branch that targets the same instruction. If this was intended then a WFI might be more effective. DISPLAY recursive branch detected at PC=%{PC}.

PRIMARY KEY PC. Fields:

PC unsigned int

Address of instruction causing the branch.

ArchMsg.Warning.recursive_exception

An instruction has generated an exception that targets the same instruction. DISPLAY recursive %{TYPE} exception detected at PC=%{PC} (CPSR=%{CPSR}). PRIMARY KEY PC. Fields:

CPSR unsigned int

Processor state.

PC unsigned int

Address of instruction causing the exception.

TYPE enum

Type of exception.

ArchMsg.Warning.reentrant_vector_catch

TAGS **UNPREDICTABLE** DISPLAY Vector catch debug event on Prefetch or Data abort vector %{ADDR}. Fields:

ADDR unsigned int

vector address.

ArchMsg.Warning.reserved_it_state

TAGS **UNPREDICTABLE** DISPLAY Execution with reserved IT state %{ITSTATE}. Fields:

ITSTATE unsigned int

reserved IT state.

ArchMsg.Warning.tlb_contents_unknown

Translation has been enabled without invalidating the TLB. DISPLAY Translation was enabled in the %{PAS:(Secure|Non-secure|Root|Realm)} EL%{EL:d} translation regime (vmid=%{VMID:x}) without invalidating %{INVALIDITY:(|d-side |i-side |)}entries since power-on: %{INVALIDITY:(|D||I)}TLB could contain **UNKNOWN** content. Fields:

EL unsigned int

Exception level of translation regime.

INVALIDITY unsigned int

Which TLB may be invalid (if TLBs are separate).

NON_SECURE bool

Whether the translation regime is non-secure.

PAS unsigned int

Physical address space of translation regime.

VMID unsigned int

Current VMID, if applicable.

ArchMsg.Warning.unknown_DLR_DSPSR

Exiting debug state with unknown DLR or DSPSR.

ArchMsg.Warning.unknown_ELR_SPSR

Returning from debug exception with unknown ELR or SPSR.

ArchMsg.Warning.unpred_FAR_on_LRCPC3_non_atomic_mem_access

TAGS **UNPREDICTABLE** Unpredictable FAR/FSTATUS on LRCPC3 pair instructions resulting in two separate memory accesses
DISPLAY Unpredictable FAR on LRCPC3 non atomic memory access at %{ADDR1} with FSTATUS %{FSTATUS1} or %{ADDR2} with FSTATUS %{FSTATUS2}. Fields:

ADDR1 unsigned int

Possible FAR address 1.

ADDR2 unsigned int

Possible FAR address 2.

FSTATUS1 unsigned int

Possible fault status value 1.

FSTATUS2 unsigned int

Possible fault status value 2.

ArchMsg.Warning.unpred_async_tag_check_fail_on_sve_speculative_load_do_update_ffr

Unpredictable set to FALSE of the FFR element associated to SVE speculative load with an async tag check fail TAGS **UNPREDICTABLE**.

ArchMsg.Warning.unpredictable_a32_breakpoint

TAGS **UNPREDICTABLE** DISPLAY Unpredictable breakpoint on a A32 instruction at %{ADDR1} Offset %{OFFSET} BAS %{BAS} for a %{IS_ADDRESS_MISMATCH_BREAKPOINT:(Mismatch|Match)} Breakpoint number %{N:d} Breakpoint address %{ADDR2}. Fields:

ADDR1 unsigned int

Address.

ADDR2 unsigned int

Breakpoint address.

BAS unsigned int

BAS field.

IS_ADDRESS_MISMATCH_BREAKPOINT bool

Is a Mismatch breakpoint, not a Match breakpoint.

N unsigned int

Breakpoint number.

OFFSET unsigned int

Always has a value of zero.

ArchMsg.Warning.unpredictable_banked_register_access

TAGS **UNPREDICTABLE** DISPLAY Access to banked register (R=%{R:d} SYSm=%{SYSM:2x}) is unpredictable. Fields:

R unsigned int

R bit.

SYSM unsigned int

SYSm.

ArchMsg.Warning.unpredictable_ccfail_undef

TAGS **UNPREDICTABLE** DISPLAY An instruction %{OPCODE} at VA=%{VADDR} failed its condition codes check but would otherwise have caused a non-data-dependent trap or undefined exception. Behaviour is implementation-defined choice of exception or nop. Fields:

OPCODE unsigned int

opcode of the instruction.

VADDR unsigned int

Virtual address of the instruction being executed.

ArchMsg.Warning.unpredictable_ici_bits

TAGS **UNPREDICTABLE** DISPLAY ICI bits do not apply or are architecturally unpredictable for instruction.

ArchMsg.Warning.unpredictable_t32_breakpoint

TAGS **UNPREDICTABLE** DISPLAY Unpredictable breakpoint on a T32 instruction at %{ADDR1} Offset %{OFFSET} BAS %{BAS} for a %{IS_ADDRESS_MISMATCH_BREAKPOINT:(Mismatch|Match)} Breakpoint number %{N:d} Breakpoint address %{ADDR2}. Fields:

ADDR1 unsigned int

Address.

ADDR2 unsigned int

Breakpoint address.

BAS unsigned int

BAS field.

IS_ADDRESS_MISMATCH_BREAKPOINT bool

Is a Mismatch breakpoint, not a Match breakpoint.

N unsigned int

Breakpoint number.

OFFSET unsigned int

Indicates which halfword of a 32-bit Thumb instruction the breakpoint is set on. 0 for the first halfword or if the instruction is 16 bits wide, or 1 for the second halfword.

ArchMsg.Warning.unpredictable_t32_vector_catch

TAGS **UNPREDICTABLE** DISPLAY Unpredictable Vector catch exception on a T32 instruction at %{ADDR} Offset %{OFFSET}. Fields:

ADDR unsigned int

Address.

OFFSET unsigned int

Indicates which halfword of a 32-bit Thumb instruction the vector catch exception is set on. 0 for the first halfword or if the instruction is 16 bits wide, or 1 for the second halfword.

ArchMsg.Warning.unpredictable_watchpoint_far

TAGS **UNPREDICTABLE** DISPLAY The value for FAR/EDWAR for hitting this watchpoint may be anywhere in the range %{LOWER_BOUND} to %{UPPER_BOUND}. Fields:

LOWER_BOUND unsigned int

the lowest address accessed by the instruction that triggered the watchpoint.

UPPER_BOUND unsigned int

the highest watchpointed address accessed by that instruction.

ArchMsg.Warning.unpredictable_watchpoint_far_isscmp

DISPLAY The value for FAR/EDWAR for hitting this watchpoint may be anywhere in the range %{LOWER_BOUND} to %{UPPER_BOUND}. Fields:

LOWER_BOUND unsigned int

the lowest address accessed by the instruction that triggered the watchpoint.

UPPER_BOUND unsigned int

the highest watchpointed address accessed by that instruction.

ArchMsg.Warning.warning_AdvSIMDExpandImmUnexpectedZero

TAGS **UNPREDICTABLE** DISPLAY AdvSIMDExpandImm may treat this immediate value as **UNPREDICTABLE**.

ArchMsg.Warning.warning_ConditionalSMC

TAGS **UNPREDICTABLE** DISPLAY SMC instruction has **UNPREDICTABLE** effects when conditional and when combined with traps.

ArchMsg.Warning.warning_TLBID_out_of_range

TLBID out of range in TLBI according to the configuration of NIS, NVIS, NOS and NVOS. Fields:

NIS unsigned int

NIS.

NOS unsigned int

NOS.

NVIS unsigned int

NVIS.

NVOS unsigned int

NVOS.

TLBID unsigned int

The TLBID value used in the TLBI instruction.

ArchMsg.Warning.warning_TTL_or_TG_mismatches_current_translation_granule

TTL or TG mismatches current translation granule in TLBI VA/RVA/IPA/RIPA. Fields:

CURRENT_TG unsigned int

Current TG as indicated by TCR_ELx or VTCR_EL2.

TLBI_TG unsigned int

TG (or TTL[3:2]) used in TLBI instruction.

ArchMsg.Warning.warning_access_crosses_page_boundary

TAGS **UNPREDICTABLE** DISPLAY Access crosses page boundary.

ArchMsg.Warning.warning_access_crosses_page_boundary_has_fault

TAGS **UNPREDICTABLE** DISPLAY Access that crosses page boundary has fault(s).

ArchMsg.Warning.warning_access_crosses_page_boundary_has_fault_on_both_pages

TAGS **UNPREDICTABLE** DISPLAY Access that crosses page boundary has a fault on each page.
Fields:

ADDR1 unsigned int

FAR address on first page.

ADDR2 unsigned int

FAR address on second page.

FSTATUS1 unsigned int

Fault status on first page.

FSTATUS2 unsigned int

Fault status on second page.

ArchMsg.Warning.warning_access_crosses_page_boundary_spanning_different_memory

TAGS **UNPREDICTABLE** DISPLAY Access at address %{ADDR} crosses page from MEM_TYPE %{MEMTYPE_PAGE1} to %{MEMTYPE_PAGE2}. Fields:

ADDR unsigned int

address of access crossing page.

MEMTYPE_PAGE1 signed int

First Page's memory type (0-Normal, 1-Non-Normal).

MEMTYPE_PAGE2 signed int

Second Page's memory type (0-Normal, 1-Non-Normal).

ArchMsg.Warning.warning_access_wraps_around_memory

TAGS **UNPREDICTABLE** DISPLAY Access wraps around memory in %{WIDTH:d}bit mode.
Fields:

WIDTH unsigned int

Address width.

ArchMsg.Warning.warning_bcr_linking_status

TAGS **UNPREDICTABLE** DISPLAY DBGBCR.{BT2,BT} combination gives unpredictable behaviour.

ArchMsg.Warning.warning_bcr_mask_reserved

TAGS **UNPREDICTABLE** DISPLAY DBGBCR_ELn.MASK is set to a reserved value.

ArchMsg.Warning.warning_bvr_masked_bit_not_zero

TAGS **UNPREDICTABLE** DISPLAY DBGBVRn_EL1.VA contains a bit masked by DBGBCRn_EL1.MASK that is not zero.

ArchMsg.Warning.warning_ccsidr_unimplemented_level

TAGS **UNPREDICTABLE** DISPLAY CCSIDR read while CSSELR %{CSSELR:x} points to an unimplemented cache level. Fields:

CSSELR unsigned int

current effective value of CSSELR.

ArchMsg.Warning.warning_change_to_ns_when_tge_set

TAGS **UNPREDICTABLE** DISPLAY Attempting to change NS to 1 when HCR.TGE = 1.

ArchMsg.Warning.warning_contiguous_bit_check_abort

TAGS **UNPREDICTABLE** DISPLAY An abort occurred whilst attempting to check TLB entry is contiguous at %{ENTRY_ADDR}. Fields:

ENTRY_ADDR unsigned int

address of conflicting TLB entry.

ArchMsg.Warning.warning_contiguous_bit_error

TAGS **UNPREDICTABLE** DISPLAY TLB entry at %{ENTRY_ADDR} was not contiguous with entry at %{CONTIG_ADDR} Contents are %{ENTRY_DATA} but expected %{CONTIG_DATA}. Fields:

CONTIG_ADDR unsigned int

address of the TLB first read, with which this entry is expected to be contiguous.

CONTIG_DATA unsigned int

expected contents based on the entry at CONTIG_ADDR.

ENTRY_ADDR unsigned int

address of conflicting TLB entry.

ENTRY_DATA unsigned int

contents of conflicting TLB entry.

ArchMsg.Warning.warning_contiguous_bit_error_D128

TAGS **UNPREDICTABLE** DISPLAY TLB entry at %{ENTRY_ADDR} was not contiguous with entry at %{CONTIG_ADDR} Contents are %{ENTRY_DATA_HI} %{ENTRY_DATA_LO} but expected %{CONTIG_DATA}. Fields:

CONTIG_ADDR unsigned int

address of the TLB first read, with which this entry is expected to be contiguous.

CONTIG_DATA unsigned int

expected contents based on the entry at CONTIG_ADDR.

ENTRY_ADDR unsigned int

address of conflicting TLB entry.

ENTRY_DATA_HI unsigned int

Higher bit contents of conflicting TLB entry.

ENTRY_DATA_LO unsigned int

Lower bit contents of conflicting TLB entry.

ArchMsg.Warning.warning_cp10_cp11_mismatch

TAGS **UNPREDICTABLE** DISPLAY Unpredictable configuration of CP10 and CP11 controls at EL
%{EL:d}. Fields:

EL unsigned int

EL owning the control.

ArchMsg.Warning.warning_cp10_cp11_reserved_value

TAGS **UNPREDICTABLE** DISPLAY Unpredictable configuration of CPACR CP10 (%{CP10}) or
CP11 (%{CP11}). Fields:

CP10 unsigned int

CP10 access.

CP11 unsigned int

CP11 access.

ArchMsg.Warning.warning_csselr_level_out_of_range

TAGS **UNPREDICTABLE** DISPLAY CSSELR.Level out of range, written value %{LEVEL} with only
%{IMPLEMENTED:d} levels implemented. Fields:

IMPLEMENTED unsigned int

number of cache levels implemented.

LEVEL unsigned int

written value of CSSELR.Level.

TYPE enum

The type of cache selected by CSSELR.InD and TnD.

ArchMsg.Warning.warning_debug_flow_control_bits_not_obeyed

TAGS **UNPREDICTABLE** DISPLAY Flow control bits for %{REGISTER} were not obeyed, causing
%{ERROR}. Fields:

ERROR enum

type of error resulting.

REGISTER enum

register under consideration.

ArchMsg.Warning.warning_debug_register_access_during_reset

TAGS **UNPREDICTABLE** DISPLAY Unpredictable %{IS_WRITE:(read|write)} access to debug register offset %{OFFSET} during reset. Fields:

IS_WRITE unsigned int

Write Not Read.

OFFSET unsigned int

Register Offset.

ArchMsg.Warning.warning_decode_cps_inconsistent_fields

TAGS **UNPREDICTABLE** DISPLAY The mode setting fields in the CPS instruction are inconsistent.

ArchMsg.Warning.warning_decode_invalid_state

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable in current exception-level/security state.

ArchMsg.Warning.warning_default_cacheable_mmu_on

TAGS **UNPREDICTABLE** DISPLAY Default cacheable is enabled (HCR.DC = 1) while MMU is enabled.

ArchMsg.Warning.warning_default_cacheable_vmmu_off

TAGS **UNPREDICTABLE** DISPLAY Default cacheable is enabled (HCR.DC = 1) while HCR.VM = 0.

ArchMsg.Warning.warning_deprecated_wvr_bit_2_set

TAGS **UNPREDICTABLE** DISPLAY DBGWVR%{N:d}_EL1=%{WVR:h} has bit 2 set, deprecated in ARMv8. Fields:

N unsigned int

index.

WVR unsigned int

DBGWVR.

ArchMsg.Warning.warning_eret_while_software_step_active_pending

Missing ISB between setting MDSCR_EL1.SS and ERET TAGS **UNPREDICTABLE**.

ArchMsg.Warning.warning_exclusive_to_non_normal

TAGS **UNPREDICTABLE** DISPLAY Exclusive Access to Non-Normal Memory %{ADDR}. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_exclusive_to_non_writeback

TAGS **UNPREDICTABLE** DISPLAY Exclusive Access to Non-Writeback-Cacheable Memory %{ADDR}. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_execute_from_device_memory

TAGS **UNPREDICTABLE** DISPLAY An attempt was made to execute from Device memory.

ArchMsg.Warning.warning_illegal_cpsr_mode

TAGS **UNPREDICTABLE** DISPLAY Writing an illegal or unimplement mode (%{NEW_MODE:h}) to CPSR was unpredictable. Fields:

NEW_MODE unsigned int

New value of CPSR.M.

ArchMsg.Warning.warning_illegal_srs_mode

TAGS **UNPREDICTABLE** DISPLAY Illegal or **UNPREDICTABLE** mode (%{MODE:h}) used for SRS instruction. Fields:

MODE unsigned int

mode for Banked SP.

ArchMsg.Warning.warning_implementation_defined_read_debug_register_in_SCS

Read Access to Debug registers from the processor is **IMPLEMENTATION DEFINED** TAGS **IMP_DEF**. Fields:

PPB_OFFSET unsigned int

debug register offset.

ArchMsg.Warning.warning_implementation_defined_sequential_security_transitions_supported

The behavior of sequential instruction fetches that cross from non-secure to secure memory and contain an SG instruction is **CONSTRAINED UNPREDICTABLE** TAGS **UNPREDICTABLE**.

ArchMsg.Warning.warning_implementation_defined_stack_limit_check_supported

It is **IMPLEMENTATION DEFINED** whether stack pointer limit checking is performed for this instruction TAGS **IMP_DEF**. Fields:

ADDRESS unsigned int

Address of instruction.

OPCODE unsigned int

instruction opcode.

ArchMsg.Warning.warning_implementation_defined_taggednonsharedmemorytype

TAGS **IMP_DEF** DISPLAY Memory attribute has Tagged, non shared memory type; it is **IMPLEMENTATION DEFINED** whether the memory location is treated as Tagged or non-Tagged.

ArchMsg.Warning.warning_implementation_defined_write_debug_register_in_SCS

Write Access to Debug registers from the processor is **IMPLEMENTATION DEFINED** TAGS **IMP_DEF**. Fields:

DATA unsigned int

data attempted to be written.

PPB_OFFSET unsigned int

debug register offset.

ArchMsg.Warning.warning_invalid_tcr_granule

TAGS UNPREDICTABLE DISPLAY TCR.TG{%TG_ID:(0|1)} bits have been set to a granule size {%REQUEST:d}K not implemented - using {%SUBSTITUTE:d}K. Fields:

REQUEST signed int

page size requested (or 0 for reserved).

SUBSTITUTE signed int

best guess available page size.

TG_ID bool

bits TG0 or TG1.

ArchMsg.Warning.warning_load_multiple_user_registers_from_user_mode

TAGS UNPREDICTABLE DISPLAY An LDM(user registers) instruction executed from user mode.

ArchMsg.Warning.warning_mte_in_mem_attr_not_supported

TAGS UNPREDICTABLE DISPLAY Memory attribute {%ATTR} has Tagged memory type but Armv8.5-MTE is not implemented. Fields:

ATTR unsigned int

Memory attribute.

ArchMsg.Warning.warning_reserved_breakpoint_state_match

TAGS UNPREDICTABLE DISPLAY {%IS_BREAKPOINT:(Watchpoint|Breakpoint)} programmed with a reserved combination of HMC, SSC and {%IS_BREAKPOINT:(PAC|PMC)}. Fields:

IS_BREAKPOINT unsigned int

Is a breakpoint, not a watchpoint.

ArchMsg.Warning.warning_shareability

TAGS UNPREDICTABLE DISPLAY Unpredictable: combination of the 1st and 2nd stages of translation is Normal Inner Non-Cacheable, Outer Non-Cacheable.

ArchMsg.Warning.warning_software_step_set_while_enabled

TAGS UNPREDICTABLE DISPLAY MDSCR_EL1.SS set to 1 while software step debug exceptions are enabled.

ArchMsg.Warning.warning_thumb_instruction_wraps_around_memory

TAGS UNPREDICTABLE DISPLAY Thumb instruction wraps around memory.

ArchMsg.Warning.warning_ttbr_sbz_bits_are_not_zero

TAGS UNPREDICTABLE DISPLAY {%TTBR}[{%MSB:d}:0] should be zero, but are not. Fields:

MSB unsigned int

x-1.

TTBR enum

which TTBR register.

ArchMsg.Warning.warning_unaligned_address_dbgdtrrx_write

TAGS **UNPREDICTABLE** DISPLAY Unaligned address %{ADDR} in X0/R0 when DBGDTRRX written in memory access mode. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unaligned_address_dbgdtrtx_read

TAGS **UNPREDICTABLE** DISPLAY Unaligned address %{ADDR} in X0/R0 when DBGDTRTX read in memory access mode. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unaligned_to_device

TAGS **UNPREDICTABLE** DISPLAY Unaligned Access to Device Memory %{ADDR}. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unaligned_to_strongly_ordered

TAGS **UNPREDICTABLE** DISPLAY Unaligned Access to Strongly Ordered Memory %{ADDR}. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unknown_sau_rnr

TAGS **UNPREDICTABLE** DISPLAY SAU_RNR was set to an unsupported value.

ArchMsg.Warning.warning_unpredictable_AIRCR_PRIS_and_BFHFNMINs

TAGS **UNPREDICTABLE** DISPLAY The effect of setting both AIRCR.BFHFNMINs and AIRCR.PRIS to 1 is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_AIRCR_VECTCLRACTIVE_when_not_in_debug

TAGS **UNPREDICTABLE** DISPLAY The effect of writing a 1 to AIRCR.VECTCLRACTIVE if the processor is not halted in Debug state is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_AIRCR_VECTRESET_and_SYSRESETREQ

TAGS **UNPREDICTABLE** DISPLAY When the processor is halted in Debug state, if a write to the register writes a 1 to both VECTRESET and SYSRESETREQ, the behavior is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_AIRCR_VECTRESET_when_not_in_debug

TAGS **UNPREDICTABLE** DISPLAY The effect of writing a 1 to AIRCR.VECTRESET if the processor is not halted in Debug state is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_AIRCR_incorrect_VKEY

TAGS **UNPREDICTABLE** DISPLAY The value 0x05FA must be written to AIRCR.VKEY, otherwise the register write is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_EXC_RETURN_Reserved_Bit

EXC_RETURN[23:7] are reserved with the special condition that all bits should be written as one. Values other than all 1s are **UNPREDICTABLE**. TAGS **UNPREDICTABLE**. Fields:

EXC_RETURN unsigned int

Special PC value.

SBO_MASK unsigned int

Which bits Should Be One to not UNPRED.

ArchMsg.Warning.warning_unpredictable_PMEVTYPEPn_ELO_VS_reserved_value

TAGS **UNPREDICTABLE** DISPLAY Unpredictable reserved value of PMEVTYPEN<n>_ELO.VS.

ArchMsg.Warning.warning_unpredictable_change_to_DEMCR_MON_STEP_at_insufficient_priority

TAGS **UNPREDICTABLE** DISPLAY The effect of changing DEMCR.MON_STEP at an execution priority that is lower than the priority of the DebugMonitor exception is **CONSTRAINED UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_change_to_priority_of_active_exception

TAGS **UNPREDICTABLE** DISPLAY Changing the priority of an active exception is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_clear_lspact

TAGS **UNPREDICTABLE** DISPLAY **UNPREDICTABLE** state on context switch. Fields:

DRVEXC_TAKEN unsigned int

Derived exception taken, not pended.

REACHED_LAST_STORE unsigned int

Only the last faulted.

ArchMsg.Warning.warning_unpredictable_dbg_exit_unaligned_dlr

TAGS **UNPREDICTABLE** DISPLAY Debug exit to AArch32 with DLR[0] set to 1. PC[0] can be set to 0 or to DLR[0].

ArchMsg.Warning.warning_unpredictable_exception_catch

TAGS **UNPREDICTABLE** DISPLAY Generation of an exception catch event is unpredictable.

ArchMsg.Warning.warning_unpredictable_exception_return_inconsistent_state

TAGS **UNPREDICTABLE** DISPLAY state on exception return is unpredictable.

ArchMsg.Warning.warning_unpredictable_exception_return_instruction

TAGS **UNPREDICTABLE** DISPLAY exception return instruction is unpredictable.

ArchMsg.Warning.warning_unpredictable_in_debug_state

TAGS **UNPREDICTABLE** DISPLAY Instruction is **UNPREDICTABLE** when executed in debug state.

ArchMsg.Warning.warning_unpredictable_mair_encoding

TAGS **UNPREDICTABLE** DISPLAY Memory attribute %{ATTR} has invalid MAIR encoding; this may lead to unpredictable results. Fields:

ATTR unsigned int

Memory attribute.

ArchMsg.Warning.warning_unpredictable_pmu_counter_access

TAGS **UNPREDICTABLE** DISPLAY Access to PMU counters from non-secure EL0 or EL1 is **UNPREDICTABLE** with MDCR_EL2.HPMN set to 0.

ArchMsg.Warning.warning_unpredictable_prioritization_breakpoint_match_vector_catch

TAGS **UNPREDICTABLE** DISPLAY Two events with the same priority occurred at the same instruction: Address Matching Vector Catch debug event, and Breakpoint debug event. It is constrained unpredictable which is taken.

ArchMsg.Warning.warning_unpredictable_stack_selection

TAGS **UNPREDICTABLE** DISPLAY **UNPREDICTABLE** state on context switch.

ArchMsg.Warning.warning_unpredictable_tsize_out_of_range

TAGS **UNPREDICTABLE** DISPLAY %{TCR}.T%{TTBR:d}SZ (%{TSIZE:d} bits) is out of range. Must be between 25 and %{TSIZE_MAX:d}. Fields:

TCR enum

Which TCR.

TSIZE signed int

Value of TSize.

TSIZE_MAX signed int

The maximum allowed value.

TTBR unsigned int

Which TTBR region.

ArchMsg.Warning.warning_unpredictable_unaligned_pc_as_base_register

TAGS **UNPREDICTABLE** DISPLAY The PC value must be word-aligned, otherwise the behavior of the instruction is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_unaligned_pop_stack

TAGS **UNPREDICTABLE** DISPLAY **CONSTRAINED UNPREDICTABLE** unaligned stack frame %{FRAME_ADDR:x} in exception-return. Fields:

FRAME_ADDR unsigned int

unaligned stack frame address.

ArchMsg.Warning.warning_unpredictable_unaligned_to_non_writeback

TAGS **UNPREDICTABLE** DISPLAY Unaligned Access to Non-Writeback-Cacheable Memory.

ArchMsg.Warning.warning_unpredictable_vmsa_memattrib

TAGS **UNPREDICTABLE** DISPLAY Unpredictable vmsa memory attribute with texcb value programmed to %{TEXCB} when tex remap is %{TEX_REMAP}. Fields:

TEXCB unsigned int

texcb value.

TEX_REMAP bool

use tex remap.

ArchMsg.Warning.warning_unpredictable_vtcr_t0sz_sl0

TAGS **UNPREDICTABLE** DISPLAY The combination of VTCR.TOSZ=%{TOSZ:d} and VTCR.SLO=%{SLO:d} is unpredictable. Fields:

SLO signed int

VTCR.SLO, starting level for stage 2 translation table walks.

TOSZ signed int

VTCR.TOSZ, required input address range.

ArchMsg.Warning.warning_unpredictable_write_DHCSR

TAGS **UNPREDICTABLE** DISPLAY **UNPREDICTABLE** set of bit-changes in DHCSR. Fields:

NEW unsigned int

data attempted to be written.

OLD unsigned int

previous value.

ArchMsg.Warning.warning_unpredictable_write_SHCSR

TAGS **UNPREDICTABLE** DISPLAY **UNPREDICTABLE** setting a pending bit to 1 for an exception with priority >= execution priority.

ArchMsg.Warning.warning_unsupported_access_to_memory_mapped_register

TAGS **UNPREDICTABLE** DISPLAY Access to memory mapped register is unsupported.

ArchMsg.Warning.warning_user_jmcr_access

TAGS **UNPREDICTABLE** DISPLAY User %{WRITE:(write|read)} access to JMCR is unpredictable. Fields:

WRITE unsigned int

access is write.

ArchMsg.Warning.warning_wcr_mask_and_bas

TAGS **UNPREDICTABLE** DISPLAY DBGWCR_EL1.MASK is non-zero and BAS is not 0xff.

ArchMsg.Warning.warning_wcr_mask_reserved

TAGS **UNPREDICTABLE** DISPLAY DBGWCR_ELn.MASK is set to a reserved value.

ArchMsg.Warning.warning_wcr_non_configuous_bas

TAGS **UNPREDICTABLE** DISPLAY DBGWCR_EL1.BAS has non-contiguous set of ones.

ArchMsg.Warning.warning_wvr_masked_bit_not_zero

TAGS **UNPREDICTABLE** DISPLAY DBGWVRn_EL1.VA contains a bit masked by DBGWCRn_EL1.MASK that is not zero.

ArchMsg.Why.why_branch_target_exception

A branch target exception has occurred. DISPLAY branch target exception: target pc=%{PC}, target opcode=%{OPCODE}. PRIMARY KEY PC. Fields:

BTYPE unsigned int

pstate.BTYPE at time branch taken.

OPCODE unsigned int

opcode at branch target.

PC unsigned int

Virtual Address of branch target.

ArchMsg.Why.why_illegal_state

An illegal mode change has occurred. DISPLAY illegal mode change: %{MESSAGE}. Fields:

MESSAGE string

The message.

ArchMsg.warning_invalid_interval_counter_reload_value

Invalid interval counter reload value in PMSIRR_EL1.INTERVAL: software should set this to a value greater than or equal to the minimum indicated by PMSIDR_EL1.Interval.

ArchMsg.warning_invalid_spe_buffer_write_pointer

Invalid value programmed in PMBPTR_EL1.PTR or PMBLIMITR_EL1.LIMIT: Current write pointer must be at least one sample record below the write limit pointer and PMBPTR_EL1.PTR[63:56] must be equal to PMBLIMITR_EL1.LIMIT[63:56].

BRANCH_MISPREDICT

Simulating branch mispredict. Fields:

PC unsigned int

Origin address (or 0 if unavailable).

BRA_DIR

Direct branches, to immediate address. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISSET enum

The instruction set of the branch instruction.

IS_COND bool

Indicates if this is a conditional branch.

IS_HINTED bool

Whether the conditional instruction is hinted.

IS_ISB bool

Whether the instruction is ISB.

IS_LINK bool

Indicates if this is a branch with link.

PC unsigned int

The address of the branch instruction.

TARGET_ISET enum

The instruction set after the branch.

TARGET_PC unsigned int

The address the instruction branches to.

BRA_INDIR

Indirect branches, perhaps to a register. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISET enum

The instruction set of the branch instruction.

IS_COND bool

Indicates if this is a conditional branch.

IS_HINTED bool

Whether the conditional instruction is hinted.

IS_ISB bool

Whether the instruction is ISB.

IS_LINK bool

Indicates if this is a branch with link.

PC unsigned int

The address of the branch instruction.

TARGET_ISET enum

The instruction set after the branch.

TARGET_PC unsigned int

The address the instruction branches to.

BRA_INDIR_CCFAIL

Indirect branches fail, perhaps to a register.

BRA_RET_CCFAIL

Branch return fail, perhaps to a register.

CACHE_MAINTENANCE_OP

Cache Maintenance Operation. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

DATA unsigned int

Specified MVA or Set/way.

FUNCTION enum

Clean / Invalidate.

SCOPE enum

Affected region.

SIDE enum

Inst / Data.

CCFAIL

Conditional instruction condition check fail. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CCFAIL_UNC

Unconditional instruction condition check fail. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CCPASS

Conditional instruction condition check pass. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CCPASS_UNC

Unconditional instruction condition check pass. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CODE_CACHE_MAINT

Code cache maintenance. Fields:

END_INCL unsigned int

Inclusive end address for VA or PA requests.

NS bool

Non-secure world for PA.

START unsigned int

Start address for VA or PA requests.

TYPE enum

Request type.

COMPILE_BLOCK_END

Last instruction of basic block translated. Fields:

VADDR unsigned int

Address of next instruction after this basic block.

COMPILE_BLOCK_START

First instruction of basic block translated. Fields:

VADDR unsigned int

Address of first instruction in basic block.

COMPILE_INST

ARM instruction compiled. Fields:

DISASS string

Disassembly of the instruction.

ISSET enum

The instruction set of this instruction.

ITSTATE unsigned int

The ITSTATE current for the instruction.

OPCODE unsigned int

The opcode of the instruction.

PC unsigned int

The address of the instruction.

SIZE unsigned int

The size of the instruction in bytes.

CONTEXTIDR

Context ID Register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

NS bool

Secure or nonsecure banked register is accessed.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The new value written.

CONTEXT_SYNC

Called for every context synchronization event. Fields:

ARCH_CSE bool

Architectural Context Synchronization Event.

PC unsigned int

The address of the synchronization instruction.

SYSREG_SYNC bool

System register synchronization event.

CORE_ENDIAN

Core BE8 Big-Endian state changed. Fields:

BE8 bool

Core BE8 Big-Endian state.

CORE_INFO

Static processor attributes. Only triggered by a call to DumpState(). Fields:

ARCH_PROFILE enum

The architecture profile of the core.

CLUSTER_ID unsigned int

The cluster ID of this processor.

CORE_NUM unsigned int

The number of this core in an MP processor.

DETAILED_CORE_FEATURES enum

Detailed core features as a bitfield. Enum values are bitmasks to extract the info from the field.

FPU_VERSION enum

The VFP version implemented by the core.

MEM_ARCH enum

The memory architecture of the core.

MIDR_EL1 unsigned int

MIDR_EL1 value.

NUM_CORES unsigned int

The number of cores in this MP processor.

QUANTUM_SIZE unsigned int

The default quantum size of the core (in ticks).

SECURITY_FEATURES bool

Does the core have security features?.

CORE_LOADS

Processor load accesses. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ACQREL enum

Is this an acquire/release.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DATA unsigned int

The data read or written.

ELEMENT_SIZE unsigned int

Width of each element.

ENDIANNESS enum

Byte ordering of this access.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

LOCK enum

Normal, exclusive or locked access.

MECID unsigned int

Memory Encryption Context ID, if applicable.

MECID2 unsigned int

Memory Encryption Context ID of the second page, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PAGE_SIZE unsigned int

Page size as $\log_2(\text{size in bytes})$.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK, 2=Exclusive Failed, 3=Unsupported.

SIZE unsigned int

Width of the access in bytes. Only required if DATA is not traced.

TRANS bool

Is this a translated access.

VADDR unsigned int

The virtual address of the access.

CORE_REGS

Changes of the core registers R0 to R14. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number, 0 to 14.

MODE enum

Bank of the register accessed.

OLD_VALUE unsigned int

The old value overwritten.

PHYS_ID enum

The physical register accessed.

VALUE unsigned int

The new value written to the register.

CORE_REGS64

Changes of the core registers X0..X30, SP_ELn. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ID enum

The register number, 0-30 for X0-X30, >=32 for SP_ELn.

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

CORE_REGS64_READ

General purpose AArch64 register has been read X0..X30, SP_ELn. Fields:

ID enum

The register number, 0-30 for X0-X30, >=32 for SP_ELn.

VALUE unsigned int

The value read from the register.

CORE_STALLED_TRANSITION

Detect when a core is prevented from progressing PChannel states. Fields:

CURRENT_STATE enum

Current power mode of core.

REASON string

The reason a transition was stalled, and what must be done to unblock it.

TARGET_STATE enum

Desired power mode of core after transition.

CORE_STORES

Processor store accesses. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ACQREL enum

Is this an acquire/release.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DATA unsigned int

The data read or written.

ELEMENT_SIZE unsigned int

Width of each element.

ENDIANNESS enum

Byte ordering of this access.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

LOCK enum

Normal, exclusive or locked access.

MECID unsigned int

Memory Encryption Context ID, if applicable.

MECID2 unsigned int

Memory Encryption Context ID of the second page, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PAGE_SIZE unsigned int

Page size as $\log_2(\text{size in bytes})$.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK, 2=Exclusive Failed, 3=Unsupported.

SIZE unsigned int

Width of the access in bytes. Only required if DATA is not traced.

TRANS bool

Is this a translated access.

VADDR unsigned int

The virtual address of the access.

CP14_READ

System Coprocessor register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP14 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc1 unsigned int

opcode 1.

opc2 unsigned int
opcode 2.

CP14_WRITE

System Coprocessor register write. Fields:

CORE_NUM unsigned int
Core number in a multi processor.

CRm unsigned int
CRm.

CRn unsigned int
CRn.

NS bool
Secure or nonsecure banked register is accessed.

REG_NAME string
Name of the CP14 register accessed.

SECURITY_STATE enum
The current security state.

UNDEF bool
The register accessed is undefined.

UPDATED_VALUE unsigned int
Updated value of the register now it has been written.

VALUE unsigned int
The new value written.

opc1 unsigned int
opcode 1.

opc2 unsigned int
opcode 2.

CP15_READ

System Coprocessor register read. Fields:

CORE_NUM unsigned int
Core number in a multi processor.

CRm unsigned int
CRm.

CRn unsigned int
CRn.

NS bool
Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP15 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc1 unsigned int

opcode 1.

opc2 unsigned int

opcode 2.

CP15_READ64

System Coprocessor register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP15 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc unsigned int

opcode 1.

CP15_WRITE

System Control Coprocessor register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP15 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc1 unsigned int

opcode 1.

opc2 unsigned int

opcode 2.

CP15_WRITE64

System Coprocessor register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP15 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc unsigned int

opcode 1.

CPSR

CPSR change. (In AArch64 PSTATE represented in SPSR format). Fields:

CORE_NUM unsigned int

Core number in a multi processor.

OLD_VALUE unsigned int

The old CPSR value.

UNKNOWN unsigned int

Bits within the register that have unknown value.

VALUE unsigned int

The new CPSR value.

CRYPTO_SPEC

Every crypto instruction speculatively executed.

CorePowerStateChange

Core power-up state has changed. DISPLAY Core powered %{PU:(up|down)}. Fields:

PU bool

Whether the core powered up.

DATA_CACHE_SET_ALLOC_TAGS

Set allocation tags from DC GVA instruction. Fields:

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

NSDESC unsigned int

The security state of the access.

PADDR unsigned int

The physical address of the access.

PAS enum

The physical address space of the page.

RESPONSE enum

0=Aborted, 1=OK.

SIZE unsigned int

The size of memory in bytes.

TAG unsigned int

The value of the tag being set.

TAGGING_TYPE enum

Type of Memory tagging (Physical or Virtual).

TAG_VADDR unsigned int

The virtual address of the Virtual Tag if Virtual tagging is used; otherwise 0.

VADDR unsigned int

Virtual address used for setting allocation tags.

DATA_CACHE_ZERO

Zero and invalidate cache line from DC ZVA instruction. Fields:

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

NSDESC unsigned int

The security state of the access.

PADDR unsigned int

The physical address of the access.

PADDR_TAG unsigned int

If tag checked, the physical address tag for the access.

PAS enum

The physical address space of the page.

RESPONSE enum

0=Aborted, 1=OK.

SIZE unsigned int

The size of zero'd memory in bytes.

TAG_CHECKED bool

Is this access tag checked.

VADDR unsigned int

Virtual address of the zero'd cache line.

DEBUG_EVENT

Hardware debug support event. Fields:

EVENT enum

Description of event.

VALUE unsigned int

data value.

DEBUG_TARGET_ISET

Hardware debug target ISA. Fields:

TARGET_ISET enum

The target instruction set of the debug state.

DMI_ALLOCATE

DMI allocated. Fields:

ACCESS_DIRECTION enum

READ / WRITE.

PADDR unsigned int

The physical address of the DMI.

PTR unsigned int

DMI range pointer.

SIDE enum

DSIDE / ISIDE.

SIZE unsigned int

The size of DMI.

VADDR unsigned int

The virtual address of the DMI..

DMI_HIT

DMI hit, access used DMI (only fire for small DMI regions i.e. less than a full page). Fields:

ACCESS_DIRECTION enum

READ / WRITE.

PADDR unsigned int

The physical address of the DMI.

SIDE enum

DSIDE / ISIDE.

SIZE unsigned int

The size of the DMI.

VADDR unsigned int

The virtual address of the DMI..

DMI_REVOKE

DMI revoked. Fields:

ACCESS_DIRECTION enum

READ / WRITE.

PADDR unsigned int

The physical address of the DMI.

SIDE enum

DSIDE / ISIDE.

SIZE unsigned int

The size of DMI.

VADDR unsigned int

The virtual address of the DMI..

END_COMPILE

Compilation end. Fields:

END_OF_PAGE_COUNT unsigned int

Number of basic blocks exited due to page boundary since START_COMPILE.

FETCHFAIL_COUNT unsigned int

Number of basic blocks exited due to ifetch failure START_COMPILE.

INST_COUNT unsigned int

Number of instructions compiled since START_COMPILE.

NONSEQ_COUNT unsigned int

Number of basic blocks exited due to non-sequential instructions since START_COMPILE.

PAGE_STRADDLE_COUNT unsigned int

Number of basic blocks exited due to unaligned instructions crossing page since START_COMPILE.

ESB

Error Synchronization Barrier hit. Fields:

ASYNC bool

ESB is async.

IMPLICIT bool

ESB is implicit.

EXCEPTION

Exception taken. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ESR unsigned int

The value assigned to ESR when exception is taken.

ESR_EC unsigned int

The Value assigned to ESR.EC when exception is taken.

IS_PHYSICAL bool

Physical or Virtual exception.

IS_TRAP bool

Exception coming from a trap.

LR unsigned int

The value assigned to the link register.

PC unsigned int

The location where the exception occurred.

PREFERRED_RETURN unsigned int

The preferred return address for the exception.

TAKEN_LOCALLY bool

Whether exception is taken locally(same exception level).

TARGET_ISET enum

The instruction set of the exception handler code.

TARGET_PC unsigned int

The address the exception branches to.

VECTOR enum

The exception vector.

VECTOR_OFFSET unsigned int

The offset of an exception vector from the base address.

EXCEPTION_END

Every exception completed.

EXCEPTION_INFO

Trace human-readable reason of the current exception for data/instruction aborts, floating point exceptions and unknown exceptions. Fields:

DESCRIPTION string

Reason for current exception.

ESR_EC unsigned int

The value of ESR.EC when exception is taken.

EXCEPTION_RAISE

Every exception raised.

EXCEPTION_RETURN

Branches on leaving exception. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISET enum

The instruction set of the branch instruction.

IS_COND bool

Indicates if this is a conditional branch.

IS_HINTED bool

Whether the conditional instruction is hinted.

IS_ISB bool

Whether the instruction is ISB.

IS_LINK bool

Indicates if this is a branch with link.

PC unsigned int

The address of the branch instruction.

TARGET_ISET enum

The instruction set after the branch.

TARGET_PC unsigned int

The address the instruction branches to.

EXCEPTION_RETURN_PREBRANCH

Exception return event that must occur before PC and processor state have updated.

EXCEPTION_START

Every exception started.

FIQ_TAKEN

FIQ taken exception.

FP_EXCEPTIONS_SIGNALLED

Trace which FP exceptions were signalled as a part of the FP operations performed by an instruction. Fields:

FP_EXCEPTIONS_SIGNALLED_MASK unsigned int

Indicates which FP exceptions were signalled by an instruction. This uses the same encoding as the cumulative floating-point exception bitfields in FPSR.

FP_STATE

Floating point state. Fields:

FPSR unsigned int

FPSR register value.

MASK unsigned int

The mask indicating updated cumulative bits.

FREQ_CHANGED

Core frequency changed. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The instruction count of this core.

NEW_FREQ signed int

The new frequency of this core (expressed in Hz).

OLD_FREQ signed int

The old frequency of this core (expressed in Hz).

HLT

HLT instruction occurred. Fields:

IMM16 unsigned int

The 16-bit immediate value encoded in HLT instruction.

INST

Every instruction executed. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DEBUG_STATE bool

The instruction is executed in debug state.

DISASS string

Disassembly of instruction.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISSET enum

The current instruction set.

ITSTATE unsigned int

The current ITSTATE.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

MODE enum

The mode the core is in.

NS unsigned int

The core's non-secure bit.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

OPCODE unsigned int

The opcode of the instruction.

PADDR unsigned int

The physical address of the instruction.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the instruction.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the page.

PC unsigned int

The address of the instruction.

SECURITY_STATE enum

The core's security state.

SIZE unsigned int

The size of the instruction in bytes.

INST_END

Every instruction completed.

INST_START

Every instruction started. Fields:

DEBUG_STATE bool

The instruction is executed in debug state.

ISSET enum

The current instruction set.

MODE enum

The mode the core is in.

NS enum

The current Secure State.

PC unsigned int

The address of the conditional instruction.

SECURITY_STATE enum

The current Security State.

INST_STRADDLE

Instruction straddles boundary. Fields:

DEBUG_STATE bool

The instruction is executed in debug state.

ISSET enum

The current instruction set.

MODE enum

The mode the core is in.

NS enum

The current Secure State.

PC unsigned int

The address of the conditional instruction.

SECURITY_STATE enum

The current Security State.

IRQ_TAKEN

IRQ taken exception.

ISB

ISB instruction occurred. Fields:

PC unsigned int

The address of the ISB instruction.

LOAD_MULTIPLE_REGS_LIST

List of destination registers (not source) of a multiple registers load instruction. Fields:

REG string

Register to add into the list.

LOCAL_MONITOR

Local monitor activity. Fields:

PADDR unsigned int

Local Monitor Address.

State enum

State of the monitor (Open/Exclusive).

MAPPING_CREATE

A memory mapping has been created. This is an implementation detail of the model and so may change without notice. Fields:

ASID unsigned int

The ASID used during translation, or 0 if no ASID was used.

IS_VALID bool

Whether the mapping is valid or not.

PADDR unsigned int

Base physical address of region, or 0 if not known.

SIDE enum

I-side or D-side.

SIZE unsigned int

Size of the page, or 0 if not known.

VADDR unsigned int

Base virtual address of region.

VMID unsigned int

The VMID used during translation, or 0 if no VMID was used.

MAPPING_DESTROY

A memory mapping has been destroyed. This is an implementation detail of the model and so may change without notice. Fields:

ASID unsigned int

The ASID used during translation.

CAUSE enum

The cause of invalidation.

PADDR unsigned int

Base physical address of page.

SIDE enum

I-side or D-side.

SIZE unsigned int

Size of the page.

VADDR unsigned int

Base virtual address of page.

VMID unsigned int

The VMID used during translation.

MEMMAP_DEBUG_READ

Memory mapped reads to the debug registers. Fields:

ADDR unsigned int

address.

CORE_NUM unsigned int

Core number in a multi processor.

EXT bool

Whether access is from an external device (such as the DAP).

NS enum

Secure state of the access.

REG_NAME string

Name of the debug register accessed.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

MEMMAP_DEBUG_WRITE

Memory mapped writes to the debug registers. Fields:

ADDR unsigned int

address.

CORE_NUM unsigned int

Core number in a multi processor.

EXT bool

Whether access is from an external device (such as the DAP).

NS enum

Secure state of the access.

REG_NAME string

Name of the debug register accessed.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The new value written.

MMU_D128_TTB_READ

This event is triggered by reads caused by a D128 translation table walk. Fields:

ABORTED bool

Was the walk successful.

DATA_HIGH unsigned int

The data upper bits read.

DATA_LOW unsigned int

The data lower bits read.

IPA unsigned int

For stage 1, the IPA of the read.

LEVEL unsigned int

Translation table level.

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

OUTADDR unsigned int

The output physical address.

PADDR unsigned int

The physical address of the read.

PASmem enum

Physical Address Space of the accessed memory.

PASreq enum

Physical Address Space of the requested lookup.

REGIME_EL enum

Entry matches in this translation regime.

REQUESTER enum

The requester of this table walk.

SIDE enum

Inst / Data.

STAGE unsigned int

Translation stage.

MMU_D128_TTB_WRITE

This event is triggered by writes caused by a D128 translation table walk. Fields:

ABORTED bool

Was the walk successful.

DATA_HIGH unsigned int

The data upper bits written.

DATA_LOW unsigned int

The data lower bits written.

IPA unsigned int

For stage 1, the IPA of the write.

IS_HACDBS_ACCESS bool

If the access was due to HACDBS_CLEAN.

LEVEL unsigned int

Translation table level.

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

OUTADDR unsigned int

The output physical address.

PADDR unsigned int

The physical address of the write.

PASmem enum

Physical Address Space of the accessed memory.

PASreq enum

Physical Address Space of the requested lookup.

REGIME_EL enum

Entry matches in this translation regime.

SIDE enum

Inst / Data.

STAGE unsigned int

Translation stage.

MMU_TRANS

Address translation information. Fields:

ASID unsigned int

Address space identifier.

CORE_NUM unsigned int

Core number in a multi processor.

Hyp bool

Entry matches in Hyp state only.

INNERCACHE_RA bool

Is the inner cache allocate on read.

INNERCACHE_TRANSIENT bool

Is the inner write-through transient.

INNERCACHE_TYPE enum

Inner Caching scheme (NC/MB/WA).

INNERCACHE_WA bool

Is the inner cache allocate on write.

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MEMTYPE enum

Memory type.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC enum

Is secure side supposed to access secure or nonsecure memory.

OUTERCACHE_RA bool

Is the outer cache allocate on read.

OUTERCACHE_TRANSIENT bool

Is the outer write-through transient.

OUTERCACHE_TYPE enum

Outer Caching scheme (NC/MB/WA).

OUTERCACHE_WA bool

Is the outer cache allocate on write.

PADDR unsigned int

Physical address of the access.

PAGESIZE unsigned int

Page size as log2(size).

PAS enum

PAS supposed to be accessed is secure or nonsecure memory or Root/Realm if RME is implemented.

SH enum

Shareability.

SIDE enum

Inst / Data.

STAGE1_PERM unsigned int

Stage 1 permissions, [5:3] Privileged access (XWR) [2:0] User access (XWR).

STAGE2_PERM unsigned int

Stage 2 permission mask [5:3] Privileged access (XWR) [2:0] User access (XWR).

VADDR unsigned int

Virtual address of the access.

VMID unsigned int

Virtual machine identifier.

XS bool

XS Attribute.

nG enum

Flag indicating whether ASID will be matched.

MMU_TTB_READ

This event is triggered by reads caused by a translation table walk. Fields:

ABORTED bool

Was the walk successful.

DATA unsigned int

The data read.

DESC_KIND enum

The encoding scheme of the descriptors used for this walk.

IPA unsigned int

For stage 1, the IPA of the read.

LEVEL unsigned int

Translation table level.

LPAE bool

Is this for an LPAE translation (*DEPRECATED*, use DESC_KIND).

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSmem bool

Non secure memory access.

NSreq bool

Non secure request.

OUTADDR unsigned int

The output physical address.

PADDR unsigned int

The physical address of the read.

PASmem enum

Physical Address Space of the accessed memory.

PASreq enum

Physical Address Space of the requested lookup.

REGIME_EL enum

Entry matches in this translation regime.

REQUESTER enum

The requester of this table walk.

SIDE enum

Inst / Data.

STAGE unsigned int

Translation stage.

MMU_TTB_WRITE

This event is triggered by writes caused by a translation table walk. Fields:

ABORTED bool

Was the walk successful.

DATA unsigned int

The data written.

DESC_KIND enum

The encoding scheme of the descriptors used for this walk.

IPA unsigned int

For stage 1, the IPA of the write.

IS_HACDBS_ACCESS bool

If the access was due to HACDBS_CLEAN.

LEVEL unsigned int

Translation table level.

LPAE bool

Is this for an LPAE translation (*DEPRECATED*, use DESC_KIND).

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSmem bool

Non secure memory access.

NSreq bool

Non secure request.

OUTADDR unsigned int

The output physical address.

PADDR unsigned int

The physical address of the write.

PASmem enum

Physical Address Space of the accessed memory.

PASreq enum

Physical Address Space of the requested lookup.

REGIME_EL enum

Entry matches in this translation regime.

SIDE enum

Inst / Data.

STAGE unsigned int

Translation stage.

MODE_CHANGE

Mode change. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

MODE enum

The new mode.

NON_SECURE enum

The core's new non-secure bit.

OLD_MODE enum

The old mode.

SECURITY_STATE enum

The new security state.

NOT_TAG_CHECKED_INST

MTE Tag not checked for current instruction.

PERIODIC

Called for every quantum. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The instruction count of this CPU.

PC unsigned int

The address of the next instruction to be executed on this CPU.

PMU_COUNTER_OVERFLOW

PMU counter overflow. Fields:

COUNTER_GROUP enum

Counter group.

COUNTER_RANGE enum

Counter range.

EVENT_ID unsigned int

Event Identifier, field is valid only for Event counter group.

INDEX unsigned int

Counter index (as selected by PMSELR), field is valid only for Event/Cycle counter groups.

INTERRUPT bool

Is interrupt enabled on overflow for this counter.

POTINDEX_UPDATE

Value of POT index for current pc changed. Fields:

NEW_POTINDEX unsigned int

New POTIndex.

OLD_POTINDEX unsigned int

Old POTIndex.

PREFETCH_MEMORY64

Prefetch from PRFM or PRFUM instructions. Fields:

PRFOP enum

Prefetch hint.

VADDR unsigned int

Virtual address of the location that should be prefetched.

PRELOAD64_REQUEST

Preload requests from PRFM or PRFUM instructions. Fields:

PRFOP enum

Prefetch hint.

VADDR unsigned int

Virtual address of the location that should be prefetched.

PRELOAD_DATA

Data preload from PLD instruction. Fields:

VADDR unsigned int

Virtual address of the data that should be preloaded.

PRELOAD_INST

Instruction preload from PLI instruction. Fields:

VADDR unsigned int

Virtual address of the instruction that should be preloaded.

PRE_CORE_LOAD

Trace just before a core load. Fields:

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

PRE_CORE_STORE

Trace just before a core store. Fields:

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

PRE_TTB_READ

This event is triggered before reads caused by a translation table walk. Fields:

ADDR unsigned int

The physical address of the read.

PSTATE

PSTATE fields update. Fields:

ALLINT unsigned int

All interrupt mask bit.

BTYPE unsigned int

Branch target identification bit.

CORE_NUM unsigned int

Core number in a multi processor.

DAIF unsigned int

Exception mask bits.

DIT unsigned int

Data Independent Timing bit.

EL unsigned int

Current Exception level field.

EXLOCK unsigned int

Exception return state lock bit.

IL unsigned int

Illegal Execution state bit.

NZCV unsigned int

Condition flags.

PACM unsigned int

PACM bit.

PAN unsigned int

Privileged Access Never state bit.

PM unsigned int

PMU exception mask bit.

PPEND unsigned int

PMU exception pending bit.

SM unsigned int

StreamingMode status bit.

SP unsigned int

Stack pointer register selection bit.

SS unsigned int

Software Step bit.

SSBS unsigned int

Speculative Store Bypass Safe bit.

TCO unsigned int

Tag Check Override bit.

TOUCHED unsigned int

Mask of the fields that have been touched since this trace has last been emitted. Note that touched does not necessarily mean the value changed. The index of the field is the bit index in the mask plus the index of this field plus one.

UAO unsigned int

User Access Override bit.

UINJ unsigned int

Undefined Instruction exceptions bit.

ZA unsigned int

ZA storage and ZT0 register enable bit.

nRW unsigned int

Current Execution state bit.

PSTATE_PACTIVE_OPMODE_TRANSITION

Trace active and Opmode mode transitions on PChannel. Fields:

DEVPACTIVE **unsigned int**

Integer representation of DEVPACTIVE[] signals.

NEW_STATE **enum**

Operating mode after the transition.

OLD_STATE **enum**

Operating mode before the transition.

PSTATE_PACTIVE_OPMODE_TRANSITION_DENIED

Trace the denied active and Opmode mode transitions on PChannel. Fields:

DEVPACTIVE **unsigned int**

Integer representation of DEVPACTIVE[] signals.

NEW_STATE **enum**

Operating mode after the transition.

OLD_STATE **enum**

Operating mode before the transition.

REASON **string**

Reason for the transition being denied.

PSTATE_PACTIVE_POWER_TRANSITION

Trace active and power mode transitions on PChannel. Fields:

DEVPACTIVE **unsigned int**

Integer representation of DEVPACTIVE[] signals.

NEW_STATE **enum**

Power mode after the transition.

OLD_STATE **enum**

Power mode before the transition.

PSTATE_PACTIVE_POWER_TRANSITION_DENIED

Trace the denied active and power mode transitions on PChannel. Fields:

DEVPACTIVE **unsigned int**

Integer representation of DEVPACTIVE[] signals.

NEW_STATE **enum**

Power mode after the transition.

OLD_STATE **enum**

Power mode before the transition.

REASON **string**

Reason for the transition being denied.

PSTATE_PACTIVE_THREAD_OPMODE_TRANSITION

Trace pactive and Opmode mode transitions for threads on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Operating mode after the transition.

OLD_STATE enum

Operating mode before the transition.

QUANTUM_START

Called at the start of every quantum.

RANGE_PRELOAD_REQUEST

Preload requests from RPRFM. Fields:

COUNT unsigned int

Number of blocks of data to access.

LENGTH unsigned int

Number of contiguous bytes to be accessed in each block.

OP unsigned int

Range prefetch operation .

REUSE unsigned int

Maximum number of bytes to be accessed before next RPRFM .

STRIDE signed int

Stride to advance after accessing LENGTH bytes.

VADDR unsigned int

Base virtual address of range prefetch request.

RUN_STATE

Run state transition. Fields:

INST_COUNT unsigned int

Ticks count at point of transition.

NEW enum

New run state.

OLD enum

Old run state.

SBP_SAMPLE_COLLISION

DISPLAY Statistical profiling samples collided .

SBP_SAMPLE_DISCARDED

DISPLAY Operation discarded by statistical profiling unit when configured in discard mode.
Fields:

BUFFER_POINTER unsigned int

Value of buffer pointer.

DATA unsigned int

Data discarded.

SIZE unsigned int

Size of sample, in bytes.

SBP_SAMPLE_END

DISPLAY Statistical profiling of an operation ended.

SBP_SAMPLE_POP

DISPLAY Operation which might be sampled by statistical profiling.

SBP_SAMPLE_RECORDED

DISPLAY Operation sampled by statistical profiling has been written to the buffer at %{BUFFER_ADDRESS} with size %{SIZE} and data %{DATA}. Fields:

BUFFER_ADDRESS unsigned int

Virtual Address of start of buffer write.

DATA unsigned int

Data written to the buffer.

SIZE unsigned int

Size of sample written to buffer, in bytes.

SBP_SAMPLE_START

DISPLAY Statistical profiling of an operation started.

SBP_SAMPLE_TAKEN

DISPLAY Operation sampled by statistical profiling after filtering.

SEMIHOSTING_CALL

Call of a semihost function occurred. Fields:

PC unsigned int

The program counter after the semihosting call.

SEMIHOSTING_PRECALL

About to call semihost. Fields:

EL2 bool

Use EL2 translation regime.

NS bool

Is Non-Secure.

REG_WIDTH unsigned int

The current register width in bytes.

SIGNAL

External signal state change. Fields:

SIGNAL *enum*

Signal that changed.

STATE *bool*

Signal asserted state.

SOFTWARE_STEP

Return the Debug Software Step state. Fields:

STATE *enum*

Software Step state.

SPSR

SPSR change. Fields:

CORE_NUM *unsigned int*

Core number in a multi processor.

MODE *enum*

Which of the banked SPSR registers is written.

OLD_VALUE *unsigned int*

The old SPSR value.

VALUE *unsigned int*

The new SPSR value.

START_COMPILE

Compilation started. Fields:

VADDR *unsigned int*

Instruction where compilation begins.

STG_MULTIPLE_MICRO_OPS_ADDRESS_RANGE

Address range for MTE stores with multiple uops. Fields:

ADDR_END *unsigned int*

Final address.

ADDR_START *unsigned int*

Starting address.

STORE_MULTIPLE_REGS_ADDRESS_RANGE

Address range for stores with multiple registers. Fields:

ADDR_END *unsigned int*

Final address.

ADDR_START *unsigned int*

Starting address.

SUBOPTIMAL_LDST_RETIRED

suboptimal load/store. Fields:

ACCESS_SIZE unsigned int

Log2 of the access width in bytes used for alignment checking (1: half-word, 2: word ...).

OPTIMAL_ALIGN_SELECTOR unsigned int

Selector to decide the optimal alignment size (1: access width, 2: 4-bytes, 3: 8-bytes, ..., 12: 4-kbytes).

VADDR unsigned int

The virtual address of the access.

SYNC

Called for every synchronization, that is every quantum break. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The instruction count of this CPU.

LOCAL_QUANTUM unsigned int

The local quantum of this CPU.

LOCAL_TIME unsigned int

The local time of this CPU.

SYNC_REASON enum

Reason for sync.

SYSCALL

System call instruction executed. Fields:

IMM unsigned int

Immediate value of the system call instruction.

TYPE enum

System call type.

VADDR unsigned int

Instruction that caused the system call.

SYSREG_READ128

System Coprocessor 128 bit register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

REGNUM enum

Internal register number.

REGNUM_PHYS enum

Internal register number physically accessed (e.g. in presense of VHE redirection).

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_READ64

System Coprocessor register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REGNUM enum

Internal register number.

REGNUM_OPERAND enum

Register number corresponding to the instruction operand.

REGNUM_PHYS enum

Internal register number physically accessed (e.g. in presense of VHE redirection).

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_UPDATE128

Triggers when the system updates a register. Fields:

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG enum

Register number.

UNKNOWN unsigned int

Bits of the register which became unknown.

VALUE unsigned int

Value written to the register.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_UPDATE32

Triggers when the system updates a register. Fields:

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG enum

Register number.

UNKNOWN unsigned int

Bits of the register which became unknown.

VALUE unsigned int

Value written to the register.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_UPDATE64

Triggers when the system updates a register. Fields:

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG enum

Register number.

UNKNOWN unsigned int

Bits of the register which became unknown.

VALUE unsigned int

Value written to the register.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_WRITE128

System Coprocessor 128 bit register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

REGNUM enum

Internal register number.

REGNUM_PHYS enum

Internal register number physically accessed (e.g. in presense of VHE redirection).

UNDEF bool

The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_WRITE64

System Coprocessor register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REGNUM enum

Internal register number.

REGNUM_OPERAND enum

Register number corresponding to the instruction operand.

REGNUM_PHYS enum

Internal register number physically accessed (e.g. in presense of VHE redirection).

UNDEF bool

The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

UNALIGNED_LDST_RETIRE

Processor unaligned load/store. Fields:

ACCESS_SIZE unsigned int

Log2 of the access width in bytes used for alignment checking (1=>half-word, 2->word ...).

NUMBER_OF_BEATS unsigned int

Number of accesses (beats) in this burst (Total size in bytes is ACCESS_SIZE * NUMBER_OF_BEATS).

VADDR unsigned int

The virtual address of the access.

VFP_D_REGS

VFP/NEON D 64 bit register write. Fields:

ALIASING enum

Type of register aliasing used.

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a write to the corresponding bit in the register. For example, 0x1 would indicate a write to VALUE[0].

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

VFP_D_REGS_READ

VFP/NEON D 64 bit register read. Fields:

ID unsigned int

The register number.

VALUE unsigned int

The current value of the register.

VFP_Q_REGS

VFP/NEON Q 128 bit register write. Fields:

ALIASING enum

Type of register aliasing used.

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a write to the corresponding byte in the register. For example, 0x1 would indicate a write to VALUE[7:0].

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

VFP_Q_REGS_READ

VFP/NEON Q 128 bit register read. Fields:

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a read to the corresponding byte in the register. For example, 0x1 would indicate a read to VALUE[7:0].

VALUE unsigned int

The current value of the register.

VFP_SYS_REGS

Writes to the VFP/NEON units system registers. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ID enum

Which VFP system register is written.

MASK unsigned int

The mask indicating updated cumulative bits.

OLD_VALUE unsigned int

The register's old value overwritten.

VALUE unsigned int

The new value written to the VFP system register.

VFP_S_REGS

VFP/NEON S 32 bit register write. Fields:

ALIASING enum

Type of register aliasing used.

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a write to the corresponding bit in the register. For example, 0x1 would indicate a write to VALUE[0].

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

VFP_S_REGS_READ

VFP/NEON S 32 bit register read. Fields:

ID unsigned int

The register number.

VALUE unsigned int

The current value of the register.

VIRTUAL_TAG_ALLOCATED

Virtual tag information when allocated. Fields:

TAG_VADDR unsigned int

Virtual tag virtual address.

VADDR unsigned int

Data Virtual address.

VIRTUAL_TAG unsigned int

Virtual tag value.

WAYPOINT

Signals end of basic block execution. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ISET enum

Origin instruction set.

IS_COND bool

Indicates if this is a conditional waypoint.

PC unsigned int

Origin address (or 0 if unavailable).

TAKEN bool

Indicates if this waypoint was taken.

TARGET unsigned int

Destination address.

TARGET_ISET enum

Destination instruction set.

WFE_END

WFE ended. Fields:

INST_COUNT unsigned int

Ticks count when leaving WFE.

WFE_EVENT_REGISTER

WFE event register status: set/clear, reason. Fields:

INST_COUNT unsigned int

Ticks count.

REASON enum

Reason for set/clear. Only REASON==0 (Cleared by WFE) clears the bit, all other reasons set it.

WFE_IGNORED

WFE ignored. Fields:

EVENT bool

This WFE was ignored because the event register was set.

INST_COUNT unsigned int

Ticks count when ignoring WFE.

TRAPPED bool

This WFE was trapped.

WFE_START

WFE entered. Fields:

INST_COUNT unsigned int

Ticks count when entering WFE.

WFI_END

WFI ended. Fields:

INST_COUNT unsigned int

Ticks count when leaving WFI.

WFI_IGNORED

WFI ignored. Fields:

DISABLED bool

This WFI was ignored because WFI is disabled.

INST_COUNT unsigned int

Ticks count when ignoring WFI.

REASON enum

specifies reason why WFI trace was ignored.

TRAPPED bool

This WFI was trapped.

WFI_START

WFI entered. Fields:

INST_COUNT unsigned int

Ticks count when entering WFI.

WFI_WAKEUP

WFI wakeup. Fields:

INST_COUNT unsigned int

Ticks count when WFI wakeup occurred.

REASON enum

Reason for wakeup.

2.35 ARM_Cortex-A78

This section describes the trace sources.

AA64_ASE_SVE_REGS

Modification of AArch64 Advanced SIMD or SVE Vector Registers (via SIMD or Scalar operations), SVE Predicate Registers and ZTO. Fields:

ID enum

The register identifier.

MASK unsigned int

Bitmask of the register to signal the modified bits in the VALUE field.

SM bool

Whether the PE is in Streaming Mode.

VALUE unsigned int

Value of AArch64 vector register.

ASYNC_MEMORY_FAULT

Context ID Register write. Fields:

FAULT unsigned int

Fault status present in ESR.ISS encoding for Serror in ARCH64 and present in DFSR encoding for external abort format in ARCH32.

PADDR unsigned int

Physical Address (or 0 if unavailable).

VADDR signed int

Virtual Address (or 0 if unavailable).

ATOMIC_END_ACCESS

Bus trace access for atomics. Fields:

ACCESS_FAIL bool

Memory access failed.

ADDR unsigned int

The virtual address of the access.

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

COMPARE_VALUE unsigned int

Compare value for CAS.

LOAD_VALUE unsigned int

Loaded values.

NSDESC unsigned int

The security state of the access.

OPERAND_VALUE unsigned int

Operation's operand.

OPERATION enum

Operation type.

PADDR **unsigned int**

The physical address of the access.

PRIV **bool**

Is this a privileged access?.

ATOMIC_START_ACCESS

Bus trace access for atomics. Fields:

ADDR **unsigned int**

The virtual address of the access.

ATTR **unsigned int**

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

COMPARE_VALUE **unsigned int**

Compare value for CAS.

NSDESC **unsigned int**

The security state of the access.

OPERAND_VALUE **unsigned int**

Operation's operand.

OPERATION **enum**

Operation type.

PADDR **unsigned int**

The physical address of the access.

PRIV **bool**

Is this a privileged access?.

ArchMsg.Error.exit_code

DISPLAY exit code: %{EXITCODE} component kind: %{KIND} reason: %{REASON}. Fields:

EXITCODE **signed int**

Exit code to be returned.

KIND **string**

Component kind that invoked the exit code trace.

REASON **string**

Reason why the exit code trace was invoked (optional).

ArchMsg.Warning.IgnoredWarmResetRequest

Event fires when software attempts to request warm reset through write to deprecated DBGPRCR.CWRR.

ArchMsg.Warning.RAS_ErrselrUnpredictableState

DISPLAY ERRSELR is in unpredictable state due to value greater than number_of_error_records written to it.

ArchMsg.Warning.RAS_UnpredictableStateWrite

DISPLAY %{REG_NAME} write ignored due to unpredictable ERRSELR state. Fields:

REG_NAME string

Name of register being written.

ArchMsg.Warning.ReportedRevisionRegisterRead

DISPLAY Reported revision and patch do not match that of the cpu. Fields:

ORIGINAL_VALUE unsigned int

The value the register would have based on the modelled revision.

REGISTER string

The name of the register that has ben read.

REPORTED_VALUE unsigned int

The value the register has been overridden to.

ArchMsg.Warning.Unpredictably Indexed PM Event Register

TAGS **UNPREDICTABLE** DISPLAY Event %{IsDirect:(Selector value|Register)} %{SEL} is >= %{N}. Fields:

IsDirect unsigned int

Direct Access rather than indirect.

N unsigned int

Number of Accessible Registers.

SEL unsigned int

Selector.

ArchMsg.Warning.cache_contents_unknown

A cache has been activated without being invalidated and may contain **UNKNOWN** content. DISPLAY %{SIDE:(D|I)}-Cache was enabled in %{REGIME:(SECURE|NON_SECURE|HYP)} regime but was not invalidated since power-on: cache lines could contain **UNKNOWN** content %{SIDE: (Note: D-side caches must invalidate SECURE lines even if the cache is only used by NON_SECURE code in order to ensure that unexpected dirty lines tagged SECURE are not naturally evicted)}. Fields:

REGIME unsigned int

Current translation regime.

SIDE unsigned int

0 d-side cache or unified cache, 1 i-side.

ArchMsg.Warning.dcimvac_matches_watchpoint

TAGS **UNPREDICTABLE** DISPLAY Watchpoints matching an AArch32 DCIMVA are implementation defined.

ArchMsg.Warning.decode_fieldmismatch

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with different fields %{FIELD1} and %{FIELD2}. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_fracbitsnegative

TAGS **UNPREDICTABLE** DISPLAY VCVT Instruction is unpredictable with a negative number of fraction bits.

ArchMsg.Warning.decode_initblock

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable in an IT block.

ArchMsg.Warning.decode_initblocknotlast

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable in an IT block when not the last.

ArchMsg.Warning.decode_invalid_condition

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable when the condition is not AL.

ArchMsg.Warning.decode_invalidfieldcombination

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with the values in %{FIELD1} and %{FIELD2}. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_invalidvalue

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with the value in %{FIELD}. Fields:

FIELD enum

Field name.

ArchMsg.Warning.decode_mem_hint_unallocated

TAGS **UNPREDICTABLE** DISPLAY There is no memory hint allocated to this bit pattern.

ArchMsg.Warning.decode_registermatch

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with %{FIELD1} and %{FIELD2} as the same register. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_registermismatch

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with %{FIELD1} and %{FIELD2} as different registers. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_registeroutofrange

TAGS **UNPREDICTABLE** DISPLAY Floating-point instruction is unpredictable indexing beyond the end of the register bank.

ArchMsg.Warning.decode_sbzsbo

TAGS **UNPREDICTABLE** DISPLAY Reserved bits in the instruction are not canonical.

ArchMsg.Warning.decode_transactiontoobig

TAGS **UNPREDICTABLE** DISPLAY Attempt to perform a memory transaction of more than 128 bytes.

ArchMsg.Warning.decode_unpred_other

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable.

ArchMsg.Warning.decode_unpreduseofpc

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable when %{FIELD} is PC. Fields:

FIELD enum

Field name.

ArchMsg.Warning.decode_unpreduseofr13

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable when %{FIELD} is R13. Fields:

FIELD enum

Field name.

ArchMsg.Warning.decode_unpreduseofr13inreglist

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with R13 in register-list.

ArchMsg.Warning.decode_useofd16orhigher

TAGS **UNPREDICTABLE** DISPLAY Attempt to use register D16 or higher where it is not allowed.

ArchMsg.Warning.decode_writebackandbaseinlist

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with write back when the base register is in the transfer list.

ArchMsg.Warning.decode_zeroregistersinlist

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable when no registers are to be transferred.

ArchMsg.Warning.recursive_branch

An instruction is performing a branch that targets the same instruction. If this was intended then a WFI might be more effective. DISPLAY recursive branch detected at PC=%{PC}.

PRIMARY KEY PC. Fields:

PC unsigned int

Address of instruction causing the branch.

ArchMsg.Warning.recursive_exception

An instruction has generated an exception that targets the same instruction. DISPLAY recursive %{TYPE} exception detected at PC=%{PC} (CPSR=%{CPSR}). PRIMARY KEY PC. Fields:

CPSR unsigned int

Processor state.

PC unsigned int

Address of instruction causing the exception.

TYPE enum

Type of exception.

ArchMsg.Warning.reentrant_vector_catch

TAGS **UNPREDICTABLE** DISPLAY Vector catch debug event on Prefetch or Data abort vector %{ADDR}. Fields:

ADDR unsigned int

vector address.

ArchMsg.Warning.reserved_it_state

TAGS **UNPREDICTABLE** DISPLAY Execution with reserved IT state %{ITSTATE}. Fields:

ITSTATE unsigned int

reserved IT state.

ArchMsg.Warning.tlb_contents_unknown

Translation has been enabled without invalidating the TLB. DISPLAY Translation was enabled in the %{PAS:(Secure|Non-secure|Root|Realm)} EL%{EL:d} translation regime (vmid=%{VMID:x}) without invalidating %{INVALIDITY:(|d-side |i-side |)}entries since power-on: %{INVALIDITY:(|D||I)}TLB could contain **UNKNOWN** content. Fields:

EL unsigned int

Exception level of translation regime.

INVALIDITY unsigned int

Which TLB may be invalid (if TLBs are separate).

NON_SECURE bool

Whether the translation regime is non-secure.

PAS unsigned int

Physical address space of translation regime.

VMID unsigned int

Current VMID, if applicable.

ArchMsg.Warning.unknown_DLR_DSPSR

Exiting debug state with unknown DLR or DSPSR.

ArchMsg.Warning.unknown_ELR_SPSR

Returning from debug exception with unknown ELR or SPSR.

ArchMsg.Warning.unpred_FAR_on_LRCPC3_non_atomic_mem_access

TAGS **UNPREDICTABLE** Unpredictable FAR/FSTATUS on LRCPC3 pair instructions resulting in two separate memory accesses
DISPLAY Unpredictable FAR on LRCPC3 non atomic memory access at %{ADDR1} with FSTATUS %{FSTATUS1} or %{ADDR2} with FSTATUS %{FSTATUS2}. Fields:

ADDR1 unsigned int

Possible FAR address 1.

ADDR2 unsigned int

Possible FAR address 2.

FSTATUS1 unsigned int

Possible fault status value 1.

FSTATUS2 unsigned int

Possible fault status value 2.

ArchMsg.Warning.unpred_async_tag_check_fail_on_sve_speculative_load_do_update_ffr

Unpredictable set to FALSE of the FFR element associated to SVE speculative load with an async tag check fail TAGS **UNPREDICTABLE**.

ArchMsg.Warning.unpredictable_a32_breakpoint

TAGS **UNPREDICTABLE** DISPLAY Unpredictable breakpoint on a A32 instruction at %{ADDR1} Offset %{OFFSET} BAS %{BAS} for a %{IS_ADDRESS_MISMATCH_BREAKPOINT:(Mismatch|Match)} Breakpoint number %{N:d} Breakpoint address %{ADDR2}. Fields:

ADDR1 unsigned int

Address.

ADDR2 unsigned int

Breakpoint address.

BAS unsigned int

BAS field.

IS_ADDRESS_MISMATCH_BREAKPOINT bool

Is a Mismatch breakpoint, not a Match breakpoint.

N unsigned int

Breakpoint number.

OFFSET unsigned int

Always has a value of zero.

ArchMsg.Warning.unpredictable_banked_register_access

TAGS **UNPREDICTABLE** DISPLAY Access to banked register (R=%{R:d} SYSm=%{SYSM:2x}) is unpredictable. Fields:

R unsigned int

R bit.

SYSM unsigned int

SYSm.

ArchMsg.Warning.unpredictable_ccfail_undef

TAGS **UNPREDICTABLE** DISPLAY An instruction %{OPCODE} at VA=%{VADDR} failed its condition codes check but would otherwise have caused a non-data-dependent trap or undefined exception. Behaviour is implementation-defined choice of exception or nop. Fields:

OPCODE unsigned int

opcode of the instruction.

VADDR unsigned int

Virtual address of the instruction being executed.

ArchMsg.Warning.unpredictable_ici_bits

TAGS **UNPREDICTABLE** DISPLAY ICI bits do not apply or are architecturally unpredictable for instruction.

ArchMsg.Warning.unpredictable_t32_breakpoint

TAGS **UNPREDICTABLE** DISPLAY Unpredictable breakpoint on a T32 instruction at %{ADDR1} Offset %{OFFSET} BAS %{BAS} for a %{IS_ADDRESS_MISMATCH_BREAKPOINT:(Mismatch|Match)} Breakpoint number %{N:d} Breakpoint address %{ADDR2}. Fields:

ADDR1 unsigned int

Address.

ADDR2 unsigned int

Breakpoint address.

BAS unsigned int

BAS field.

IS_ADDRESS_MISMATCH_BREAKPOINT bool

Is a Mismatch breakpoint, not a Match breakpoint.

N unsigned int

Breakpoint number.

OFFSET unsigned int

Indicates which halfword of a 32-bit Thumb instruction the breakpoint is set on. 0 for the first halfword or if the instruction is 16 bits wide, or 1 for the second halfword.

ArchMsg.Warning.unpredictable_t32_vector_catch

TAGS **UNPREDICTABLE** DISPLAY Unpredictable Vector catch exception on a T32 instruction at %{ADDR} Offset %{OFFSET}. Fields:

ADDR unsigned int

Address.

OFFSET unsigned int

Indicates which halfword of a 32-bit Thumb instruction the vector catch exception is set on. 0 for the first halfword or if the instruction is 16 bits wide, or 1 for the second halfword.

ArchMsg.Warning.unpredictable_watchpoint_far

TAGS **UNPREDICTABLE** DISPLAY The value for FAR/EDWAR for hitting this watchpoint may be anywhere in the range %{LOWER_BOUND} to %{UPPER_BOUND}. Fields:

LOWER_BOUND unsigned int

the lowest address accessed by the instruction that triggered the watchpoint.

UPPER_BOUND unsigned int

the highest watchpointed address accessed by that instruction.

ArchMsg.Warning.unpredictable_watchpoint_far_isscmp

DISPLAY The value for FAR/EDWAR for hitting this watchpoint may be anywhere in the range %{LOWER_BOUND} to %{UPPER_BOUND}. Fields:

LOWER_BOUND unsigned int

the lowest address accessed by the instruction that triggered the watchpoint.

UPPER_BOUND unsigned int

the highest watchpointed address accessed by that instruction.

ArchMsg.Warning.warning_AdvSIMDExpandImmUnexpectedZero

TAGS **UNPREDICTABLE** DISPLAY AdvSIMDExpandImm may treat this immediate value as **UNPREDICTABLE**.

ArchMsg.Warning.warning_ConditionalSMC

TAGS **UNPREDICTABLE** DISPLAY SMC instruction has **UNPREDICTABLE** effects when conditional and when combined with traps.

ArchMsg.Warning.warning_TLBID_out_of_range

TLBID out of range in TLBI according to the configuration of NIS, NVIS, NOS and NVOS. Fields:

NIS unsigned int

NIS.

NOS unsigned int

NOS.

NVIS unsigned int

NVIS.

NVOS unsigned int

NVOS.

TLBID unsigned int

The TLBID value used in the TLBI instruction.

ArchMsg.Warning.warning_TTL_or_TG_mismatches_current_translation_granule

TTL or TG mismatches current translation granule in TLBI VA/RVA/IPA/RIPA. Fields:

CURRENT_TG unsigned int

Current TG as indicated by TCR_ELx or VTCR_EL2.

TLBI_TG unsigned int

TG (or TTL[3:2]) used in TLBI instruction.

ArchMsg.Warning.warning_access_crosses_page_boundary

TAGS **UNPREDICTABLE** DISPLAY Access crosses page boundary.

ArchMsg.Warning.warning_access_crosses_page_boundary_has_fault

TAGS **UNPREDICTABLE** DISPLAY Access that crosses page boundary has fault(s).

ArchMsg.Warning.warning_access_crosses_page_boundary_has_fault_on_both_pages

TAGS **UNPREDICTABLE** DISPLAY Access that crosses page boundary has a fault on each page.
Fields:

ADDR1 unsigned int

FAR address on first page.

ADDR2 unsigned int

FAR address on second page.

FSTATUS1 unsigned int

Fault status on first page.

FSTATUS2 unsigned int

Fault status on second page.

ArchMsg.Warning.warning_access_crosses_page_boundary_spanning_different_memory

TAGS **UNPREDICTABLE** DISPLAY Access at address %{ADDR} crosses page from MEM_TYPE %{MEMTYPE_PAGE1} to %{MEMTYPE_PAGE2}. Fields:

ADDR unsigned int

address of access crossing page.

MEMTYPE_PAGE1 signed int

First Page's memory type (0-Normal, 1-Non-Normal).

MEMTYPE_PAGE2 signed int

Second Page's memory type (0-Normal, 1-Non-Normal).

ArchMsg.Warning.warning_access_wraps_around_memory

TAGS **UNPREDICTABLE** DISPLAY Access wraps around memory in %{WIDTH:d}bit mode.
Fields:

WIDTH unsigned int

Address width.

ArchMsg.Warning.warning_bcr_linking_status

TAGS **UNPREDICTABLE** DISPLAY DBGBCR.{BT2,BT} combination gives unpredictable behaviour.

ArchMsg.Warning.warning_bcr_mask_reserved

TAGS **UNPREDICTABLE** DISPLAY DBGBCR_ELn.MASK is set to a reserved value.

ArchMsg.Warning.warning_bvr_masked_bit_not_zero

TAGS **UNPREDICTABLE** DISPLAY DBGBVRn_EL1.VA contains a bit masked by DBGBCRn_EL1.MASK that is not zero.

ArchMsg.Warning.warning_ccsidr_unimplemented_level

TAGS **UNPREDICTABLE** DISPLAY CCSIDR read while CSSELR %{CSSELR:x} points to an unimplemented cache level. Fields:

CSSELR unsigned int

current effective value of CSSELR.

ArchMsg.Warning.warning_change_to_ns_when_tge_set

TAGS **UNPREDICTABLE** DISPLAY Attempting to change NS to 1 when HCR.TGE = 1.

ArchMsg.Warning.warning_contiguous_bit_check_abort

TAGS **UNPREDICTABLE** DISPLAY An abort occurred whilst attempting to check TLB entry is contiguous at %{ENTRY_ADDR}. Fields:

ENTRY_ADDR unsigned int

address of conflicting TLB entry.

ArchMsg.Warning.warning_contiguous_bit_error

TAGS **UNPREDICTABLE** DISPLAY TLB entry at %{ENTRY_ADDR} was not contiguous with entry at %{CONTIG_ADDR} Contents are %{ENTRY_DATA} but expected %{CONTIG_DATA}. Fields:

CONTIG_ADDR unsigned int

address of the TLB first read, with which this entry is expected to be contiguous.

CONTIG_DATA unsigned int

expected contents based on the entry at CONTIG_ADDR.

ENTRY_ADDR unsigned int

address of conflicting TLB entry.

ENTRY_DATA unsigned int

contents of conflicting TLB entry.

ArchMsg.Warning.warning_contiguous_bit_error_D128

TAGS **UNPREDICTABLE** DISPLAY TLB entry at %{ENTRY_ADDR} was not contiguous with entry at %{CONTIG_ADDR} Contents are %{ENTRY_DATA_HI} %{ENTRY_DATA_LO} but expected %{CONTIG_DATA}. Fields:

CONTIG_ADDR unsigned int

address of the TLB first read, with which this entry is expected to be contiguous.

CONTIG_DATA unsigned int

expected contents based on the entry at CONTIG_ADDR.

ENTRY_ADDR unsigned int

address of conflicting TLB entry.

ENTRY_DATA_HI unsigned int

Higher bit contents of conflicting TLB entry.

ENTRY_DATA_LO unsigned int

Lower bit contents of conflicting TLB entry.

ArchMsg.Warning.warning_cp10_cp11_mismatch

TAGS **UNPREDICTABLE** DISPLAY Unpredictable configuration of CP10 and CP11 controls at EL
%{EL:d}. Fields:

EL unsigned int

EL owning the control.

ArchMsg.Warning.warning_cp10_cp11_reserved_value

TAGS **UNPREDICTABLE** DISPLAY Unpredictable configuration of CPACR CP10 (%{CP10}) or
CP11 (%{CP11}). Fields:

CP10 unsigned int

CP10 access.

CP11 unsigned int

CP11 access.

ArchMsg.Warning.warning_csselr_level_out_of_range

TAGS **UNPREDICTABLE** DISPLAY CSSELR.Level out of range, written value %{LEVEL} with only
%{IMPLEMENTED:d} levels implemented. Fields:

IMPLEMENTED unsigned int

number of cache levels implemented.

LEVEL unsigned int

written value of CSSELR.Level.

TYPE enum

The type of cache selected by CSSELR.InD and TnD.

ArchMsg.Warning.warning_debug_flow_control_bits_not_obeyed

TAGS **UNPREDICTABLE** DISPLAY Flow control bits for %{REGISTER} were not obeyed, causing
%{ERROR}. Fields:

ERROR enum

type of error resulting.

REGISTER enum

register under consideration.

ArchMsg.Warning.warning_debug_register_access_during_reset

TAGS **UNPREDICTABLE** DISPLAY Unpredictable %{IS_WRITE:(read|write)} access to debug register offset %{OFFSET} during reset. Fields:

IS_WRITE unsigned int

Write Not Read.

OFFSET unsigned int

Register Offset.

ArchMsg.Warning.warning_decode_cps_inconsistent_fields

TAGS **UNPREDICTABLE** DISPLAY The mode setting fields in the CPS instruction are inconsistent.

ArchMsg.Warning.warning_decode_invalid_state

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable in current exception-level/security state.

ArchMsg.Warning.warning_default_cacheable_mmu_on

TAGS **UNPREDICTABLE** DISPLAY Default cacheable is enabled (HCR.DC = 1) while MMU is enabled.

ArchMsg.Warning.warning_default_cacheable_vmmu_off

TAGS **UNPREDICTABLE** DISPLAY Default cacheable is enabled (HCR.DC = 1) while HCR.VM = 0.

ArchMsg.Warning.warning_deprecated_wvr_bit_2_set

TAGS **UNPREDICTABLE** DISPLAY DBGWVR%{N:d}_EL1=%{WVR:h} has bit 2 set, deprecated in ARMv8. Fields:

N unsigned int

index.

WVR unsigned int

DBGWVR.

ArchMsg.Warning.warning_eret_while_software_step_active_pending

Missing ISB between setting MDSCR_EL1.SS and ERET TAGS **UNPREDICTABLE**.

ArchMsg.Warning.warning_exclusive_to_non_normal

TAGS **UNPREDICTABLE** DISPLAY Exclusive Access to Non-Normal Memory %{ADDR}. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_exclusive_to_non_writeback

TAGS **UNPREDICTABLE** DISPLAY Exclusive Access to Non-Writeback-Cacheable Memory %{ADDR}. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_execute_from_device_memory

TAGS **UNPREDICTABLE** DISPLAY An attempt was made to execute from Device memory.

ArchMsg.Warning.warning_illegal_cpsr_mode

TAGS **UNPREDICTABLE** DISPLAY Writing an illegal or unimplement mode ({NEW_MODE:h}) to CPSR was unpredictable. Fields:

NEW_MODE unsigned int

New value of CPSR.M.

ArchMsg.Warning.warning_illegal_srs_mode

TAGS **UNPREDICTABLE** DISPLAY Illegal or **UNPREDICTABLE** mode ({MODE:h}) used for SRS instruction. Fields:

MODE unsigned int

mode for Banked SP.

ArchMsg.Warning.warning_implementation_defined_read_debug_register_in_SCS

Read Access to Debug registers from the processor is **IMPLEMENTATION DEFINED** TAGS **IMP_DEF**. Fields:

PPB_OFFSET unsigned int

debug register offset.

ArchMsg.Warning.warning_implementation_defined_sequential_security_transitions_supported

The behavior of sequential instruction fetches that cross from non-secure to secure memory and contain an SG instruction is **CONSTRAINED UNPREDICTABLE** TAGS **UNPREDICTABLE**.

ArchMsg.Warning.warning_implementation_defined_stack_limit_check_supported

It is **IMPLEMENTATION DEFINED** whether stack pointer limit checking is performed for this instruction TAGS **IMP_DEF**. Fields:

ADDRESS unsigned int

Address of instruction.

OPCODE unsigned int

instruction opcode.

ArchMsg.Warning.warning_implementation_defined_taggednonsharedmemorytype

TAGS **IMP_DEF** DISPLAY Memory attribute has Tagged, non shared memory type; it is **IMPLEMENTATION DEFINED** whether the memory location is treated as Tagged or non-Tagged.

ArchMsg.Warning.warning_implementation_defined_write_debug_register_in_SCS

Write Access to Debug registers from the processor is **IMPLEMENTATION DEFINED** TAGS **IMP_DEF**. Fields:

DATA unsigned int

data attempted to be written.

PPB_OFFSET unsigned int

debug register offset.

ArchMsg.Warning.warning_invalid_tcr_granule

TAGS UNPREDICTABLE DISPLAY TCR.TG{%TG_ID:(0|1)} bits have been set to a granule size {%REQUEST:d}K not implemented - using {%SUBSTITUTE:d}K. Fields:

REQUEST signed int

page size requested (or 0 for reserved).

SUBSTITUTE signed int

best guess available page size.

TG_ID bool

bits TG0 or TG1.

ArchMsg.Warning.warning_load_multiple_user_registers_from_user_mode

TAGS UNPREDICTABLE DISPLAY An LDM(user registers) instruction executed from user mode.

ArchMsg.Warning.warning_mte_in_mem_attr_not_supported

TAGS UNPREDICTABLE DISPLAY Memory attribute {%ATTR} has Tagged memory type but Armv8.5-MTE is not implemented. Fields:

ATTR unsigned int

Memory attribute.

ArchMsg.Warning.warning_reserved_breakpoint_state_match

TAGS UNPREDICTABLE DISPLAY {%IS_BREAKPOINT:(Watchpoint|Breakpoint)} programmed with a reserved combination of HMC, SSC and {%IS_BREAKPOINT:(PAC|PMC)}. Fields:

IS_BREAKPOINT unsigned int

Is a breakpoint, not a watchpoint.

ArchMsg.Warning.warning_shareability

TAGS UNPREDICTABLE DISPLAY Unpredictable: combination of the 1st and 2nd stages of translation is Normal Inner Non-Cacheable, Outer Non-Cacheable.

ArchMsg.Warning.warning_software_step_set_while_enabled

TAGS UNPREDICTABLE DISPLAY MDSCR_EL1.SS set to 1 while software step debug exceptions are enabled.

ArchMsg.Warning.warning_thumb_instruction_wraps_around_memory

TAGS UNPREDICTABLE DISPLAY Thumb instruction wraps around memory.

ArchMsg.Warning.warning_ttbr_sbz_bits_are_not_zero

TAGS UNPREDICTABLE DISPLAY {%TTBR}[{%MSB:d}:0] should be zero, but are not. Fields:

MSB unsigned int

x-1.

TTBR enum

which TTBR register.

ArchMsg.Warning.warning_unaligned_address_dbgdtrrx_write

TAGS **UNPREDICTABLE** DISPLAY Unaligned address %{ADDR} in X0/R0 when DBGDTRRX written in memory access mode. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unaligned_address_dbgdtrtx_read

TAGS **UNPREDICTABLE** DISPLAY Unaligned address %{ADDR} in X0/R0 when DBGDTRTX read in memory access mode. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unaligned_to_device

TAGS **UNPREDICTABLE** DISPLAY Unaligned Access to Device Memory %{ADDR}. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unaligned_to_strongly_ordered

TAGS **UNPREDICTABLE** DISPLAY Unaligned Access to Strongly Ordered Memory %{ADDR}. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unknown_sau_rnr

TAGS **UNPREDICTABLE** DISPLAY SAU_RNR was set to an unsupported value.

ArchMsg.Warning.warning_unpredictable_AIRCR_PRIS_and_BFHFNMINs

TAGS **UNPREDICTABLE** DISPLAY The effect of setting both AIRCR.BFHFNMINs and AIRCR.PRIS to 1 is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_AIRCR_VECTCLRACTIVE_when_not_in_debug

TAGS **UNPREDICTABLE** DISPLAY The effect of writing a 1 to AIRCR.VECTCLRACTIVE if the processor is not halted in Debug state is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_AIRCR_VECTRESET_and_SYSRESETREQ

TAGS **UNPREDICTABLE** DISPLAY When the processor is halted in Debug state, if a write to the register writes a 1 to both VECTRESET and SYSRESETREQ, the behavior is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_AIRCR_VECTRESET_when_not_in_debug

TAGS **UNPREDICTABLE** DISPLAY The effect of writing a 1 to AIRCR.VECTRESET if the processor is not halted in Debug state is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_AIRCR_incorrect_VKEY

TAGS **UNPREDICTABLE** DISPLAY The value 0x05FA must be written to AIRCR.VKEY, otherwise the register write is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_EXC_RETURN_Reserved_Bit

EXC_RETURN[23:7] are reserved with the special condition that all bits should be written as one. Values other than all 1s are **UNPREDICTABLE**. TAGS **UNPREDICTABLE**. Fields:

EXC_RETURN unsigned int

Special PC value.

SBO_MASK unsigned int

Which bits Should Be One to not UNPRED.

ArchMsg.Warning.warning_unpredictable_PMEVTYPEPn_ELO_VS_reserved_value

TAGS **UNPREDICTABLE** DISPLAY Unpredictable reserved value of PMEVTYPEn<n>_ELO.VS.

ArchMsg.Warning.warning_unpredictable_change_to_DEMCR_MON_STEP_at_insufficient_priority

TAGS **UNPREDICTABLE** DISPLAY The effect of changing DEMCR.MON_STEP at an execution priority that is lower than the priority of the DebugMonitor exception is **CONSTRAINED UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_change_to_priority_of_active_exception

TAGS **UNPREDICTABLE** DISPLAY Changing the priority of an active exception is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_clear_lspact

TAGS **UNPREDICTABLE** DISPLAY **UNPREDICTABLE** state on context switch. Fields:

DRVEXC_TAKEN unsigned int

Derived exception taken, not pended.

REACHED_LAST_STORE unsigned int

Only the last faulted.

ArchMsg.Warning.warning_unpredictable_dbg_exit_unaligned_dlr

TAGS **UNPREDICTABLE** DISPLAY Debug exit to AArch32 with DLR[0] set to 1. PC[0] can be set to 0 or to DLR[0].

ArchMsg.Warning.warning_unpredictable_exception_catch

TAGS **UNPREDICTABLE** DISPLAY Generation of an exception catch event is unpredictable.

ArchMsg.Warning.warning_unpredictable_exception_return_inconsistent_state

TAGS **UNPREDICTABLE** DISPLAY state on exception return is unpredictable.

ArchMsg.Warning.warning_unpredictable_exception_return_instruction

TAGS **UNPREDICTABLE** DISPLAY exception return instruction is unpredictable.

ArchMsg.Warning.warning_unpredictable_in_debug_state

TAGS **UNPREDICTABLE** DISPLAY Instruction is **UNPREDICTABLE** when executed in debug state.

ArchMsg.Warning.warning_unpredictable_mair_encoding

TAGS **UNPREDICTABLE** DISPLAY Memory attribute %{ATTR} has invalid MAIR encoding; this may lead to unpredictable results. Fields:

ATTR unsigned int

Memory attribute.

ArchMsg.Warning.warning_unpredictable_pmu_counter_access

TAGS **UNPREDICTABLE** DISPLAY Access to PMU counters from non-secure EL0 or EL1 is **UNPREDICTABLE** with MDCR_EL2.HPMN set to 0.

ArchMsg.Warning.warning_unpredictable_prioritization_breakpoint_match_vector_catch

TAGS **UNPREDICTABLE** DISPLAY Two events with the same priority occurred at the same instruction: Address Matching Vector Catch debug event, and Breakpoint debug event. It is constrained unpredictable which is taken.

ArchMsg.Warning.warning_unpredictable_stack_selection

TAGS **UNPREDICTABLE** DISPLAY **UNPREDICTABLE** state on context switch.

ArchMsg.Warning.warning_unpredictable_tsize_out_of_range

TAGS **UNPREDICTABLE** DISPLAY %{TCR}.T%{TTBR:d}SZ (%{TSIZE:d} bits) is out of range. Must be between 25 and %{TSIZE_MAX:d}. Fields:

TCR enum

Which TCR.

TSIZE signed int

Value of TSize.

TSIZE_MAX signed int

The maximum allowed value.

TTBR unsigned int

Which TTBR region.

ArchMsg.Warning.warning_unpredictable_unaligned_pc_as_base_register

TAGS **UNPREDICTABLE** DISPLAY The PC value must be word-aligned, otherwise the behavior of the instruction is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_unaligned_pop_stack

TAGS **UNPREDICTABLE** DISPLAY **CONSTRAINED UNPREDICTABLE** unaligned stack frame %{FRAME_ADDR:x} in exception-return. Fields:

FRAME_ADDR unsigned int

unaligned stack frame address.

ArchMsg.Warning.warning_unpredictable_unaligned_to_non_writeback

TAGS **UNPREDICTABLE** DISPLAY Unaligned Access to Non-Writeback-Cacheable Memory.

ArchMsg.Warning.warning_unpredictable_vmsa_memattrib

TAGS **UNPREDICTABLE** DISPLAY Unpredictable vmsa memory attribute with texcb value programmed to %{TEXCB} when tex remap is %{TEX_REMAP}. Fields:

TEXCB unsigned int

texcb value.

TEX_REMAP bool

use tex remap.

ArchMsg.Warning.warning_unpredictable_vtcr_t0sz_sl0

TAGS **UNPREDICTABLE** DISPLAY The combination of VTCR.TOSZ=%{TOSZ:d} and VTCR.SLO=%{SLO:d} is unpredictable. Fields:

SLO signed int

VTCR.SLO, starting level for stage 2 translation table walks.

TOSZ signed int

VTCR.TOSZ, required input address range.

ArchMsg.Warning.warning_unpredictable_write_DHCSR

TAGS **UNPREDICTABLE** DISPLAY **UNPREDICTABLE** set of bit-changes in DHCSR. Fields:

NEW unsigned int

data attempted to be written.

OLD unsigned int

previous value.

ArchMsg.Warning.warning_unpredictable_write_SHCSR

TAGS **UNPREDICTABLE** DISPLAY **UNPREDICTABLE** setting a pending bit to 1 for an exception with priority >= execution priority.

ArchMsg.Warning.warning_unsupported_access_to_memory_mapped_register

TAGS **UNPREDICTABLE** DISPLAY Access to memory mapped register is unsupported.

ArchMsg.Warning.warning_user_jmcr_access

TAGS **UNPREDICTABLE** DISPLAY User %{WRITE:(write|read)} access to JMCR is unpredictable. Fields:

WRITE unsigned int

access is write.

ArchMsg.Warning.warning_wcr_mask_and_bas

TAGS **UNPREDICTABLE** DISPLAY DBGWCR_EL1.MASK is non-zero and BAS is not 0xff.

ArchMsg.Warning.warning_wcr_mask_reserved

TAGS **UNPREDICTABLE** DISPLAY DBGWCR_ELn.MASK is set to a reserved value.

ArchMsg.Warning.warning_wcr_non_configuous_bas

TAGS **UNPREDICTABLE** DISPLAY DBGWCR_EL1.BAS has non-contiguous set of ones.

ArchMsg.Warning.warning_wvr_masked_bit_not_zero

TAGS **UNPREDICTABLE** DISPLAY DBGWVRn_EL1.VA contains a bit masked by DBGWCRn_EL1.MASK that is not zero.

ArchMsg.Why.why_branch_target_exception

A branch target exception has occurred. DISPLAY branch target exception: target pc=%{PC}, target opcode=%{OPCODE}. PRIMARY KEY PC. Fields:

BTYPE unsigned int

pstate.BTYPE at time branch taken.

OPCODE unsigned int

opcode at branch target.

PC unsigned int

Virtual Address of branch target.

ArchMsg.Why.why_illegal_state

An illegal mode change has occurred. DISPLAY illegal mode change: %{MESSAGE}. Fields:

MESSAGE string

The message.

ArchMsg.warning_invalid_interval_counter_reload_value

Invalid interval counter reload value in PMSIRR_EL1.INTERVAL: software should set this to a value greater than or equal to the minimum indicated by PMSIDR_EL1.Interval.

ArchMsg.warning_invalid_spe_buffer_write_pointer

Invalid value programmed in PMBPTR_EL1.PTR or PMBLIMITR_EL1.LIMIT: Current write pointer must be at least one sample record below the write limit pointer and PMBPTR_EL1.PTR[63:56] must be equal to PMBLIMITR_EL1.LIMIT[63:56].

BRANCH_MISPREDICT

Simulating branch mispredict. Fields:

PC unsigned int

Origin address (or 0 if unavailable).

BRA_DIR

Direct branches, to immediate address. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISSET enum

The instruction set of the branch instruction.

IS_COND bool

Indicates if this is a conditional branch.

IS_HINTED bool

Whether the conditional instruction is hinted.

IS_ISB bool

Whether the instruction is ISB.

IS_LINK bool

Indicates if this is a branch with link.

PC unsigned int

The address of the branch instruction.

TARGET_ISET enum

The instruction set after the branch.

TARGET_PC unsigned int

The address the instruction branches to.

BRA_INDIR

Indirect branches, perhaps to a register. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISET enum

The instruction set of the branch instruction.

IS_COND bool

Indicates if this is a conditional branch.

IS_HINTED bool

Whether the conditional instruction is hinted.

IS_ISB bool

Whether the instruction is ISB.

IS_LINK bool

Indicates if this is a branch with link.

PC unsigned int

The address of the branch instruction.

TARGET_ISET enum

The instruction set after the branch.

TARGET_PC unsigned int

The address the instruction branches to.

BRA_INDIR_CCFAIL

Indirect branches fail, perhaps to a register.

BRA_RET_CCFAIL

Branch return fail, perhaps to a register.

CACHE_MAINTENANCE_OP

Cache Maintenance Operation. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

DATA unsigned int

Specified MVA or Set/way.

FUNCTION enum

Clean / Invalidate.

SCOPE enum

Affected region.

SIDE enum

Inst / Data.

CCFAIL

Conditional instruction condition check fail. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CCFAIL_UNC

Unconditional instruction condition check fail. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CCPASS

Conditional instruction condition check pass. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CCPASS_UNC

Unconditional instruction condition check pass. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CODE_CACHE_MAINT

Code cache maintenance. Fields:

END_INCL unsigned int

Inclusive end address for VA or PA requests.

NS bool

Non-secure world for PA.

START unsigned int

Start address for VA or PA requests.

TYPE enum

Request type.

COMPILE_BLOCK_END

Last instruction of basic block translated. Fields:

VADDR unsigned int

Address of next instruction after this basic block.

COMPILE_BLOCK_START

First instruction of basic block translated. Fields:

VADDR unsigned int

Address of first instruction in basic block.

COMPILE_INST

ARM instruction compiled. Fields:

DISASS string

Disassembly of the instruction.

ISSET enum

The instruction set of this instruction.

ITSTATE unsigned int

The ITSTATE current for the instruction.

OPCODE unsigned int

The opcode of the instruction.

PC unsigned int

The address of the instruction.

SIZE unsigned int

The size of the instruction in bytes.

CONTEXTIDR

Context ID Register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

NS bool

Secure or nonsecure banked register is accessed.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The new value written.

CONTEXT_SYNC

Called for every context synchronization event. Fields:

ARCH_CSE bool

Architectural Context Synchronization Event.

PC unsigned int

The address of the synchronization instruction.

SYSREG_SYNC bool

System register synchronization event.

CORE_ENDIAN

Core BE8 Big-Endian state changed. Fields:

BE8 bool

Core BE8 Big-Endian state.

CORE_INFO

Static processor attributes. Only triggered by a call to DumpState(). Fields:

ARCH_PROFILE enum

The architecture profile of the core.

CLUSTER_ID unsigned int

The cluster ID of this processor.

CORE_NUM unsigned int

The number of this core in an MP processor.

DETAILED_CORE_FEATURES enum

Detailed core features as a bitfield. Enum values are bitmasks to extract the info from the field.

FPU_VERSION enum

The VFP version implemented by the core.

MEM_ARCH enum

The memory architecture of the core.

MIDR_EL1 unsigned int

MIDR_EL1 value.

NUM_CORES unsigned int

The number of cores in this MP processor.

QUANTUM_SIZE unsigned int

The default quantum size of the core (in ticks).

SECURITY_FEATURES bool

Does the core have security features?.

CORE_LOADS

Processor load accesses. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ACQREL enum

Is this an acquire/release.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DATA unsigned int

The data read or written.

ELEMENT_SIZE unsigned int

Width of each element.

ENDIANNESS enum

Byte ordering of this access.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

LOCK enum

Normal, exclusive or locked access.

MECID unsigned int

Memory Encryption Context ID, if applicable.

MECID2 unsigned int

Memory Encryption Context ID of the second page, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PAGE_SIZE unsigned int

Page size as $\log_2(\text{size in bytes})$.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK, 2=Exclusive Failed, 3=Unsupported.

SIZE unsigned int

Width of the access in bytes. Only required if DATA is not traced.

TRANS bool

Is this a translated access.

VADDR unsigned int

The virtual address of the access.

CORE_REGS

Changes of the core registers R0 to R14. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number, 0 to 14.

MODE enum

Bank of the register accessed.

OLD_VALUE unsigned int

The old value overwritten.

PHYS_ID enum

The physical register accessed.

VALUE unsigned int

The new value written to the register.

CORE_REGS64

Changes of the core registers X0..X30, SP_ELn. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ID enum

The register number, 0-30 for X0-X30, >=32 for SP_ELn.

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

CORE_REGS64_READ

General purpose AArch64 register has been read X0..X30, SP_ELn. Fields:

ID enum

The register number, 0-30 for X0-X30, >=32 for SP_ELn.

VALUE unsigned int

The value read from the register.

CORE_STALLED_TRANSITION

Detect when a core is prevented from progressing PChannel states. Fields:

CURRENT_STATE enum

Current power mode of core.

REASON string

The reason a transition was stalled, and what must be done to unblock it.

TARGET_STATE enum

Desired power mode of core after transition.

CORE_STORES

Processor store accesses. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ACQREL enum

Is this an acquire/release.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DATA unsigned int

The data read or written.

ELEMENT_SIZE unsigned int

Width of each element.

ENDIANNESS enum

Byte ordering of this access.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

LOCK enum

Normal, exclusive or locked access.

MECID unsigned int

Memory Encryption Context ID, if applicable.

MECID2 unsigned int

Memory Encryption Context ID of the second page, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PAGE_SIZE unsigned int

Page size as $\log_2(\text{size in bytes})$.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK, 2=Exclusive Failed, 3=Unsupported.

SIZE unsigned int

Width of the access in bytes. Only required if DATA is not traced.

TRANS bool

Is this a translated access.

VADDR unsigned int

The virtual address of the access.

CP14_READ

System Coprocessor register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP14 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc1 unsigned int

opcode 1.

opc2 unsigned int
opcode 2.

CP14_WRITE

System Coprocessor register write. Fields:

CORE_NUM unsigned int
Core number in a multi processor.

CRm unsigned int
CRm.

CRn unsigned int
CRn.

NS bool
Secure or nonsecure banked register is accessed.

REG_NAME string
Name of the CP14 register accessed.

SECURITY_STATE enum
The current security state.

UNDEF bool
The register accessed is undefined.

UPDATED_VALUE unsigned int
Updated value of the register now it has been written.

VALUE unsigned int
The new value written.

opc1 unsigned int
opcode 1.

opc2 unsigned int
opcode 2.

CP15_READ

System Coprocessor register read. Fields:

CORE_NUM unsigned int
Core number in a multi processor.

CRm unsigned int
CRm.

CRn unsigned int
CRn.

NS bool
Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP15 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc1 unsigned int

opcode 1.

opc2 unsigned int

opcode 2.

CP15_READ64

System Coprocessor register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP15 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc unsigned int

opcode 1.

CP15_WRITE

System Control Coprocessor register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP15 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc1 unsigned int

opcode 1.

opc2 unsigned int

opcode 2.

CP15_WRITE64

System Coprocessor register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP15 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc unsigned int

opcode 1.

CPSR

CPSR change. (In AArch64 PSTATE represented in SPSR format). Fields:

CORE_NUM unsigned int

Core number in a multi processor.

OLD_VALUE unsigned int

The old CPSR value.

UNKNOWN unsigned int

Bits within the register that have unknown value.

VALUE unsigned int

The new CPSR value.

CRYPTO_SPEC

Every crypto instruction speculatively executed.

CorePowerStateChange

Core power-up state has changed. DISPLAY Core powered %{PU:(up|down)}. Fields:

PU bool

Whether the core powered up.

DATA_CACHE_SET_ALLOC_TAGS

Set allocation tags from DC GVA instruction. Fields:

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

NSDESC unsigned int

The security state of the access.

PADDR unsigned int

The physical address of the access.

PAS enum

The physical address space of the page.

RESPONSE enum

0=Aborted, 1=OK.

SIZE unsigned int

The size of memory in bytes.

TAG unsigned int

The value of the tag being set.

TAGGING_TYPE enum

Type of Memory tagging (Physical or Virtual).

TAG_VADDR unsigned int

The virtual address of the Virtual Tag if Virtual tagging is used; otherwise 0.

VADDR unsigned int

Virtual address used for setting allocation tags.

DATA_CACHE_ZERO

Zero and invalidate cache line from DC ZVA instruction. Fields:

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

NSDESC unsigned int

The security state of the access.

PADDR unsigned int

The physical address of the access.

PADDR_TAG unsigned int

If tag checked, the physical address tag for the access.

PAS enum

The physical address space of the page.

RESPONSE enum

0=Aborted, 1=OK.

SIZE unsigned int

The size of zero'd memory in bytes.

TAG_CHECKED bool

Is this access tag checked.

VADDR unsigned int

Virtual address of the zero'd cache line.

DEBUG_EVENT

Hardware debug support event. Fields:

EVENT enum

Description of event.

VALUE unsigned int

data value.

DEBUG_TARGET_ISET

Hardware debug target ISA. Fields:

TARGET_ISET enum

The target instruction set of the debug state.

DMI_ALLOCATE

DMI allocated. Fields:

ACCESS_DIRECTION enum

READ / WRITE.

PADDR unsigned int

The physical address of the DMI.

PTR unsigned int

DMI range pointer.

SIDE enum

DSIDE / ISIDE.

SIZE unsigned int

The size of DMI.

VADDR unsigned int

The virtual address of the DMI..

DMI_HIT

DMI hit, access used DMI (only fire for small DMI regions i.e. less than a full page). Fields:

ACCESS_DIRECTION enum

READ / WRITE.

PADDR unsigned int

The physical address of the DMI.

SIDE enum

DSIDE / ISIDE.

SIZE unsigned int

The size of the DMI.

VADDR unsigned int

The virtual address of the DMI..

DMI_REVOKE

DMI revoked. Fields:

ACCESS_DIRECTION enum

READ / WRITE.

PADDR unsigned int

The physical address of the DMI.

SIDE enum

DSIDE / ISIDE.

SIZE unsigned int

The size of DMI.

VADDR unsigned int

The virtual address of the DMI..

END_COMPILE

Compilation end. Fields:

END_OF_PAGE_COUNT unsigned int

Number of basic blocks exited due to page boundary since START_COMPILE.

FETCHFAIL_COUNT unsigned int

Number of basic blocks exited due to ifetch failure START_COMPILE.

INST_COUNT unsigned int

Number of instructions compiled since START_COMPILE.

NONSEQ_COUNT unsigned int

Number of basic blocks exited due to non-sequential instructions since START_COMPILE.

PAGE_STRADDLE_COUNT unsigned int

Number of basic blocks exited due to unaligned instructions crossing page since START_COMPILE.

ESB

Error Synchronization Barrier hit. Fields:

ASYNC bool

ESB is async.

IMPLICIT bool

ESB is implicit.

EXCEPTION

Exception taken. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ESR unsigned int

The value assigned to ESR when exception is taken.

ESR_EC unsigned int

The Value assigned to ESR.EC when exception is taken.

IS_PHYSICAL bool

Physical or Virtual exception.

IS_TRAP bool

Exception coming from a trap.

LR unsigned int

The value assigned to the link register.

PC unsigned int

The location where the exception occurred.

PREFERRED_RETURN unsigned int

The preferred return address for the exception.

TAKEN_LOCALLY bool

Whether exception is taken locally(same exception level).

TARGET_ISET enum

The instruction set of the exception handler code.

TARGET_PC unsigned int

The address the exception branches to.

VECTOR enum

The exception vector.

VECTOR_OFFSET unsigned int

The offset of an exception vector from the base address.

EXCEPTION_END

Every exception completed.

EXCEPTION_INFO

Trace human-readable reason of the current exception for data/instruction aborts, floating point exceptions and unknown exceptions. Fields:

DESCRIPTION string

Reason for current exception.

ESR_EC unsigned int

The value of ESR.EC when exception is taken.

EXCEPTION_RAISE

Every exception raised.

EXCEPTION_RETURN

Branches on leaving exception. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISSET enum

The instruction set of the branch instruction.

IS_COND bool

Indicates if this is a conditional branch.

IS_HINTED bool

Whether the conditional instruction is hinted.

IS_ISB bool

Whether the instruction is ISB.

IS_LINK bool

Indicates if this is a branch with link.

PC unsigned int

The address of the branch instruction.

TARGET_ISET enum

The instruction set after the branch.

TARGET_PC unsigned int

The address the instruction branches to.

EXCEPTION_RETURN_PREBRANCH

Exception return event that must occur before PC and processor state have updated.

EXCEPTION_START

Every exception started.

FIQ_TAKEN

FIQ taken exception.

FP_EXCEPTIONS_SIGNALLED

Trace which FP exceptions were signalled as a part of the FP operations performed by an instruction. Fields:

FP_EXCEPTIONS_SIGNALLED_MASK unsigned int

Indicates which FP exceptions were signalled by an instruction. This uses the same encoding as the cumulative floating-point exception bitfields in FPSR.

FP_STATE

Floating point state. Fields:

FPSR unsigned int

FPSR register value.

MASK unsigned int

The mask indicating updated cumulative bits.

FREQ_CHANGED

Core frequency changed. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The instruction count of this core.

NEW_FREQ signed int

The new frequency of this core (expressed in Hz).

OLD_FREQ signed int

The old frequency of this core (expressed in Hz).

HLT

HLT instruction occurred. Fields:

IMM16 unsigned int

The 16-bit immediate value encoded in HLT instruction.

INST

Every instruction executed. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DEBUG_STATE bool

The instruction is executed in debug state.

DISASS string

Disassembly of instruction.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISSET enum

The current instruction set.

ITSTATE unsigned int

The current ITSTATE.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

MODE enum

The mode the core is in.

NS unsigned int

The core's non-secure bit.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

OPCODE unsigned int

The opcode of the instruction.

PADDR unsigned int

The physical address of the instruction.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the instruction.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the page.

PC unsigned int

The address of the instruction.

SECURITY_STATE enum

The core's security state.

SIZE unsigned int

The size of the instruction in bytes.

INST_END

Every instruction completed.

INST_START

Every instruction started. Fields:

DEBUG_STATE bool

The instruction is executed in debug state.

ISSET enum

The current instruction set.

MODE enum

The mode the core is in.

NS enum

The current Secure State.

PC unsigned int

The address of the conditional instruction.

SECURITY_STATE enum

The current Security State.

INST_STRADDLE

Instruction straddles boundary. Fields:

DEBUG_STATE bool

The instruction is executed in debug state.

ISSET enum

The current instruction set.

MODE enum

The mode the core is in.

NS enum

The current Secure State.

PC unsigned int

The address of the conditional instruction.

SECURITY_STATE enum

The current Security State.

IRQ_TAKEN

IRQ taken exception.

ISB

ISB instruction occurred. Fields:

PC unsigned int

The address of the ISB instruction.

LOAD_MULTIPLE_REGS_LIST

List of destination registers (not source) of a multiple registers load instruction. Fields:

REG string

Register to add into the list.

LOCAL_MONITOR

Local monitor activity. Fields:

PADDR unsigned int

Local Monitor Address.

State enum

State of the monitor (Open/Exclusive).

MAPPING_CREATE

A memory mapping has been created. This is an implementation detail of the model and so may change without notice. Fields:

ASID unsigned int

The ASID used during translation, or 0 if no ASID was used.

IS_VALID bool

Whether the mapping is valid or not.

PADDR unsigned int

Base physical address of region, or 0 if not known.

SIDE enum

I-side or D-side.

SIZE unsigned int

Size of the page, or 0 if not known.

VADDR unsigned int

Base virtual address of region.

VMID unsigned int

The VMID used during translation, or 0 if no VMID was used.

MAPPING_DESTROY

A memory mapping has been destroyed. This is an implementation detail of the model and so may change without notice. Fields:

ASID unsigned int

The ASID used during translation.

CAUSE enum

The cause of invalidation.

PADDR unsigned int

Base physical address of page.

SIDE enum

I-side or D-side.

SIZE unsigned int

Size of the page.

VADDR unsigned int

Base virtual address of page.

VMID unsigned int

The VMID used during translation.

MEMMAP_DEBUG_READ

Memory mapped reads to the debug registers. Fields:

ADDR unsigned int

address.

CORE_NUM unsigned int

Core number in a multi processor.

EXT bool

Whether access is from an external device (such as the DAP).

NS enum

Secure state of the access.

REG_NAME string

Name of the debug register accessed.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

MEMMAP_DEBUG_WRITE

Memory mapped writes to the debug registers. Fields:

ADDR unsigned int

address.

CORE_NUM unsigned int

Core number in a multi processor.

EXT bool

Whether access is from an external device (such as the DAP).

NS enum

Secure state of the access.

REG_NAME string

Name of the debug register accessed.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The new value written.

MMU_D128_TTB_READ

This event is triggered by reads caused by a D128 translation table walk. Fields:

ABORTED bool

Was the walk successful.

DATA_HIGH unsigned int

The data upper bits read.

DATA_LOW unsigned int

The data lower bits read.

IPA unsigned int

For stage 1, the IPA of the read.

LEVEL unsigned int

Translation table level.

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

OUTADDR unsigned int

The output physical address.

PADDR unsigned int

The physical address of the read.

PASmem enum

Physical Address Space of the accessed memory.

PASreq enum

Physical Address Space of the requested lookup.

REGIME_EL enum

Entry matches in this translation regime.

REQUESTER enum

The requester of this table walk.

SIDE enum

Inst / Data.

STAGE unsigned int

Translation stage.

MMU_D128_TTB_WRITE

This event is triggered by writes caused by a D128 translation table walk. Fields:

ABORTED bool

Was the walk successful.

DATA_HIGH unsigned int

The data upper bits written.

DATA_LOW unsigned int

The data lower bits written.

IPA unsigned int

For stage 1, the IPA of the write.

IS_HACDBS_ACCESS bool

If the access was due to HACDBS_CLEAN.

LEVEL unsigned int

Translation table level.

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

OUTADDR unsigned int

The output physical address.

PADDR unsigned int

The physical address of the write.

PASmem enum

Physical Address Space of the accessed memory.

PASreq enum

Physical Address Space of the requested lookup.

REGIME_EL enum

Entry matches in this translation regime.

SIDE enum

Inst / Data.

STAGE unsigned int

Translation stage.

MMU_TRANS

Address translation information. Fields:

ASID unsigned int

Address space identifier.

CORE_NUM unsigned int

Core number in a multi processor.

Hyp bool

Entry matches in Hyp state only.

INNERCACHE_RA bool

Is the inner cache allocate on read.

INNERCACHE_TRANSIENT bool

Is the inner write-through transient.

INNERCACHE_TYPE enum

Inner Caching scheme (NC/MB/WA).

INNERCACHE_WA bool

Is the inner cache allocate on write.

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MEMTYPE enum

Memory type.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC enum

Is secure side supposed to access secure or nonsecure memory.

OUTERCACHE_RA bool

Is the outer cache allocate on read.

OUTERCACHE_TRANSIENT bool

Is the outer write-through transient.

OUTERCACHE_TYPE enum

Outer Caching scheme (NC/MB/WA).

OUTERCACHE_WA bool

Is the outer cache allocate on write.

PADDR unsigned int

Physical address of the access.

PAGESIZE unsigned int

Page size as log2(size).

PAS enum

PAS supposed to be accessed is secure or nonsecure memory or Root/Realm if RME is implemented.

SH enum

Shareability.

SIDE enum

Inst / Data.

STAGE1_PERM unsigned int

Stage 1 permissions, [5:3] Privileged access (XWR) [2:0] User access (XWR).

STAGE2_PERM unsigned int

Stage 2 permission mask [5:3] Privileged access (XWR) [2:0] User access (XWR).

VADDR unsigned int

Virtual address of the access.

VMID unsigned int

Virtual machine identifier.

XS bool

XS Attribute.

nG enum

Flag indicating whether ASID will be matched.

MMU_TTB_READ

This event is triggered by reads caused by a translation table walk. Fields:

ABORTED bool

Was the walk successful.

DATA unsigned int

The data read.

DESC_KIND enum

The encoding scheme of the descriptors used for this walk.

IPA unsigned int

For stage 1, the IPA of the read.

LEVEL unsigned int

Translation table level.

LPAE bool

Is this for an LPAE translation (*DEPRECATED*, use DESC_KIND).

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSmem bool

Non secure memory access.

NSreq bool

Non secure request.

OUTADDR unsigned int

The output physical address.

PADDR unsigned int

The physical address of the read.

PASmem enum

Physical Address Space of the accessed memory.

PASreq enum

Physical Address Space of the requested lookup.

REGIME_EL enum

Entry matches in this translation regime.

REQUESTER enum

The requester of this table walk.

SIDE enum

Inst / Data.

STAGE unsigned int

Translation stage.

MMU_TTB_WRITE

This event is triggered by writes caused by a translation table walk. Fields:

ABORTED bool

Was the walk successful.

DATA unsigned int

The data written.

DESC_KIND enum

The encoding scheme of the descriptors used for this walk.

IPA unsigned int

For stage 1, the IPA of the write.

IS_HACDBS_ACCESS bool

If the access was due to HACDBS_CLEAN.

LEVEL unsigned int

Translation table level.

LPAE bool

Is this for an LPAE translation (*DEPRECATED*, use DESC_KIND).

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSmem bool

Non secure memory access.

NSreq bool

Non secure request.

OUTADDR unsigned int

The output physical address.

PADDR unsigned int

The physical address of the write.

PASmem enum

Physical Address Space of the accessed memory.

PASreq enum

Physical Address Space of the requested lookup.

REGIME_EL enum

Entry matches in this translation regime.

SIDE enum

Inst / Data.

STAGE unsigned int

Translation stage.

MODE_CHANGE

Mode change. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

MODE enum

The new mode.

NON_SECURE enum

The core's new non-secure bit.

OLD_MODE enum

The old mode.

SECURITY_STATE enum

The new security state.

NOT_TAG_CHECKED_INST

MTE Tag not checked for current instruction.

PERIODIC

Called for every quantum. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The instruction count of this CPU.

PC unsigned int

The address of the next instruction to be executed on this CPU.

PMU_COUNTER_OVERFLOW

PMU counter overflow. Fields:

COUNTER_GROUP enum

Counter group.

COUNTER_RANGE enum

Counter range.

EVENT_ID unsigned int

Event Identifier, field is valid only for Event counter group.

INDEX unsigned int

Counter index (as selected by PMSELR), field is valid only for Event/Cycle counter groups.

INTERRUPT bool

Is interrupt enabled on overflow for this counter.

POTINDEX_UPDATE

Value of POT index for current pc changed. Fields:

NEW_POTINDEX unsigned int

New POTIndex.

OLD_POTINDEX unsigned int

Old POTIndex.

PREFETCH_MEMORY64

Prefetch from PRFM or PRFUM instructions. Fields:

PRFOP enum

Prefetch hint.

VADDR unsigned int

Virtual address of the location that should be prefetched.

PRELOAD64_REQUEST

Preload requests from PRFM or PRFUM instructions. Fields:

PRFOP enum

Prefetch hint.

VADDR unsigned int

Virtual address of the location that should be prefetched.

PRELOAD_DATA

Data preload from PLD instruction. Fields:

VADDR unsigned int

Virtual address of the data that should be preloaded.

PRELOAD_INST

Instruction preload from PLI instruction. Fields:

VADDR unsigned int

Virtual address of the instruction that should be preloaded.

PRE_CORE_LOAD

Trace just before a core load. Fields:

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

PRE_CORE_STORE

Trace just before a core store. Fields:

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

PRE_TTB_READ

This event is triggered before reads caused by a translation table walk. Fields:

ADDR unsigned int

The physical address of the read.

PSTATE

PSTATE fields update. Fields:

ALLINT unsigned int

All interrupt mask bit.

BTYPE unsigned int

Branch target identification bit.

CORE_NUM unsigned int

Core number in a multi processor.

DAIF unsigned int

Exception mask bits.

DIT unsigned int

Data Independent Timing bit.

EL unsigned int

Current Exception level field.

EXLOCK unsigned int

Exception return state lock bit.

IL unsigned int

Illegal Execution state bit.

NZCV unsigned int

Condition flags.

PACM unsigned int

PACM bit.

PAN unsigned int

Privileged Access Never state bit.

PM unsigned int

PMU exception mask bit.

PPEND unsigned int

PMU exception pending bit.

SM unsigned int

StreamingMode status bit.

SP unsigned int

Stack pointer register selection bit.

SS unsigned int

Software Step bit.

SSBS unsigned int

Speculative Store Bypass Safe bit.

TCO unsigned int

Tag Check Override bit.

TOUCHED unsigned int

Mask of the fields that have been touched since this trace has last been emitted. Note that touched does not necessarily mean the value changed. The index of the field is the bit index in the mask plus the index of this field plus one.

UAO unsigned int

User Access Override bit.

UINJ unsigned int

Undefined Instruction exceptions bit.

ZA unsigned int

ZA storage and ZT0 register enable bit.

nRW unsigned int

Current Execution state bit.

PSTATE_PACTIVE_OPMODE_TRANSITION

Trace active and Opmode mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Operating mode after the transition.

OLD_STATE enum

Operating mode before the transition.

PSTATE_PACTIVE_OPMODE_TRANSITION_DENIED

Trace the denied active and Opmode mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Operating mode after the transition.

OLD_STATE enum

Operating mode before the transition.

REASON string

Reason for the transition being denied.

PSTATE_PACTIVE_POWER_TRANSITION

Trace active and power mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Power mode after the transition.

OLD_STATE enum

Power mode before the transition.

PSTATE_PACTIVE_POWER_TRANSITION_DENIED

Trace the denied active and power mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Power mode after the transition.

OLD_STATE enum

Power mode before the transition.

REASON string

Reason for the transition being denied.

PSTATE_PACTIVE_THREAD_OPMODE_TRANSITION

Trace pactive and Opmode mode transitions for threads on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Operating mode after the transition.

OLD_STATE enum

Operating mode before the transition.

QUANTUM_START

Called at the start of every quantum.

RANGE_PRELOAD_REQUEST

Preload requests from RPRFM. Fields:

COUNT unsigned int

Number of blocks of data to access.

LENGTH unsigned int

Number of contiguous bytes to be accessed in each block.

OP unsigned int

Range prefetch operation .

REUSE unsigned int

Maximum number of bytes to be accessed before next RPRFM .

STRIDE signed int

Stride to advance after accessing LENGTH bytes.

VADDR unsigned int

Base virtual address of range prefetch request.

RUN_STATE

Run state transition. Fields:

INST_COUNT unsigned int

Ticks count at point of transition.

NEW enum

New run state.

OLD enum

Old run state.

SBP_SAMPLE_COLLISION

DISPLAY Statistical profiling samples collided .

SBP_SAMPLE_DISCARDED

DISPLAY Operation discarded by statistical profiling unit when configured in discard mode.
Fields:

BUFFER_POINTER unsigned int

Value of buffer pointer.

DATA unsigned int

Data discarded.

SIZE unsigned int

Size of sample, in bytes.

SBP_SAMPLE_END

DISPLAY Statistical profiling of an operation ended.

SBP_SAMPLE_POP

DISPLAY Operation which might be sampled by statistical profiling.

SBP_SAMPLE_RECORDED

DISPLAY Operation sampled by statistical profiling has been written to the buffer at %{BUFFER_ADDRESS} with size %{SIZE} and data %{DATA}. Fields:

BUFFER_ADDRESS unsigned int

Virtual Address of start of buffer write.

DATA unsigned int

Data written to the buffer.

SIZE unsigned int

Size of sample written to buffer, in bytes.

SBP_SAMPLE_START

DISPLAY Statistical profiling of an operation started.

SBP_SAMPLE_TAKEN

DISPLAY Operation sampled by statistical profiling after filtering.

SEMIHOSTING_CALL

Call of a semihost function occurred. Fields:

PC unsigned int

The program counter after the semihosting call.

SEMIHOSTING_PRECALL

About to call semihost. Fields:

EL2 bool

Use EL2 translation regime.

NS bool

Is Non-Secure.

REG_WIDTH unsigned int

The current register width in bytes.

SIGNAL

External signal state change. Fields:

SIGNAL *enum*

Signal that changed.

STATE *bool*

Signal asserted state.

SOFTWARE_STEP

Return the Debug Software Step state. Fields:

STATE *enum*

Software Step state.

SPSR

SPSR change. Fields:

CORE_NUM *unsigned int*

Core number in a multi processor.

MODE *enum*

Which of the banked SPSR registers is written.

OLD_VALUE *unsigned int*

The old SPSR value.

VALUE *unsigned int*

The new SPSR value.

START_COMPILE

Compilation started. Fields:

VADDR *unsigned int*

Instruction where compilation begins.

STG_MULTIPLE_MICRO_OPS_ADDRESS_RANGE

Address range for MTE stores with multiple uops. Fields:

ADDR_END *unsigned int*

Final address.

ADDR_START *unsigned int*

Starting address.

STORE_MULTIPLE_REGS_ADDRESS_RANGE

Address range for stores with multiple registers. Fields:

ADDR_END *unsigned int*

Final address.

ADDR_START *unsigned int*

Starting address.

SUBOPTIMAL_LDST_RETIRED

suboptimal load/store. Fields:

ACCESS_SIZE unsigned int

Log2 of the access width in bytes used for alignment checking (1: half-word, 2: word ...).

OPTIMAL_ALIGN_SELECTOR unsigned int

Selector to decide the optimal alignment size (1: access width, 2: 4-bytes, 3: 8-bytes, ..., 12: 4-kbytes).

VADDR unsigned int

The virtual address of the access.

SYNC

Called for every synchronization, that is every quantum break. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The instruction count of this CPU.

LOCAL_QUANTUM unsigned int

The local quantum of this CPU.

LOCAL_TIME unsigned int

The local time of this CPU.

SYNC_REASON enum

Reason for sync.

SYSCALL

System call instruction executed. Fields:

IMM unsigned int

Immediate value of the system call instruction.

TYPE enum

System call type.

VADDR unsigned int

Instruction that caused the system call.

SYSREG_READ128

System Coprocessor 128 bit register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

REGNUM enum

Internal register number.

REGNUM_PHYS enum

Internal register number physically accessed (e.g. in presense of VHE redirection).

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_READ64

System Coprocessor register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REGNUM enum

Internal register number.

REGNUM_OPERAND enum

Register number corresponding to the instruction operand.

REGNUM_PHYS enum

Internal register number physically accessed (e.g. in presense of VHE redirection).

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_UPDATE128

Triggers when the system updates a register. Fields:

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG enum

Register number.

UNKNOWN unsigned int

Bits of the register which became unknown.

VALUE unsigned int

Value written to the register.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_UPDATE32

Triggers when the system updates a register. Fields:

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG enum

Register number.

UNKNOWN unsigned int

Bits of the register which became unknown.

VALUE unsigned int

Value written to the register.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_UPDATE64

Triggers when the system updates a register. Fields:

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG enum

Register number.

UNKNOWN unsigned int

Bits of the register which became unknown.

VALUE unsigned int

Value written to the register.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_WRITE128

System Coprocessor 128 bit register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

REGNUM enum

Internal register number.

REGNUM_PHYS enum

Internal register number physically accessed (e.g. in presense of VHE redirection).

UNDEF bool

The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_WRITE64

System Coprocessor register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REGNUM enum

Internal register number.

REGNUM_OPERAND enum

Register number corresponding to the instruction operand.

REGNUM_PHYS enum

Internal register number physically accessed (e.g. in presense of VHE redirection).

UNDEF bool

The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

UNALIGNED_LDST_RETIRE

Processor unaligned load/store. Fields:

ACCESS_SIZE unsigned int

Log2 of the access width in bytes used for alignment checking (1=>half-word, 2->word ...).

NUMBER_OF_BEATS unsigned int

Number of accesses (beats) in this burst (Total size in bytes is ACCESS_SIZE * NUMBER_OF_BEATS).

VADDR unsigned int

The virtual address of the access.

VFP_D_REGS

VFP/NEON D 64 bit register write. Fields:

ALIASING enum

Type of register aliasing used.

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a write to the corresponding bit in the register. For example, 0x1 would indicate a write to VALUE[0].

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

VFP_D_REGS_READ

VFP/NEON D 64 bit register read. Fields:

ID unsigned int

The register number.

VALUE unsigned int

The current value of the register.

VFP_Q_REGS

VFP/NEON Q 128 bit register write. Fields:

ALIASING enum

Type of register aliasing used.

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a write to the corresponding byte in the register. For example, 0x1 would indicate a write to VALUE[7:0].

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

VFP_Q_REGS_READ

VFP/NEON Q 128 bit register read. Fields:

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a read to the corresponding byte in the register. For example, 0x1 would indicate a read to VALUE[7:0].

VALUE unsigned int

The current value of the register.

VFP_SYS_REGS

Writes to the VFP/NEON units system registers. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ID enum

Which VFP system register is written.

MASK unsigned int

The mask indicating updated cumulative bits.

OLD_VALUE unsigned int

The register's old value overwritten.

VALUE unsigned int

The new value written to the VFP system register.

VFP_S_REGS

VFP/NEON S 32 bit register write. Fields:

ALIASING enum

Type of register aliasing used.

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a write to the corresponding bit in the register. For example, 0x1 would indicate a write to VALUE[0].

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

VFP_S_REGS_READ

VFP/NEON S 32 bit register read. Fields:

ID unsigned int

The register number.

VALUE unsigned int

The current value of the register.

VIRTUAL_TAG_ALLOCATED

Virtual tag information when allocated. Fields:

TAG_VADDR unsigned int

Virtual tag virtual address.

VADDR unsigned int

Data Virtual address.

VIRTUAL_TAG unsigned int

Virtual tag value.

WAYPOINT

Signals end of basic block execution. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ISET enum

Origin instruction set.

IS_COND bool

Indicates if this is a conditional waypoint.

PC unsigned int

Origin address (or 0 if unavailable).

TAKEN bool

Indicates if this waypoint was taken.

TARGET unsigned int

Destination address.

TARGET_ISET enum

Destination instruction set.

WFE_END

WFE ended. Fields:

INST_COUNT unsigned int

Ticks count when leaving WFE.

WFE_EVENT_REGISTER

WFE event register status: set/clear, reason. Fields:

INST_COUNT unsigned int

Ticks count.

REASON enum

Reason for set/clear. Only REASON==0 (Cleared by WFE) clears the bit, all other reasons set it.

WFE_IGNORED

WFE ignored. Fields:

EVENT bool

This WFE was ignored because the event register was set.

INST_COUNT unsigned int

Ticks count when ignoring WFE.

TRAPPED bool

This WFE was trapped.

WFE_START

WFE entered. Fields:

INST_COUNT unsigned int

Ticks count when entering WFE.

WFI_END

WFI ended. Fields:

INST_COUNT unsigned int

Ticks count when leaving WFI.

WFI_IGNORED

WFI ignored. Fields:

DISABLED bool

This WFI was ignored because WFI is disabled.

INST_COUNT unsigned int

Ticks count when ignoring WFI.

REASON enum

specifies reason why WFI trace was ignored.

TRAPPED bool

This WFI was trapped.

WFI_START

WFI entered. Fields:

INST_COUNT unsigned int

Ticks count when entering WFI.

WFI_WAKEUP

WFI wakeup. Fields:

INST_COUNT unsigned int

Ticks count when WFI wakeup occurred.

REASON enum

Reason for wakeup.

2.36 ARM_Cortex-A78AE

This section describes the trace sources.

AA64_ASE_SVE_REGS

Modification of AArch64 Advanced SIMD or SVE Vector Registers (via SIMD or Scalar operations), SVE Predicate Registers and ZTO. Fields:

ID enum

The register identifier.

MASK unsigned int

Bitmask of the register to signal the modified bits in the VALUE field.

SM bool

Whether the PE is in Streaming Mode.

VALUE unsigned int

Value of AArch64 vector register.

ASYNC_MEMORY_FAULT

Context ID Register write. Fields:

FAULT unsigned int

Fault status present in ESR.ISS encoding for Serror in ARCH64 and present in DFSR encoding for external abort format in ARCH32.

PADDR unsigned int

Physical Address (or 0 if unavailable).

VADDR signed int

Virtual Address (or 0 if unavailable).

ATOMIC_END_ACCESS

Bus trace access for atomics. Fields:

ACCESS_FAIL bool

Memory access failed.

ADDR unsigned int

The virtual address of the access.

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

COMPARE_VALUE unsigned int

Compare value for CAS.

LOAD_VALUE unsigned int

Loaded values.

NSDESC unsigned int

The security state of the access.

OPERAND_VALUE unsigned int

Operation's operand.

OPERATION enum

Operation type.

PADDR **unsigned int**

The physical address of the access.

PRIV **bool**

Is this a privileged access?.

ATOMIC_START_ACCESS

Bus trace access for atomics. Fields:

ADDR **unsigned int**

The virtual address of the access.

ATTR **unsigned int**

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

COMPARE_VALUE **unsigned int**

Compare value for CAS.

NSDESC **unsigned int**

The security state of the access.

OPERAND_VALUE **unsigned int**

Operation's operand.

OPERATION **enum**

Operation type.

PADDR **unsigned int**

The physical address of the access.

PRIV **bool**

Is this a privileged access?.

ArchMsg.Error.exit_code

DISPLAY exit code: %{EXITCODE} component kind: %{KIND} reason: %{REASON}. Fields:

EXITCODE **signed int**

Exit code to be returned.

KIND **string**

Component kind that invoked the exit code trace.

REASON **string**

Reason why the exit code trace was invoked (optional).

ArchMsg.Warning.IgnoredWarmResetRequest

Event fires when software attempts to request warm reset through write to deprecated DBGPRCR.CWRR.

ArchMsg.Warning.RAS_ErrselrUnpredictableState

DISPLAY ERRSELR is in unpredictable state due to value greater than number_of_error_records written to it.

ArchMsg.Warning.RAS_UnpredictableStateWrite

DISPLAY %{REG_NAME} write ignored due to unpredictable ERRSELR state. Fields:

REG_NAME string

Name of register being written.

ArchMsg.Warning.ReportedRevisionRegisterRead

DISPLAY Reported revision and patch do not match that of the cpu. Fields:

ORIGINAL_VALUE unsigned int

The value the register would have based on the modelled revision.

REGISTER string

The name of the register that has ben read.

REPORTED_VALUE unsigned int

The value the register has been overridden to.

ArchMsg.Warning.Unpredictably Indexed PM Event Register

TAGS **UNPREDICTABLE** DISPLAY Event %{IsDirect:(Selector value|Register)} %{SEL} is >= %{N}. Fields:

IsDirect unsigned int

Direct Access rather than indirect.

N unsigned int

Number of Accessible Registers.

SEL unsigned int

Selector.

ArchMsg.Warning.cache_contents_unknown

A cache has been activated without being invalidated and may contain **UNKNOWN** content. DISPLAY %{SIDE:(D|I)}-Cache was enabled in %{REGIME:(SECURE|NON_SECURE|HYP)} regime but was not invalidated since power-on: cache lines could contain **UNKNOWN** content %{SIDE:(Note: D-side caches must invalidate SECURE lines even if the cache is only used by NON_SECURE code in order to ensure that unexpected dirty lines tagged SECURE are not naturally evicted)}. Fields:

REGIME unsigned int

Current translation regime.

SIDE unsigned int

0 d-side cache or unified cache, 1 i-side.

ArchMsg.Warning.dcimvac_matches_watchpoint

TAGS **UNPREDICTABLE** DISPLAY Watchpoints matching an AArch32 DCIMVA are implementation defined.

ArchMsg.Warning.decode_fieldmismatch

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with different fields %{FIELD1} and %{FIELD2}. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_fracbitsnegative

TAGS **UNPREDICTABLE** DISPLAY VCVT Instruction is unpredictable with a negative number of fraction bits.

ArchMsg.Warning.decode_initblock

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable in an IT block.

ArchMsg.Warning.decode_initblocknotlast

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable in an IT block when not the last.

ArchMsg.Warning.decode_invalid_condition

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable when the condition is not AL.

ArchMsg.Warning.decode_invalidfieldcombination

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with the values in %{FIELD1} and %{FIELD2}. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_invalidvalue

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with the value in %{FIELD}. Fields:

FIELD enum

Field name.

ArchMsg.Warning.decode_mem_hint_unallocated

TAGS **UNPREDICTABLE** DISPLAY There is no memory hint allocated to this bit pattern.

ArchMsg.Warning.decode_registermatch

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with %{FIELD1} and %{FIELD2} as the same register. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_registermismatch

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with %{FIELD1} and %{FIELD2} as different registers. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_registeroutofrange

TAGS **UNPREDICTABLE** DISPLAY Floating-point instruction is unpredictable indexing beyond the end of the register bank.

ArchMsg.Warning.decode_sbzsbo

TAGS **UNPREDICTABLE** DISPLAY Reserved bits in the instruction are not canonical.

ArchMsg.Warning.decode_transactiontoobig

TAGS **UNPREDICTABLE** DISPLAY Attempt to perform a memory transaction of more than 128 bytes.

ArchMsg.Warning.decode_unpred_other

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable.

ArchMsg.Warning.decode_unpreduseofpc

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable when %{FIELD} is PC. Fields:

FIELD enum

Field name.

ArchMsg.Warning.decode_unpreduseofr13

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable when %{FIELD} is R13. Fields:

FIELD enum

Field name.

ArchMsg.Warning.decode_unpreduseofr13inreglist

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with R13 in register-list.

ArchMsg.Warning.decode_useofd16orhigher

TAGS **UNPREDICTABLE** DISPLAY Attempt to use register D16 or higher where it is not allowed.

ArchMsg.Warning.decode_writebackandbaseinlist

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with write back when the base register is in the transfer list.

ArchMsg.Warning.decode_zeroregistersinlist

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable when no registers are to be transferred.

ArchMsg.Warning.recursive_branch

An instruction is performing a branch that targets the same instruction. If this was intended then a WFI might be more effective. DISPLAY recursive branch detected at PC=%{PC}.

PRIMARY KEY PC. Fields:

PC unsigned int

Address of instruction causing the branch.

ArchMsg.Warning.recursive_exception

An instruction has generated an exception that targets the same instruction. DISPLAY recursive %{TYPE} exception detected at PC=%{PC} (CPSR=%{CPSR}). PRIMARY KEY PC. Fields:

CPSR unsigned int

Processor state.

PC unsigned int

Address of instruction causing the exception.

TYPE enum

Type of exception.

ArchMsg.Warning.reentrant_vector_catch

TAGS **UNPREDICTABLE** DISPLAY Vector catch debug event on Prefetch or Data abort vector %{ADDR}. Fields:

ADDR unsigned int

vector address.

ArchMsg.Warning.reserved_it_state

TAGS **UNPREDICTABLE** DISPLAY Execution with reserved IT state %{ITSTATE}. Fields:

ITSTATE unsigned int

reserved IT state.

ArchMsg.Warning.tlb_contents_unknown

Translation has been enabled without invalidating the TLB. DISPLAY Translation was enabled in the %{PAS:(Secure|Non-secure|Root|Realm)} EL%{EL:d} translation regime (vmid=%{VMID:x}) without invalidating %{INVALIDITY:(|d-side |i-side |)}entries since power-on: %{INVALIDITY:(|D||I)}TLB could contain **UNKNOWN** content. Fields:

EL unsigned int

Exception level of translation regime.

INVALIDITY unsigned int

Which TLB may be invalid (if TLBs are separate).

NON_SECURE bool

Whether the translation regime is non-secure.

PAS unsigned int

Physical address space of translation regime.

VMID unsigned int

Current VMID, if applicable.

ArchMsg.Warning.unknown_DLR_DSPSR

Exiting debug state with unknown DLR or DSPSR.

ArchMsg.Warning.unknown_ELR_SPSR

Returning from debug exception with unknown ELR or SPSR.

ArchMsg.Warning.unpred_FAR_on_LRCPC3_non_atomic_mem_access

TAGS **UNPREDICTABLE** Unpredictable FAR/FSTATUS on LRCPC3 pair instructions resulting in two separate memory accesses
DISPLAY Unpredictable FAR on LRCPC3 non atomic memory access at %{ADDR1} with FSTATUS %{FSTATUS1} or %{ADDR2} with FSTATUS %{FSTATUS2}. Fields:

ADDR1 unsigned int

Possible FAR address 1.

ADDR2 unsigned int

Possible FAR address 2.

FSTATUS1 unsigned int

Possible fault status value 1.

FSTATUS2 unsigned int

Possible fault status value 2.

ArchMsg.Warning.unpred_async_tag_check_fail_on_sve_speculative_load_do_update_ffr

Unpredictable set to FALSE of the FFR element associated to SVE speculative load with an async tag check fail TAGS **UNPREDICTABLE**.

ArchMsg.Warning.unpredictable_a32_breakpoint

TAGS **UNPREDICTABLE** DISPLAY Unpredictable breakpoint on a A32 instruction at %{ADDR1} Offset %{OFFSET} BAS %{BAS} for a %{IS_ADDRESS_MISMATCH_BREAKPOINT:(Mismatch|Match)} Breakpoint number %{N:d} Breakpoint address %{ADDR2}. Fields:

ADDR1 unsigned int

Address.

ADDR2 unsigned int

Breakpoint address.

BAS unsigned int

BAS field.

IS_ADDRESS_MISMATCH_BREAKPOINT bool

Is a Mismatch breakpoint, not a Match breakpoint.

N unsigned int

Breakpoint number.

OFFSET unsigned int

Always has a value of zero.

ArchMsg.Warning.unpredictable_banked_register_access

TAGS **UNPREDICTABLE** DISPLAY Access to banked register (R=%{R:d} SYSm=%{SYSM:2x}) is unpredictable. Fields:

R unsigned int

R bit.

SYSM unsigned int

SYSm.

ArchMsg.Warning.unpredictable_ccfail_undef

TAGS **UNPREDICTABLE** DISPLAY An instruction %{OPCODE} at VA=%{VADDR} failed its condition codes check but would otherwise have caused a non-data-dependent trap or undefined exception. Behaviour is implementation-defined choice of exception or nop. Fields:

OPCODE unsigned int

opcode of the instruction.

VADDR unsigned int

Virtual address of the instruction being executed.

ArchMsg.Warning.unpredictable_ici_bits

TAGS **UNPREDICTABLE** DISPLAY ICI bits do not apply or are architecturally unpredictable for instruction.

ArchMsg.Warning.unpredictable_t32_breakpoint

TAGS **UNPREDICTABLE** DISPLAY Unpredictable breakpoint on a T32 instruction at %{ADDR1} Offset %{OFFSET} BAS %{BAS} for a %{IS_ADDRESS_MISMATCH_BREAKPOINT:(Mismatch|Match)} Breakpoint number %{N:d} Breakpoint address %{ADDR2}. Fields:

ADDR1 unsigned int

Address.

ADDR2 unsigned int

Breakpoint address.

BAS unsigned int

BAS field.

IS_ADDRESS_MISMATCH_BREAKPOINT bool

Is a Mismatch breakpoint, not a Match breakpoint.

N unsigned int

Breakpoint number.

OFFSET unsigned int

Indicates which halfword of a 32-bit Thumb instruction the breakpoint is set on. 0 for the first halfword or if the instruction is 16 bits wide, or 1 for the second halfword.

ArchMsg.Warning.unpredictable_t32_vector_catch

TAGS **UNPREDICTABLE** DISPLAY Unpredictable Vector catch exception on a T32 instruction at %{ADDR} Offset %{OFFSET}. Fields:

ADDR unsigned int

Address.

OFFSET unsigned int

Indicates which halfword of a 32-bit Thumb instruction the vector catch exception is set on. 0 for the first halfword or if the instruction is 16 bits wide, or 1 for the second halfword.

ArchMsg.Warning.unpredictable_watchpoint_far

TAGS **UNPREDICTABLE** DISPLAY The value for FAR/EDWAR for hitting this watchpoint may be anywhere in the range %{LOWER_BOUND} to %{UPPER_BOUND}. Fields:

LOWER_BOUND unsigned int

the lowest address accessed by the instruction that triggered the watchpoint.

UPPER_BOUND unsigned int

the highest watchpointed address accessed by that instruction.

ArchMsg.Warning.unpredictable_watchpoint_far_isscmp

DISPLAY The value for FAR/EDWAR for hitting this watchpoint may be anywhere in the range %{LOWER_BOUND} to %{UPPER_BOUND}. Fields:

LOWER_BOUND unsigned int

the lowest address accessed by the instruction that triggered the watchpoint.

UPPER_BOUND unsigned int

the highest watchpointed address accessed by that instruction.

ArchMsg.Warning.warning_AdvSIMDExpandImmUnexpectedZero

TAGS **UNPREDICTABLE** DISPLAY AdvSIMDExpandImm may treat this immediate value as **UNPREDICTABLE**.

ArchMsg.Warning.warning_ConditionalSMC

TAGS **UNPREDICTABLE** DISPLAY SMC instruction has **UNPREDICTABLE** effects when conditional and when combined with traps.

ArchMsg.Warning.warning_TLBID_out_of_range

TLBID out of range in TLBI according to the configuration of NIS, NVIS, NOS and NVOS. Fields:

NIS unsigned int

NIS.

NOS unsigned int

NOS.

NVIS unsigned int

NVIS.

NVOS unsigned int

NVOS.

TLBID unsigned int

The TLBID value used in the TLBI instruction.

ArchMsg.Warning.warning_TTL_or_TG_mismatches_current_translation_granule

TTL or TG mismatches current translation granule in TLBI VA/RVA/IPA/RIPA. Fields:

CURRENT_TG unsigned int

Current TG as indicated by TCR_ELx or VTCR_EL2.

TLBI_TG unsigned int

TG (or TTL[3:2]) used in TLBI instruction.

ArchMsg.Warning.warning_access_crosses_page_boundary

TAGS **UNPREDICTABLE** DISPLAY Access crosses page boundary.

ArchMsg.Warning.warning_access_crosses_page_boundary_has_fault

TAGS **UNPREDICTABLE** DISPLAY Access that crosses page boundary has fault(s).

ArchMsg.Warning.warning_access_crosses_page_boundary_has_fault_on_both_pages

TAGS **UNPREDICTABLE** DISPLAY Access that crosses page boundary has a fault on each page.
Fields:

ADDR1 unsigned int

FAR address on first page.

ADDR2 unsigned int

FAR address on second page.

FSTATUS1 unsigned int

Fault status on first page.

FSTATUS2 unsigned int

Fault status on second page.

ArchMsg.Warning.warning_access_crosses_page_boundary_spanning_different_memory

TAGS **UNPREDICTABLE** DISPLAY Access at address %{ADDR} crosses page from MEM_TYPE %{MEMTYPE_PAGE1} to %{MEMTYPE_PAGE2}. Fields:

ADDR unsigned int

address of access crossing page.

MEMTYPE_PAGE1 signed int

First Page's memory type (0-Normal, 1-Non-Normal).

MEMTYPE_PAGE2 signed int

Second Page's memory type (0-Normal, 1-Non-Normal).

ArchMsg.Warning.warning_access_wraps_around_memory

TAGS **UNPREDICTABLE** DISPLAY Access wraps around memory in %{WIDTH:d}bit mode.
Fields:

WIDTH unsigned int

Address width.

ArchMsg.Warning.warning_bcr_linking_status

TAGS **UNPREDICTABLE** DISPLAY DBGBCR.{BT2,BT} combination gives unpredictable behaviour.

ArchMsg.Warning.warning_bcr_mask_reserved

TAGS **UNPREDICTABLE** DISPLAY DBGBCR_ELn.MASK is set to a reserved value.

ArchMsg.Warning.warning_bvr_masked_bit_not_zero

TAGS **UNPREDICTABLE** DISPLAY DBGBVRn_EL1.VA contains a bit masked by DBGBCRn_EL1.MASK that is not zero.

ArchMsg.Warning.warning_ccsidr_unimplemented_level

TAGS **UNPREDICTABLE** DISPLAY CCSIDR read while CSSELR %{CSSELR:x} points to an unimplemented cache level. Fields:

CSSELR unsigned int

current effective value of CSSELR.

ArchMsg.Warning.warning_change_to_ns_when_tge_set

TAGS **UNPREDICTABLE** DISPLAY Attempting to change NS to 1 when HCR.TGE = 1.

ArchMsg.Warning.warning_contiguous_bit_check_abort

TAGS **UNPREDICTABLE** DISPLAY An abort occurred whilst attempting to check TLB entry is contiguous at %{ENTRY_ADDR}. Fields:

ENTRY_ADDR unsigned int

address of conflicting TLB entry.

ArchMsg.Warning.warning_contiguous_bit_error

TAGS **UNPREDICTABLE** DISPLAY TLB entry at %{ENTRY_ADDR} was not contiguous with entry at %{CONTIG_ADDR} Contents are %{ENTRY_DATA} but expected %{CONTIG_DATA}. Fields:

CONTIG_ADDR unsigned int

address of the TLB first read, with which this entry is expected to be contiguous.

CONTIG_DATA unsigned int

expected contents based on the entry at CONTIG_ADDR.

ENTRY_ADDR unsigned int

address of conflicting TLB entry.

ENTRY_DATA unsigned int

contents of conflicting TLB entry.

ArchMsg.Warning.warning_contiguous_bit_error_D128

TAGS **UNPREDICTABLE** DISPLAY TLB entry at %{ENTRY_ADDR} was not contiguous with entry at %{CONTIG_ADDR} Contents are %{ENTRY_DATA_HI} %{ENTRY_DATA_LO} but expected %{CONTIG_DATA}. Fields:

CONTIG_ADDR unsigned int

address of the TLB first read, with which this entry is expected to be contiguous.

CONTIG_DATA unsigned int

expected contents based on the entry at CONTIG_ADDR.

ENTRY_ADDR unsigned int

address of conflicting TLB entry.

ENTRY_DATA_HI unsigned int

Higher bit contents of conflicting TLB entry.

ENTRY_DATA_LO unsigned int

Lower bit contents of conflicting TLB entry.

ArchMsg.Warning.warning_cp10_cp11_mismatch

TAGS **UNPREDICTABLE** DISPLAY Unpredictable configuration of CP10 and CP11 controls at EL
%{EL:d}. Fields:

EL unsigned int

EL owning the control.

ArchMsg.Warning.warning_cp10_cp11_reserved_value

TAGS **UNPREDICTABLE** DISPLAY Unpredictable configuration of CPACR CP10 (%{CP10}) or
CP11 (%{CP11}). Fields:

CP10 unsigned int

CP10 access.

CP11 unsigned int

CP11 access.

ArchMsg.Warning.warning_csselr_level_out_of_range

TAGS **UNPREDICTABLE** DISPLAY CSSELR.Level out of range, written value %{LEVEL} with only
%{IMPLEMENTED:d} levels implemented. Fields:

IMPLEMENTED unsigned int

number of cache levels implemented.

LEVEL unsigned int

written value of CSSELR.Level.

TYPE enum

The type of cache selected by CSSELR.InD and TnD.

ArchMsg.Warning.warning_debug_flow_control_bits_not_obeyed

TAGS **UNPREDICTABLE** DISPLAY Flow control bits for %{REGISTER} were not obeyed, causing
%{ERROR}. Fields:

ERROR enum

type of error resulting.

REGISTER enum

register under consideration.

ArchMsg.Warning.warning_debug_register_access_during_reset

TAGS **UNPREDICTABLE** DISPLAY Unpredictable %{IS_WRITE:(read|write)} access to debug register offset %{OFFSET} during reset. Fields:

IS_WRITE unsigned int

Write Not Read.

OFFSET unsigned int

Register Offset.

ArchMsg.Warning.warning_decode_cps_inconsistent_fields

TAGS **UNPREDICTABLE** DISPLAY The mode setting fields in the CPS instruction are inconsistent.

ArchMsg.Warning.warning_decode_invalid_state

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable in current exception-level/security state.

ArchMsg.Warning.warning_default_cacheable_mmu_on

TAGS **UNPREDICTABLE** DISPLAY Default cacheable is enabled (HCR.DC = 1) while MMU is enabled.

ArchMsg.Warning.warning_default_cacheable_vmmu_off

TAGS **UNPREDICTABLE** DISPLAY Default cacheable is enabled (HCR.DC = 1) while HCR.VM = 0.

ArchMsg.Warning.warning_deprecated_wvr_bit_2_set

TAGS **UNPREDICTABLE** DISPLAY DBGWVR%{N:d}_EL1=%{WVR:h} has bit 2 set, deprecated in ARMv8. Fields:

N unsigned int

index.

WVR unsigned int

DBGWVR.

ArchMsg.Warning.warning_eret_while_software_step_active_pending

Missing ISB between setting MDSCR_EL1.SS and ERET TAGS **UNPREDICTABLE**.

ArchMsg.Warning.warning_exclusive_to_non_normal

TAGS **UNPREDICTABLE** DISPLAY Exclusive Access to Non-Normal Memory %{ADDR}. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_exclusive_to_non_writeback

TAGS **UNPREDICTABLE** DISPLAY Exclusive Access to Non-Writeback-Cacheable Memory %{ADDR}. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_execute_from_device_memory

TAGS **UNPREDICTABLE** DISPLAY An attempt was made to execute from Device memory.

ArchMsg.Warning.warning_illegal_cpsr_mode

TAGS **UNPREDICTABLE** DISPLAY Writing an illegal or unimplement mode ({NEW_MODE:h}) to CPSR was unpredictable. Fields:

NEW_MODE unsigned int

New value of CPSR.M.

ArchMsg.Warning.warning_illegal_srs_mode

TAGS **UNPREDICTABLE** DISPLAY Illegal or **UNPREDICTABLE** mode ({MODE:h}) used for SRS instruction. Fields:

MODE unsigned int

mode for Banked SP.

ArchMsg.Warning.warning_implementation_defined_read_debug_register_in_SCS

Read Access to Debug registers from the processor is **IMPLEMENTATION DEFINED** TAGS **IMP_DEF**. Fields:

PPB_OFFSET unsigned int

debug register offset.

ArchMsg.Warning.warning_implementation_defined_sequential_security_transitions_supported

The behavior of sequential instruction fetches that cross from non-secure to secure memory and contain an SG instruction is **CONSTRAINED UNPREDICTABLE** TAGS **UNPREDICTABLE**.

ArchMsg.Warning.warning_implementation_defined_stack_limit_check_supported

It is **IMPLEMENTATION DEFINED** whether stack pointer limit checking is performed for this instruction TAGS **IMP_DEF**. Fields:

ADDRESS unsigned int

Address of instruction.

OPCODE unsigned int

instruction opcode.

ArchMsg.Warning.warning_implementation_defined_taggednonsharedmemorytype

TAGS **IMP_DEF** DISPLAY Memory attribute has Tagged, non shared memory type; it is **IMPLEMENTATION DEFINED** whether the memory location is treated as Tagged or non-Tagged.

ArchMsg.Warning.warning_implementation_defined_write_debug_register_in_SCS

Write Access to Debug registers from the processor is **IMPLEMENTATION DEFINED** TAGS **IMP_DEF**. Fields:

DATA unsigned int

data attempted to be written.

PPB_OFFSET unsigned int

debug register offset.

ArchMsg.Warning.warning_invalid_tcr_granule

TAGS UNPREDICTABLE DISPLAY TCR.TG{%TG_ID:(0|1)} bits have been set to a granule size {%REQUEST:d}K not implemented - using {%SUBSTITUTE:d}K. Fields:

REQUEST signed int

page size requested (or 0 for reserved).

SUBSTITUTE signed int

best guess available page size.

TG_ID bool

bits TG0 or TG1.

ArchMsg.Warning.warning_load_multiple_user_registers_from_user_mode

TAGS UNPREDICTABLE DISPLAY An LDM(user registers) instruction executed from user mode.

ArchMsg.Warning.warning_mte_in_mem_attr_not_supported

TAGS UNPREDICTABLE DISPLAY Memory attribute {%ATTR} has Tagged memory type but Armv8.5-MTE is not implemented. Fields:

ATTR unsigned int

Memory attribute.

ArchMsg.Warning.warning_reserved_breakpoint_state_match

TAGS UNPREDICTABLE DISPLAY {%IS_BREAKPOINT:(Watchpoint|Breakpoint)} programmed with a reserved combination of HMC, SSC and {%IS_BREAKPOINT:(PAC|PMC)}. Fields:

IS_BREAKPOINT unsigned int

Is a breakpoint, not a watchpoint.

ArchMsg.Warning.warning_shareability

TAGS UNPREDICTABLE DISPLAY Unpredictable: combination of the 1st and 2nd stages of translation is Normal Inner Non-Cacheable, Outer Non-Cacheable.

ArchMsg.Warning.warning_software_step_set_while_enabled

TAGS UNPREDICTABLE DISPLAY MDSCR_EL1.SS set to 1 while software step debug exceptions are enabled.

ArchMsg.Warning.warning_thumb_instruction_wraps_around_memory

TAGS UNPREDICTABLE DISPLAY Thumb instruction wraps around memory.

ArchMsg.Warning.warning_ttbr_sbz_bits_are_not_zero

TAGS UNPREDICTABLE DISPLAY {%TTBR}[{%MSB:d}:0] should be zero, but are not. Fields:

MSB unsigned int

x-1.

TTBR enum

which TTBR register.

ArchMsg.Warning.warning_unaligned_address_dbgdtrrx_write

TAGS **UNPREDICTABLE** DISPLAY Unaligned address %{ADDR} in X0/R0 when DBGDTRRX written in memory access mode. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unaligned_address_dbgdtrtx_read

TAGS **UNPREDICTABLE** DISPLAY Unaligned address %{ADDR} in X0/R0 when DBGDTRTX read in memory access mode. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unaligned_to_device

TAGS **UNPREDICTABLE** DISPLAY Unaligned Access to Device Memory %{ADDR}. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unaligned_to_strongly_ordered

TAGS **UNPREDICTABLE** DISPLAY Unaligned Access to Strongly Ordered Memory %{ADDR}. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unknown_sau_rnr

TAGS **UNPREDICTABLE** DISPLAY SAU_RNR was set to an unsupported value.

ArchMsg.Warning.warning_unpredictable_AIRCR_PRIS_and_BFHFNMINs

TAGS **UNPREDICTABLE** DISPLAY The effect of setting both AIRCR.BFHFNMINs and AIRCR.PRIS to 1 is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_AIRCR_VECTCLRACTIVE_when_not_in_debug

TAGS **UNPREDICTABLE** DISPLAY The effect of writing a 1 to AIRCR.VECTCLRACTIVE if the processor is not halted in Debug state is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_AIRCR_VECTRESET_and_SYSRESETREQ

TAGS **UNPREDICTABLE** DISPLAY When the processor is halted in Debug state, if a write to the register writes a 1 to both VECTRESET and SYSRESETREQ, the behavior is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_AIRCR_VECTRESET_when_not_in_debug

TAGS **UNPREDICTABLE** DISPLAY The effect of writing a 1 to AIRCR.VECTRESET if the processor is not halted in Debug state is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_AIRCR_incorrect_VKEY

TAGS **UNPREDICTABLE** DISPLAY The value 0x05FA must be written to AIRCR.VKEY, otherwise the register write is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_EXC_RETURN_Reserved_Bit

EXC_RETURN[23:7] are reserved with the special condition that all bits should be written as one. Values other than all 1s are **UNPREDICTABLE**. TAGS **UNPREDICTABLE**. Fields:

EXC_RETURN unsigned int

Special PC value.

SBO_MASK unsigned int

Which bits Should Be One to not UNPRED.

ArchMsg.Warning.warning_unpredictable_PMEVTYPEPn_ELO_VS_reserved_value

TAGS **UNPREDICTABLE** DISPLAY Unpredictable reserved value of PMEVTYPEN<n>_ELO.VS.

ArchMsg.Warning.warning_unpredictable_change_to_DEMCR_MON_STEP_at_insufficient_priority

TAGS **UNPREDICTABLE** DISPLAY The effect of changing DEMCR.MON_STEP at an execution priority that is lower than the priority of the DebugMonitor exception is **CONSTRAINED UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_change_to_priority_of_active_exception

TAGS **UNPREDICTABLE** DISPLAY Changing the priority of an active exception is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_clear_lspact

TAGS **UNPREDICTABLE** DISPLAY **UNPREDICTABLE** state on context switch. Fields:

DRVEXC_TAKEN unsigned int

Derived exception taken, not pended.

REACHED_LAST_STORE unsigned int

Only the last faulted.

ArchMsg.Warning.warning_unpredictable_dbg_exit_unaligned_dlr

TAGS **UNPREDICTABLE** DISPLAY Debug exit to AArch32 with DLR[0] set to 1. PC[0] can be set to 0 or to DLR[0].

ArchMsg.Warning.warning_unpredictable_exception_catch

TAGS **UNPREDICTABLE** DISPLAY Generation of an exception catch event is unpredictable.

ArchMsg.Warning.warning_unpredictable_exception_return_inconsistent_state

TAGS **UNPREDICTABLE** DISPLAY state on exception return is unpredictable.

ArchMsg.Warning.warning_unpredictable_exception_return_instruction

TAGS **UNPREDICTABLE** DISPLAY exception return instruction is unpredictable.

ArchMsg.Warning.warning_unpredictable_in_debug_state

TAGS **UNPREDICTABLE** DISPLAY Instruction is **UNPREDICTABLE** when executed in debug state.

ArchMsg.Warning.warning_unpredictable_mair_encoding

TAGS **UNPREDICTABLE** DISPLAY Memory attribute %{ATTR} has invalid MAIR encoding; this may lead to unpredictable results. Fields:

ATTR unsigned int

Memory attribute.

ArchMsg.Warning.warning_unpredictable_pmu_counter_access

TAGS **UNPREDICTABLE** DISPLAY Access to PMU counters from non-secure EL0 or EL1 is **UNPREDICTABLE** with MDCR_EL2.HPMN set to 0.

ArchMsg.Warning.warning_unpredictable_prioritization_breakpoint_match_vector_catch

TAGS **UNPREDICTABLE** DISPLAY Two events with the same priority occurred at the same instruction: Address Matching Vector Catch debug event, and Breakpoint debug event. It is constrained unpredictable which is taken.

ArchMsg.Warning.warning_unpredictable_stack_selection

TAGS **UNPREDICTABLE** DISPLAY **UNPREDICTABLE** state on context switch.

ArchMsg.Warning.warning_unpredictable_tsize_out_of_range

TAGS **UNPREDICTABLE** DISPLAY %{TCR}.T%{TTBR:d}SZ (%{TSIZE:d} bits) is out of range. Must be between 25 and %{TSIZE_MAX:d}. Fields:

TCR enum

Which TCR.

TSIZE signed int

Value of TSize.

TSIZE_MAX signed int

The maximum allowed value.

TTBR unsigned int

Which TTBR region.

ArchMsg.Warning.warning_unpredictable_unaligned_pc_as_base_register

TAGS **UNPREDICTABLE** DISPLAY The PC value must be word-aligned, otherwise the behavior of the instruction is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_unaligned_pop_stack

TAGS **UNPREDICTABLE** DISPLAY **CONSTRAINED UNPREDICTABLE** unaligned stack frame %{FRAME_ADDR:x} in exception-return. Fields:

FRAME_ADDR unsigned int

unaligned stack frame address.

ArchMsg.Warning.warning_unpredictable_unaligned_to_non_writeback

TAGS **UNPREDICTABLE** DISPLAY Unaligned Access to Non-Writeback-Cacheable Memory.

ArchMsg.Warning.warning_unpredictable_vmsa_memattrib

TAGS **UNPREDICTABLE** DISPLAY Unpredictable vmsa memory attribute with texcb value programmed to %{TEXCB} when tex remap is %{TEX_REMAP}. Fields:

TEXCB unsigned int

texcb value.

TEX_REMAP bool

use tex remap.

ArchMsg.Warning.warning_unpredictable_vtcr_t0sz_sl0

TAGS **UNPREDICTABLE** DISPLAY The combination of VTCR.TOSZ=%{TOSZ:d} and VTCR.SLO=%{SLO:d} is unpredictable. Fields:

SLO signed int

VTCR.SLO, starting level for stage 2 translation table walks.

TOSZ signed int

VTCR.TOSZ, required input address range.

ArchMsg.Warning.warning_unpredictable_write_DHCSR

TAGS **UNPREDICTABLE** DISPLAY **UNPREDICTABLE** set of bit-changes in DHCSR. Fields:

NEW unsigned int

data attempted to be written.

OLD unsigned int

previous value.

ArchMsg.Warning.warning_unpredictable_write_SHCSR

TAGS **UNPREDICTABLE** DISPLAY **UNPREDICTABLE** setting a pending bit to 1 for an exception with priority >= execution priority.

ArchMsg.Warning.warning_unsupported_access_to_memory_mapped_register

TAGS **UNPREDICTABLE** DISPLAY Access to memory mapped register is unsupported.

ArchMsg.Warning.warning_user_jmcr_access

TAGS **UNPREDICTABLE** DISPLAY User %{WRITE:(write|read)} access to JMCR is unpredictable. Fields:

WRITE unsigned int

access is write.

ArchMsg.Warning.warning_wcr_mask_and_bas

TAGS **UNPREDICTABLE** DISPLAY DBGWCR_EL1.MASK is non-zero and BAS is not 0xff.

ArchMsg.Warning.warning_wcr_mask_reserved

TAGS **UNPREDICTABLE** DISPLAY DBGWCR_ELn.MASK is set to a reserved value.

ArchMsg.Warning.warning_wcr_non_configuous_bas

TAGS **UNPREDICTABLE** DISPLAY DBGWCR_EL1.BAS has non-contiguous set of ones.

ArchMsg.Warning.warning_wvr_masked_bit_not_zero

TAGS **UNPREDICTABLE** DISPLAY DBGWVRn_EL1.VA contains a bit masked by DBGWCRn_EL1.MASK that is not zero.

ArchMsg.Why.why_branch_target_exception

A branch target exception has occurred. DISPLAY branch target exception: target pc=%{PC}, target opcode=%{OPCODE}. PRIMARY KEY PC. Fields:

BTYPE unsigned int

pstate.BTYPE at time branch taken.

OPCODE unsigned int

opcode at branch target.

PC unsigned int

Virtual Address of branch target.

ArchMsg.Why.why_illegal_state

An illegal mode change has occurred. DISPLAY illegal mode change: %{MESSAGE}. Fields:

MESSAGE string

The message.

ArchMsg.warning_invalid_interval_counter_reload_value

Invalid interval counter reload value in PMSIRR_EL1.INTERVAL: software should set this to a value greater than or equal to the minimum indicated by PMSIDR_EL1.Interval.

ArchMsg.warning_invalid_spe_buffer_write_pointer

Invalid value programmed in PMBPTR_EL1.PTR or PMBLIMITR_EL1.LIMIT: Current write pointer must be at least one sample record below the write limit pointer and PMBPTR_EL1.PTR[63:56] must be equal to PMBLIMITR_EL1.LIMIT[63:56].

BRANCH_MISPREDICT

Simulating branch mispredict. Fields:

PC unsigned int

Origin address (or 0 if unavailable).

BRA_DIR

Direct branches, to immediate address. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISSET enum

The instruction set of the branch instruction.

IS_COND bool

Indicates if this is a conditional branch.

IS_HINTED bool

Whether the conditional instruction is hinted.

IS_ISB bool

Whether the instruction is ISB.

IS_LINK bool

Indicates if this is a branch with link.

PC unsigned int

The address of the branch instruction.

TARGET_ISET enum

The instruction set after the branch.

TARGET_PC unsigned int

The address the instruction branches to.

BRA_INDIR

Indirect branches, perhaps to a register. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISET enum

The instruction set of the branch instruction.

IS_COND bool

Indicates if this is a conditional branch.

IS_HINTED bool

Whether the conditional instruction is hinted.

IS_ISB bool

Whether the instruction is ISB.

IS_LINK bool

Indicates if this is a branch with link.

PC unsigned int

The address of the branch instruction.

TARGET_ISET enum

The instruction set after the branch.

TARGET_PC unsigned int

The address the instruction branches to.

BRA_INDIR_CCFAIL

Indirect branches fail, perhaps to a register.

BRA_RET_CCFAIL

Branch return fail, perhaps to a register.

CACHE_MAINTENANCE_OP

Cache Maintenance Operation. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

DATA unsigned int

Specified MVA or Set/way.

FUNCTION enum

Clean / Invalidate.

SCOPE enum

Affected region.

SIDE enum

Inst / Data.

CCFAIL

Conditional instruction condition check fail. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CCFAIL_UNC

Unconditional instruction condition check fail. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CCPASS

Conditional instruction condition check pass. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CCPASS_UNC

Unconditional instruction condition check pass. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CODE_CACHE_MAINT

Code cache maintenance. Fields:

END_INCL unsigned int

Inclusive end address for VA or PA requests.

NS bool

Non-secure world for PA.

START unsigned int

Start address for VA or PA requests.

TYPE enum

Request type.

COMPILE_BLOCK_END

Last instruction of basic block translated. Fields:

VADDR unsigned int

Address of next instruction after this basic block.

COMPILE_BLOCK_START

First instruction of basic block translated. Fields:

VADDR unsigned int

Address of first instruction in basic block.

COMPILE_INST

ARM instruction compiled. Fields:

DISASS string

Disassembly of the instruction.

ISSET enum

The instruction set of this instruction.

ITSTATE unsigned int

The ITSTATE current for the instruction.

OPCODE unsigned int

The opcode of the instruction.

PC unsigned int

The address of the instruction.

SIZE unsigned int

The size of the instruction in bytes.

CONTEXTIDR

Context ID Register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

NS bool

Secure or nonsecure banked register is accessed.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The new value written.

CONTEXT_SYNC

Called for every context synchronization event. Fields:

ARCH_CSE bool

Architectural Context Synchronization Event.

PC unsigned int

The address of the synchronization instruction.

SYSREG_SYNC bool

System register synchronization event.

CORE_ENDIAN

Core BE8 Big-Endian state changed. Fields:

BE8 bool

Core BE8 Big-Endian state.

CORE_INFO

Static processor attributes. Only triggered by a call to DumpState(). Fields:

ARCH_PROFILE enum

The architecture profile of the core.

CLUSTER_ID unsigned int

The cluster ID of this processor.

CORE_NUM unsigned int

The number of this core in an MP processor.

DETAILED_CORE_FEATURES enum

Detailed core features as a bitfield. Enum values are bitmasks to extract the info from the field.

FPU_VERSION enum

The VFP version implemented by the core.

MEM_ARCH enum

The memory architecture of the core.

MIDR_EL1 unsigned int

MIDR_EL1 value.

NUM_CORES unsigned int

The number of cores in this MP processor.

QUANTUM_SIZE unsigned int

The default quantum size of the core (in ticks).

SECURITY_FEATURES bool

Does the core have security features?.

CORE_LOADS

Processor load accesses. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ACQREL enum

Is this an acquire/release.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DATA unsigned int

The data read or written.

ELEMENT_SIZE unsigned int

Width of each element.

ENDIANNESS enum

Byte ordering of this access.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

LOCK enum

Normal, exclusive or locked access.

MECID unsigned int

Memory Encryption Context ID, if applicable.

MECID2 unsigned int

Memory Encryption Context ID of the second page, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PAGE_SIZE unsigned int

Page size as $\log_2(\text{size in bytes})$.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK, 2=Exclusive Failed, 3=Unsupported.

SIZE unsigned int

Width of the access in bytes. Only required if DATA is not traced.

TRANS bool

Is this a translated access.

VADDR unsigned int

The virtual address of the access.

CORE_REGS

Changes of the core registers R0 to R14. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number, 0 to 14.

MODE enum

Bank of the register accessed.

OLD_VALUE unsigned int

The old value overwritten.

PHYS_ID enum

The physical register accessed.

VALUE unsigned int

The new value written to the register.

CORE_REGS64

Changes of the core registers X0..X30, SP_ELn. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ID enum

The register number, 0-30 for X0-X30, >=32 for SP_ELn.

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

CORE_REGS64_READ

General purpose AArch64 register has been read X0..X30, SP_ELn. Fields:

ID enum

The register number, 0-30 for X0-X30, >=32 for SP_ELn.

VALUE unsigned int

The value read from the register.

CORE_STALLED_TRANSITION

Detect when a core is prevented from progressing PChannel states. Fields:

CURRENT_STATE enum

Current power mode of core.

REASON string

The reason a transition was stalled, and what must be done to unblock it.

TARGET_STATE enum

Desired power mode of core after transition.

CORE_STORES

Processor store accesses. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ACQREL enum

Is this an acquire/release.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DATA unsigned int

The data read or written.

ELEMENT_SIZE unsigned int

Width of each element.

ENDIANNESS enum

Byte ordering of this access.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

LOCK enum

Normal, exclusive or locked access.

MECID unsigned int

Memory Encryption Context ID, if applicable.

MECID2 unsigned int

Memory Encryption Context ID of the second page, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PAGE_SIZE unsigned int

Page size as $\log_2(\text{size in bytes})$.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK, 2=Exclusive Failed, 3=Unsupported.

SIZE unsigned int

Width of the access in bytes. Only required if DATA is not traced.

TRANS bool

Is this a translated access.

VADDR unsigned int

The virtual address of the access.

CP14_READ

System Coprocessor register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP14 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc1 unsigned int

opcode 1.

opc2 unsigned int
opcode 2.

CP14_WRITE

System Coprocessor register write. Fields:

CORE_NUM unsigned int
Core number in a multi processor.

CRm unsigned int
CRm.

CRn unsigned int
CRn.

NS bool
Secure or nonsecure banked register is accessed.

REG_NAME string
Name of the CP14 register accessed.

SECURITY_STATE enum
The current security state.

UNDEF bool
The register accessed is undefined.

UPDATED_VALUE unsigned int
Updated value of the register now it has been written.

VALUE unsigned int
The new value written.

opc1 unsigned int
opcode 1.

opc2 unsigned int
opcode 2.

CP15_READ

System Coprocessor register read. Fields:

CORE_NUM unsigned int
Core number in a multi processor.

CRm unsigned int
CRm.

CRn unsigned int
CRn.

NS bool
Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP15 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc1 unsigned int

opcode 1.

opc2 unsigned int

opcode 2.

CP15_READ64

System Coprocessor register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP15 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc unsigned int

opcode 1.

CP15_WRITE

System Control Coprocessor register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP15 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc1 unsigned int

opcode 1.

opc2 unsigned int

opcode 2.

CP15_WRITE64

System Coprocessor register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP15 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc unsigned int

opcode 1.

CPSR

CPSR change. (In AArch64 PSTATE represented in SPSR format). Fields:

CORE_NUM unsigned int

Core number in a multi processor.

OLD_VALUE unsigned int

The old CPSR value.

UNKNOWN unsigned int

Bits within the register that have unknown value.

VALUE unsigned int

The new CPSR value.

CRYPTO_SPEC

Every crypto instruction speculatively executed.

CorePowerStateChange

Core power-up state has changed. DISPLAY Core powered %{PU:(up|down)}. Fields:

PU bool

Whether the core powered up.

DATA_CACHE_SET_ALLOC_TAGS

Set allocation tags from DC GVA instruction. Fields:

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

NSDESC unsigned int

The security state of the access.

PADDR unsigned int

The physical address of the access.

PAS enum

The physical address space of the page.

RESPONSE enum

0=Aborted, 1=OK.

SIZE unsigned int

The size of memory in bytes.

TAG unsigned int

The value of the tag being set.

TAGGING_TYPE enum

Type of Memory tagging (Physical or Virtual).

TAG_VADDR unsigned int

The virtual address of the Virtual Tag if Virtual tagging is used; otherwise 0.

VADDR unsigned int

Virtual address used for setting allocation tags.

DATA_CACHE_ZERO

Zero and invalidate cache line from DC ZVA instruction. Fields:

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

NSDESC unsigned int

The security state of the access.

PADDR unsigned int

The physical address of the access.

PADDR_TAG unsigned int

If tag checked, the physical address tag for the access.

PAS enum

The physical address space of the page.

RESPONSE enum

0=Aborted, 1=OK.

SIZE unsigned int

The size of zero'd memory in bytes.

TAG_CHECKED bool

Is this access tag checked.

VADDR unsigned int

Virtual address of the zero'd cache line.

DEBUG_EVENT

Hardware debug support event. Fields:

EVENT enum

Description of event.

VALUE unsigned int

data value.

DEBUG_TARGET_ISET

Hardware debug target ISA. Fields:

TARGET_ISET enum

The target instruction set of the debug state.

DMI_ALLOCATE

DMI allocated. Fields:

ACCESS_DIRECTION enum

READ / WRITE.

PADDR unsigned int

The physical address of the DMI.

PTR unsigned int

DMI range pointer.

SIDE enum

DSIDE / ISIDE.

SIZE unsigned int

The size of DMI.

VADDR unsigned int

The virtual address of the DMI..

DMI_HIT

DMI hit, access used DMI (only fire for small DMI regions i.e. less than a full page). Fields:

ACCESS_DIRECTION enum

READ / WRITE.

PADDR unsigned int

The physical address of the DMI.

SIDE enum

DSIDE / ISIDE.

SIZE unsigned int

The size of the DMI.

VADDR unsigned int

The virtual address of the DMI..

DMI_REVOKE

DMI revoked. Fields:

ACCESS_DIRECTION enum

READ / WRITE.

PADDR unsigned int

The physical address of the DMI.

SIDE enum

DSIDE / ISIDE.

SIZE unsigned int

The size of DMI.

VADDR unsigned int

The virtual address of the DMI..

END_COMPILE

Compilation end. Fields:

END_OF_PAGE_COUNT unsigned int

Number of basic blocks exited due to page boundary since START_COMPILE.

FETCHFAIL_COUNT unsigned int

Number of basic blocks exited due to ifetch failure START_COMPILE.

INST_COUNT unsigned int

Number of instructions compiled since START_COMPILE.

NONSEQ_COUNT unsigned int

Number of basic blocks exited due to non-sequential instructions since START_COMPILE.

PAGE_STRADDLE_COUNT unsigned int

Number of basic blocks exited due to unaligned instructions crossing page since START_COMPILE.

ESB

Error Synchronization Barrier hit. Fields:

ASYNC bool

ESB is async.

IMPLICIT bool

ESB is implicit.

EXCEPTION

Exception taken. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ESR unsigned int

The value assigned to ESR when exception is taken.

ESR_EC unsigned int

The Value assigned to ESR.EC when exception is taken.

IS_PHYSICAL bool

Physical or Virtual exception.

IS_TRAP bool

Exception coming from a trap.

LR unsigned int

The value assigned to the link register.

PC unsigned int

The location where the exception occurred.

PREFERRED_RETURN unsigned int

The preferred return address for the exception.

TAKEN_LOCALLY bool

Whether exception is taken locally(same exception level).

TARGET_ISET enum

The instruction set of the exception handler code.

TARGET_PC unsigned int

The address the exception branches to.

VECTOR enum

The exception vector.

VECTOR_OFFSET unsigned int

The offset of an exception vector from the base address.

EXCEPTION_END

Every exception completed.

EXCEPTION_INFO

Trace human-readable reason of the current exception for data/instruction aborts, floating point exceptions and unknown exceptions. Fields:

DESCRIPTION string

Reason for current exception.

ESR_EC unsigned int

The value of ESR.EC when exception is taken.

EXCEPTION_RAISE

Every exception raised.

EXCEPTION_RETURN

Branches on leaving exception. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISET enum

The instruction set of the branch instruction.

IS_COND bool

Indicates if this is a conditional branch.

IS_HINTED bool

Whether the conditional instruction is hinted.

IS_ISB bool

Whether the instruction is ISB.

IS_LINK bool

Indicates if this is a branch with link.

PC unsigned int

The address of the branch instruction.

TARGET_ISET enum

The instruction set after the branch.

TARGET_PC unsigned int

The address the instruction branches to.

EXCEPTION_RETURN_PREBRANCH

Exception return event that must occur before PC and processor state have updated.

EXCEPTION_START

Every exception started.

FIQ_TAKEN

FIQ taken exception.

FP_EXCEPTIONS_SIGNALLED

Trace which FP exceptions were signalled as a part of the FP operations performed by an instruction. Fields:

FP_EXCEPTIONS_SIGNALLED_MASK unsigned int

Indicates which FP exceptions were signalled by an instruction. This uses the same encoding as the cumulative floating-point exception bitfields in FPSR.

FP_STATE

Floating point state. Fields:

FPSR unsigned int

FPSR register value.

MASK unsigned int

The mask indicating updated cumulative bits.

FREQ_CHANGED

Core frequency changed. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The instruction count of this core.

NEW_FREQ signed int

The new frequency of this core (expressed in Hz).

OLD_FREQ signed int

The old frequency of this core (expressed in Hz).

HLT

HLT instruction occurred. Fields:

IMM16 unsigned int

The 16-bit immediate value encoded in HLT instruction.

INST

Every instruction executed. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DEBUG_STATE bool

The instruction is executed in debug state.

DISASS string

Disassembly of instruction.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISSET enum

The current instruction set.

ITSTATE unsigned int

The current ITSTATE.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

MODE enum

The mode the core is in.

NS unsigned int

The core's non-secure bit.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

OPCODE unsigned int

The opcode of the instruction.

PADDR unsigned int

The physical address of the instruction.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the instruction.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the page.

PC unsigned int

The address of the instruction.

SECURITY_STATE enum

The core's security state.

SIZE unsigned int

The size of the instruction in bytes.

INST_END

Every instruction completed.

INST_START

Every instruction started. Fields:

DEBUG_STATE bool

The instruction is executed in debug state.

ISSET enum

The current instruction set.

MODE enum

The mode the core is in.

NS enum

The current Secure State.

PC unsigned int

The address of the conditional instruction.

SECURITY_STATE enum

The current Security State.

INST_STRADDLE

Instruction straddles boundary. Fields:

DEBUG_STATE bool

The instruction is executed in debug state.

ISSET enum

The current instruction set.

MODE enum

The mode the core is in.

NS enum

The current Secure State.

PC unsigned int

The address of the conditional instruction.

SECURITY_STATE enum

The current Security State.

IRQ_TAKEN

IRQ taken exception.

ISB

ISB instruction occurred. Fields:

PC unsigned int

The address of the ISB instruction.

LOAD_MULTIPLE_REGS_LIST

List of destination registers (not source) of a multiple registers load instruction. Fields:

REG string

Register to add into the list.

LOCAL_MONITOR

Local monitor activity. Fields:

PADDR unsigned int

Local Monitor Address.

State enum

State of the monitor (Open/Exclusive).

MAPPING_CREATE

A memory mapping has been created. This is an implementation detail of the model and so may change without notice. Fields:

ASID unsigned int

The ASID used during translation, or 0 if no ASID was used.

IS_VALID bool

Whether the mapping is valid or not.

PADDR unsigned int

Base physical address of region, or 0 if not known.

SIDE enum

I-side or D-side.

SIZE unsigned int

Size of the page, or 0 if not known.

VADDR unsigned int

Base virtual address of region.

VMID unsigned int

The VMID used during translation, or 0 if no VMID was used.

MAPPING_DESTROY

A memory mapping has been destroyed. This is an implementation detail of the model and so may change without notice. Fields:

ASID unsigned int

The ASID used during translation.

CAUSE enum

The cause of invalidation.

PADDR unsigned int

Base physical address of page.

SIDE enum

I-side or D-side.

SIZE unsigned int

Size of the page.

VADDR unsigned int

Base virtual address of page.

VMID unsigned int

The VMID used during translation.

MEMMAP_DEBUG_READ

Memory mapped reads to the debug registers. Fields:

ADDR unsigned int

address.

CORE_NUM unsigned int

Core number in a multi processor.

EXT bool

Whether access is from an external device (such as the DAP).

NS enum

Secure state of the access.

REG_NAME string

Name of the debug register accessed.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

MEMMAP_DEBUG_WRITE

Memory mapped writes to the debug registers. Fields:

ADDR unsigned int

address.

CORE_NUM unsigned int

Core number in a multi processor.

EXT bool

Whether access is from an external device (such as the DAP).

NS enum

Secure state of the access.

REG_NAME string

Name of the debug register accessed.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The new value written.

MMU_D128_TTB_READ

This event is triggered by reads caused by a D128 translation table walk. Fields:

ABORTED bool

Was the walk successful.

DATA_HIGH unsigned int

The data upper bits read.

DATA_LOW unsigned int

The data lower bits read.

IPA unsigned int

For stage 1, the IPA of the read.

LEVEL unsigned int

Translation table level.

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

OUTADDR unsigned int

The output physical address.

PADDR unsigned int

The physical address of the read.

PASmem enum

Physical Address Space of the accessed memory.

PASreq enum

Physical Address Space of the requested lookup.

REGIME_EL enum

Entry matches in this translation regime.

REQUESTER enum

The requester of this table walk.

SIDE enum

Inst / Data.

STAGE unsigned int

Translation stage.

MMU_D128_TTB_WRITE

This event is triggered by writes caused by a D128 translation table walk. Fields:

ABORTED bool

Was the walk successful.

DATA_HIGH unsigned int

The data upper bits written.

DATA_LOW unsigned int

The data lower bits written.

IPA unsigned int

For stage 1, the IPA of the write.

IS_HACDBS_ACCESS bool

If the access was due to HACDBS_CLEAN.

LEVEL unsigned int

Translation table level.

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

OUTADDR unsigned int

The output physical address.

PADDR unsigned int

The physical address of the write.

PASmem enum

Physical Address Space of the accessed memory.

PASreq enum

Physical Address Space of the requested lookup.

REGIME_EL enum

Entry matches in this translation regime.

SIDE enum

Inst / Data.

STAGE unsigned int

Translation stage.

MMU_TRANS

Address translation information. Fields:

ASID unsigned int

Address space identifier.

CORE_NUM unsigned int

Core number in a multi processor.

Hyp bool

Entry matches in Hyp state only.

INNERCACHE_RA bool

Is the inner cache allocate on read.

INNERCACHE_TRANSIENT bool

Is the inner write-through transient.

INNERCACHE_TYPE enum

Inner Caching scheme (NC/MB/WA).

INNERCACHE_WA bool

Is the inner cache allocate on write.

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MEMTYPE enum

Memory type.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC enum

Is secure side supposed to access secure or nonsecure memory.

OUTERCACHE_RA bool

Is the outer cache allocate on read.

OUTERCACHE_TRANSIENT bool

Is the outer write-through transient.

OUTERCACHE_TYPE enum

Outer Caching scheme (NC/MB/WA).

OUTERCACHE_WA bool

Is the outer cache allocate on write.

PADDR unsigned int

Physical address of the access.

PAGESIZE unsigned int

Page size as log2(size).

PAS enum

PAS supposed to be accessed is secure or nonsecure memory or Root/Realm if RME is implemented.

SH enum

Shareability.

SIDE enum

Inst / Data.

STAGE1_PERM unsigned int

Stage 1 permissions, [5:3] Privileged access (XWR) [2:0] User access (XWR).

STAGE2_PERM unsigned int

Stage 2 permission mask [5:3] Privileged access (XWR) [2:0] User access (XWR).

VADDR unsigned int

Virtual address of the access.

VMID unsigned int

Virtual machine identifier.

XS bool

XS Attribute.

nG enum

Flag indicating whether ASID will be matched.

MMU_TTB_READ

This event is triggered by reads caused by a translation table walk. Fields:

ABORTED bool

Was the walk successful.

DATA unsigned int

The data read.

DESC_KIND enum

The encoding scheme of the descriptors used for this walk.

IPA unsigned int

For stage 1, the IPA of the read.

LEVEL unsigned int

Translation table level.

LPAE bool

Is this for an LPAE translation (*DEPRECATED*, use DESC_KIND).

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSmem bool

Non secure memory access.

NSreq bool

Non secure request.

OUTADDR unsigned int

The output physical address.

PADDR unsigned int

The physical address of the read.

PASmem enum

Physical Address Space of the accessed memory.

PASreq enum

Physical Address Space of the requested lookup.

REGIME_EL enum

Entry matches in this translation regime.

REQUESTER enum

The requester of this table walk.

SIDE enum

Inst / Data.

STAGE unsigned int

Translation stage.

MMU_TTB_WRITE

This event is triggered by writes caused by a translation table walk. Fields:

ABORTED bool

Was the walk successful.

DATA unsigned int

The data written.

DESC_KIND enum

The encoding scheme of the descriptors used for this walk.

IPA unsigned int

For stage 1, the IPA of the write.

IS_HACDBS_ACCESS bool

If the access was due to HACDBS_CLEAN.

LEVEL unsigned int

Translation table level.

LPAE bool

Is this for an LPAE translation (*DEPRECATED*, use DESC_KIND).

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSmem bool

Non secure memory access.

NSreq bool

Non secure request.

OUTADDR unsigned int

The output physical address.

PADDR unsigned int

The physical address of the write.

PASmem enum

Physical Address Space of the accessed memory.

PASreq enum

Physical Address Space of the requested lookup.

REGIME_EL enum

Entry matches in this translation regime.

SIDE enum

Inst / Data.

STAGE unsigned int

Translation stage.

MODE_CHANGE

Mode change. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

MODE enum

The new mode.

NON_SECURE enum

The core's new non-secure bit.

OLD_MODE enum

The old mode.

SECURITY_STATE enum

The new security state.

NOT_TAG_CHECKED_INST

MTE Tag not checked for current instruction.

PERIODIC

Called for every quantum. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The instruction count of this CPU.

PC unsigned int

The address of the next instruction to be executed on this CPU.

PMU_COUNTER_OVERFLOW

PMU counter overflow. Fields:

COUNTER_GROUP enum

Counter group.

COUNTER_RANGE enum

Counter range.

EVENT_ID unsigned int

Event Identifier, field is valid only for Event counter group.

INDEX unsigned int

Counter index (as selected by PMSELR), field is valid only for Event/Cycle counter groups.

INTERRUPT bool

Is interrupt enabled on overflow for this counter.

POINTER_AUTH

Called when execution hits a pointer authentication check. Fields:

AUTH_STATE bool

Authentication result.

KEY_USED enum

Pointer authentication key.

POTINDEX_UPDATE

Value of POT index for current pc changed. Fields:

NEW_POTINDEX unsigned int

New POTIndex.

OLD_POTINDEX unsigned int

Old POTIndex.

PREFETCH_MEMORY64

Prefetch from PRFM or PRFUM instructions. Fields:

PRFOP enum

Prefetch hint.

VADDR unsigned int

Virtual address of the location that should be prefetched.

PRELOAD64_REQUEST

Preload requests from PRFM or PRFUM instructions. Fields:

PRFOP enum

Prefetch hint.

VADDR unsigned int

Virtual address of the location that should be prefetched.

PRELOAD_DATA

Data preload from PLD instruction. Fields:

VADDR unsigned int

Virtual address of the data that should be preloaded.

PRELOAD_INST

Instruction preload from PLI instruction. Fields:

VADDR unsigned int

Virtual address of the instruction that should be preloaded.

PRE_CORE_LOAD

Trace just before a core load. Fields:

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

PRE_CORE_STORE

Trace just before a core store. Fields:

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

PRE_TTB_READ

This event is triggered before reads caused by a translation table walk. Fields:

ADDR unsigned int

The physical address of the read.

PSTATE

PSTATE fields update. Fields:

ALLINT unsigned int

All interrupt mask bit.

BTYP unsigned int

Branch target identification bit.

CORE_NUM unsigned int

Core number in a multi processor.

DAIF unsigned int

Exception mask bits.

DIT unsigned int

Data Independent Timing bit.

EL unsigned int

Current Exception level field.

EXLOCK unsigned int

Exception return state lock bit.

IL unsigned int

Illegal Execution state bit.

NZCV unsigned int

Condition flags.

PACM unsigned int

PACM bit.

PAN unsigned int

Privileged Access Never state bit.

PM unsigned int

PMU exception mask bit.

PPEND unsigned int

PMU exception pending bit.

SM unsigned int

StreamingMode status bit.

SP unsigned int

Stack pointer register selection bit.

SS unsigned int

Software Step bit.

SSBS unsigned int

Speculative Store Bypass Safe bit.

TCO unsigned int

Tag Check Override bit.

TOUCHED unsigned int

Mask of the fields that have been touched since this trace has last been emitted. Note that touched does not necessarily mean the value changed. The index of the field is the bit index in the mask plus the index of this field plus one.

UAO unsigned int

User Access Override bit.

UINJ unsigned int

Undefined Instruction exceptions bit.

ZA unsigned int

ZA storage and ZT0 register enable bit.

nRW unsigned int

Current Execution state bit.

PSTATE_PACTIVE_OPMODE_TRANSITION

Trace pactive and Opmode mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Operating mode after the transition.

OLD_STATE enum

Operating mode before the transition.

PSTATE_PACTIVE_OPMODE_TRANSITION_DENIED

Trace the denied pactive and Opmode mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Operating mode after the transition.

OLD_STATE enum

Operating mode before the transition.

REASON string

Reason for the transition being denied.

PSTATE_PACTIVE_POWER_TRANSITION

Trace pactive and power mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Power mode after the transition.

OLD_STATE enum

Power mode before the transition.

PSTATE_PACTIVE_POWER_TRANSITION_DENIED

Trace the denied pactive and power mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Power mode after the transition.

OLD_STATE enum

Power mode before the transition.

REASON string

Reason for the transition being denied.

PSTATE_PACTIVE_THREAD_OPMODE_TRANSITION

Trace pactive and Opmode mode transitions for threads on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Operating mode after the transition.

OLD_STATE enum

Operating mode before the transition.

QUANTUM_START

Called at the start of every quantum.

RANGE_PRELOAD_REQUEST

Preload requests from RPRFM. Fields:

COUNT unsigned int

Number of blocks of data to access.

LENGTH unsigned int

Number of contiguous bytes to be accessed in each block.

OP unsigned int

Range prefetch operation .

REUSE unsigned int

Maximum number of bytes to be accessed before next RPRFM .

STRIDE signed int

Stride to advance after accessing LENGTH bytes.

VADDR unsigned int

Base virtual address of range prefetch request.

RUN_STATE

Run state transition. Fields:

INST_COUNT unsigned int

Ticks count at point of transition.

NEW enum

New run state.

OLD enum

Old run state.

SBP_SAMPLE_COLLISION

DISPLAY Statistical profiling samples collided .

SBP_SAMPLE_DISCARDED

DISPLAY Operation discarded by statistical profiling unit when configured in discard mode.

Fields:

BUFFER_POINTER unsigned int

Value of buffer pointer.

DATA unsigned int

Data discarded.

SIZE unsigned int

Size of sample, in bytes.

SBP_SAMPLE_END

DISPLAY Statistical profiling of an operation ended.

SBP_SAMPLE_POP

DISPLAY Operation which might be sampled by statistical profiling.

SBP_SAMPLE_RECORDED

DISPLAY Operation sampled by statistical profiling has been written to the buffer at %{BUFFER_ADDRESS} with size %{SIZE} and data %{DATA}. Fields:

BUFFER_ADDRESS unsigned int

Virtual Address of start of buffer write.

DATA unsigned int

Data written to the buffer.

SIZE unsigned int

Size of sample written to buffer, in bytes.

SBP_SAMPLE_START

DISPLAY Statistical profiling of an operation started.

SBP_SAMPLE_TAKEN

DISPLAY Operation sampled by statistical profiling after filtering.

SEMIHOSTING_CALL

Call of a semihost function occurred. Fields:

PC unsigned int

The program counter after the semihosting call.

SEMIHOSTING_PRECALL

About to call semihost. Fields:

EL2 **bool**

Use EL2 translation regime.

NS **bool**

Is Non-Secure.

REG_WIDTH **unsigned int**

The current register width in bytes.

SIGNAL

External signal state change. Fields:

SIGNAL **enum**

Signal that changed.

STATE **bool**

Signal asserted state.

SOFTWARE_STEP

Return the Debug Software Step state. Fields:

STATE **enum**

Software Step state.

SPSR

SPSR change. Fields:

CORE_NUM **unsigned int**

Core number in a multi processor.

MODE **enum**

Which of the banked SPSR registers is written.

OLD_VALUE **unsigned int**

The old SPSR value.

VALUE **unsigned int**

The new SPSR value.

START_COMPILE

Compilation started. Fields:

VADDR **unsigned int**

Instruction where compilation begins.

STG_MULTIPLE_MICRO_OPS_ADDRESS_RANGE

Address range for MTE stores with multiple uops. Fields:

ADDR_END **unsigned int**

Final address.

ADDR_START unsigned int

Starting address.

STORE_MULTIPLE_REGS_ADDRESS_RANGE

Address range for stores with multiple registers. Fields:

ADDR_END unsigned int

Final address.

ADDR_START unsigned int

Starting address.

SUBOPTIMAL_LDST_RETIRED

suboptimal load/store. Fields:

ACCESS_SIZE unsigned int

Log2 of the access width in bytes used for alignment checking (1: half-word, 2: word ...).

OPTIMAL_ALIGN_SELECTOR unsigned int

Selector to decide the optimal alignment size (1: access width, 2: 4-bytes, 3: 8-bytes, ..., 12: 4-kbytes).

VADDR unsigned int

The virtual address of the access.

SYNC

Called for every synchronization, that is every quantum break. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The instruction count of this CPU.

LOCAL_QUANTUM unsigned int

The local quantum of this CPU.

LOCAL_TIME unsigned int

The local time of this CPU.

SYNC_REASON enum

Reason for sync.

SYSCALL

System call instruction executed. Fields:

IMM unsigned int

Immediate value of the system call instruction.

TYPE enum

System call type.

VADDR unsigned int

Instruction that caused the system call.

SYSREG_READ128

System Coprocessor 128 bit register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

REGNUM enum

Internal register number.

REGNUM_PHYS enum

Internal register number physically accessed (e.g. in presense of VHE redirection).

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_READ64

System Coprocessor register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REGNUM enum

Internal register number.

REGNUM_OPERAND enum

Register number corresponding to the instruction operand.

REGNUM_PHYS enum

Internal register number physically accessed (e.g. in presense of VHE redirection).

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_UPDATE128

Triggers when the system updates a register. Fields:

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG enum

Register number.

UNKNOWN unsigned int

Bits of the register which became unknown.

VALUE unsigned int

Value written to the register.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_UPDATE32

Triggers when the system updates a register. Fields:

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG enum

Register number.

UNKNOWN unsigned int

Bits of the register which became unknown.

VALUE unsigned int

Value written to the register.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_UPDATE64

Triggers when the system updates a register. Fields:

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG enum

Register number.

UNKNOWN unsigned int

Bits of the register which became unknown.

VALUE unsigned int

Value written to the register.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int
opcode 2.

SYSREG_WRITE128

System Coprocessor 128 bit register write. Fields:

CORE_NUM unsigned int
Core number in a multi processor.

CRm unsigned int
CRm.

CRn unsigned int
CRn.

REGNUM enum
Internal register number.

REGNUM_PHYS enum
Internal register number physically accessed (e.g. in presense of VHE redirection).

UNDEF bool
The register accessed is undefined.

UPDATED_VALUE unsigned int
Updated value of the register now it has been written.

VALUE unsigned int
The new value written.

opc unsigned int
opcode 1.

opc0 unsigned int
opcode 0.

opc2 unsigned int
opcode 2.

SYSREG_WRITE64

System Coprocessor register write. Fields:

CORE_NUM unsigned int
Core number in a multi processor.

CRm unsigned int
CRm.

CRn unsigned int
CRn.

NS bool
Secure or nonsecure banked register is accessed.

REGNUM enum

Internal register number.

REGNUM_OPERAND enum

Register number corresponding to the instruction operand.

REGNUM_PHYS enum

Internal register number physically accessed (e.g. in presense of VHE redirection).

UNDEF bool

The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

UNALIGNED_LDST_RETIRED

Processor unaligned load/store. Fields:

ACCESS_SIZE unsigned int

Log2 of the access width in bytes used for alignment checking (1=>half-word, 2->word ...).

NUMBER_OF_BEATS unsigned int

Number of accesses (beats) in this burst (Total size in bytes is ACCESS_SIZE * NUMBER_OF_BEATS).

VADDR unsigned int

The virtual address of the access.

VFP_D_REGS

VFP/NEON D 64 bit register write. Fields:

ALIASING enum

Type of register aliasing used.

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a write to the corresponding bit in the register. For example, 0x1 would indicate a write to VALUE[0].

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

VFP_D_REGS_READ

VFP/NEON D 64 bit register read. Fields:

ID unsigned int

The register number.

VALUE unsigned int

The current value of the register.

VFP_Q_REGS

VFP/NEON Q 128 bit register write. Fields:

ALIASING enum

Type of register aliasing used.

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a write to the corresponding byte in the register. For example, 0x1 would indicate a write to VALUE[7:0].

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

VFP_Q_REGS_READ

VFP/NEON Q 128 bit register read. Fields:

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a read to the corresponding byte in the register. For example, 0x1 would indicate a read to VALUE[7:0].

VALUE unsigned int

The current value of the register.

VFP_SYS_REGS

Writes to the VFP/NEON units system registers. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ID enum

Which VFP system register is written.

MASK unsigned int

The mask indicating updated cumulative bits.

OLD_VALUE unsigned int

The register's old value overwritten.

VALUE unsigned int

The new value written to the VFP system register.

VFP_S_REGS

VFP/NEON S 32 bit register write. Fields:

ALIASING enum

Type of register aliasing used.

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a write to the corresponding bit in the register. For example, 0x1 would indicate a write to VALUE[0].

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

VFP_S_REGS_READ

VFP/NEON S 32 bit register read. Fields:

ID unsigned int

The register number.

VALUE unsigned int

The current value of the register.

VIRTUAL_TAG_ALLOCATED

Virtual tag information when allocated. Fields:

TAG_VADDR unsigned int

Virtual tag virtual address.

VADDR unsigned int

Data Virtual address.

VIRTUAL_TAG unsigned int

Virtual tag value.

WAYPOINT

Signals end of basic block execution. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ISET enum

Origin instruction set.

IS_COND bool

Indicates if this is a conditional waypoint.

PC unsigned int

Origin address (or 0 if unavailable).

TAKEN bool

Indicates if this waypoint was taken.

TARGET unsigned int

Destination address.

TARGET_ISET enum

Destination instruction set.

WFE_END

WFE ended. Fields:

INST_COUNT unsigned int

Ticks count when leaving WFE.

WFE_EVENT_REGISTER

WFE event register status: set/clear, reason. Fields:

INST_COUNT unsigned int

Ticks count.

REASON enum

Reason for set/clear. Only REASON==0 (Cleared by WFE) clears the bit, all other reasons set it.

WFE_IGNORED

WFE ignored. Fields:

EVENT bool

This WFE was ignored because the event register was set.

INST_COUNT unsigned int

Ticks count when ignoring WFE.

TRAPPED bool

This WFE was trapped.

WFE_START

WFE entered. Fields:

INST_COUNT unsigned int

Ticks count when entering WFE.

WFI_END

WFI ended. Fields:

INST_COUNT unsigned int

Ticks count when leaving WFI.

WFI_IGNORED

WFI ignored. Fields:

DISABLED bool

This WFI was ignored because WFI is disabled.

INST_COUNT unsigned int

Ticks count when ignoring WFI.

REASON enum

specifies reason why WFI trace was ignored.

TRAPPED bool

This WFI was trapped.

WFI_START

WFI entered. Fields:

INST_COUNT unsigned int

Ticks count when entering WFI.

WFI_WAKEUP

WFI wakeup. Fields:

INST_COUNT unsigned int

Ticks count when WFI wakeup occurred.

REASON enum

Reason for wakeup.

2.37 ARM_Cortex-A78C

This section describes the trace sources.

AA64_ASE_SVE_REGS

Modification of AArch64 Advanced SIMD or SVE Vector Registers (via SIMD or Scalar operations), SVE Predicate Registers and ZT0. Fields:

ID `enum`

The register identifier.

MASK `unsigned int`

Bitmask of the register to signal the modified bits in the VALUE field.

SM `bool`

Whether the PE is in Streaming Mode.

VALUE `unsigned int`

Value of AArch64 vector register.

ASYNC_MEMORY_FAULT

Context ID Register write. Fields:

FAULT `unsigned int`

Fault status present in ESR.ISS encoding for Serror in ARCH64 and present in DFSR encoding for external abort format in ARCH32.

PADDR `unsigned int`

Physical Address (or 0 if unavailable).

VADDR `signed int`

Virtual Address (or 0 if unavailable).

ATOMIC_END_ACCESS

Bus trace access for atomics. Fields:

ACCESS_FAIL `bool`

Memory access failed.

ADDR `unsigned int`

The virtual address of the access.

ATTR `unsigned int`

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

COMPARE_VALUE `unsigned int`

Compare value for CAS.

LOAD_VALUE unsigned int

Loaded values.

NSDESC unsigned int

The security state of the access.

OPERAND_VALUE unsigned int

Operation's operand.

OPERATION enum

Operation type.

PADDR unsigned int

The physical address of the access.

PRIV bool

Is this a privileged access?.

ATOMIC_START_ACCESS

Bus trace access for atomics. Fields:

ADDR unsigned int

The virtual address of the access.

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

COMPARE_VALUE unsigned int

Compare value for CAS.

NSDESC unsigned int

The security state of the access.

OPERAND_VALUE unsigned int

Operation's operand.

OPERATION enum

Operation type.

PADDR unsigned int

The physical address of the access.

PRIV bool

Is this a privileged access?.

ArchMsg.Error.exit_code

DISPLAY exit code: %{EXITCODE} component kind: %{KIND} reason: %{REASON}. Fields:

EXITCODE signed int

Exit code to be returned.

KIND string

Component kind that invoked the exit code trace.

REASON string

Reason why the exit code trace was invoked (optional).

ArchMsg.Warning.IgnoredWarmResetRequest

Event fires when software attempts to request warm reset through write to deprecated DBGPRCR.CWRR.

ArchMsg.Warning.RAS_ErrselrUnpredictableState

DISPLAY ERRSELR is in unpredictable state due to value greater than number_of_error_records written to it.

ArchMsg.Warning.RAS_UnpredictableStateWrite

DISPLAY %{REG_NAME} write ignored due to unpredictable ERRSELR state. Fields:

REG_NAME string

Name of register being written.

ArchMsg.Warning.ReportedRevisionRegisterRead

DISPLAY Reported revision and patch do not match that of the cpu. Fields:

ORIGINAL_VALUE unsigned int

The value the register would have based on the modelled revision.

REGISTER string

The name of the register that has ben read.

REPORTED_VALUE unsigned int

The value the register has been overridden to.

ArchMsg.Warning.Unpredictably Indexed PM Event Register

TAGS UNPREDICTABLE DISPLAY Event %{IsDirect:(Selector value|Register)} %{SEL} is >= %{N}. Fields:

IsDirect unsigned int

Direct Access rather than indirect.

N unsigned int

Number of Accessible Registers.

SEL unsigned int

Selector.

ArchMsg.Warning.cache_contents_unknown

A cache has been activated without being invalidated and may contain **UNKNOWN** content. DISPLAY %{SIDE:(D|I)}-Cache was enabled in %{REGIME:(SECURE|NON_SECURE|HYP)} regime but was not invalidated since power-on: cache lines could contain **UNKNOWN** content %{SIDE:(Note: D-side caches must invalidate SECURE lines even if the cache is only used by NON_SECURE code in order to ensure that unexpected dirty lines tagged SECURE are not naturally evicted)}. Fields:

REGIME unsigned int

Current translation regime.

SIDE unsigned int

0 d-side cache or unified cache, 1 i-side.

ArchMsg.Warning.dcimvac_matches_watchpoint

TAGS **UNPREDICTABLE** DISPLAY Watchpoints matching an AArch32 DCIMVA are implementation defined.

ArchMsg.Warning.decode_fieldmismatch

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with different fields %{FIELD1} and %{FIELD2}. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_fracbitsnegative

TAGS **UNPREDICTABLE** DISPLAY VCVT Instruction is unpredictable with a negative number of fraction bits.

ArchMsg.Warning.decode_initblock

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable in an IT block.

ArchMsg.Warning.decode_initblocknotlast

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable in an IT block when not the last.

ArchMsg.Warning.decode_invalid_condition

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable when the condition is not AL.

ArchMsg.Warning.decode_invalidfieldcombination

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with the values in %{FIELD1} and %{FIELD2}. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_invalidvalue

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with the value in %{FIELD}. Fields:

FIELD enum

Field name.

ArchMsg.Warning.decode_mem_hint_unallocated

TAGS **UNPREDICTABLE** DISPLAY There is no memory hint allocated to this bit pattern.

ArchMsg.Warning.decode_registermatch

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with %{FIELD1} and %{FIELD2} as the same register. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_registermismatch

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with %{FIELD1} and %{FIELD2} as different registers. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_registeroutofrange

TAGS **UNPREDICTABLE** DISPLAY Floating-point instruction is unpredictable indexing beyond the end of the register bank.

ArchMsg.Warning.decode_sbzsbo

TAGS **UNPREDICTABLE** DISPLAY Reserved bits in the instruction are not canonical.

ArchMsg.Warning.decode_transactiontoobig

TAGS **UNPREDICTABLE** DISPLAY Attempt to perform a memory transaction of more than 128 bytes.

ArchMsg.Warning.decode_unpred_other

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable.

ArchMsg.Warning.decode_unpreduseofpc

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable when %{FIELD} is PC. Fields:

FIELD enum

Field name.

ArchMsg.Warning.decode_unpreduseofr13

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable when %{FIELD} is R13. Fields:

FIELD enum

Field name.

ArchMsg.Warning.decode_unpreduseofr13inreglist

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with R13 in register-list.

ArchMsg.Warning.decode_useofd16orhigher

TAGS **UNPREDICTABLE** DISPLAY Attempt to use register D16 or higher where it is not allowed.

ArchMsg.Warning.decode_writebackandbaseinlist

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with write back when the base register is in the transfer list.

ArchMsg.Warning.decode_zeroregistersinlist

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable when no registers are to be transferred.

ArchMsg.Warning.recursive_branch

An instruction is performing a branch that targets the same instruction. If this was intended then a WFI might be more effective DISPLAY recursive branch detected at PC=%{PC}. PRIMARY KEY PC. Fields:

PC unsigned int

Address of instruction causing the branch.

ArchMsg.Warning.recursive_exception

An instruction has generated an exception that targets the same instruction. DISPLAY recursive %{TYPE} exception detected at PC=%{PC} (CPSR=%{CPSR}). PRIMARY KEY PC. Fields:

CPSR unsigned int

Processor state.

PC unsigned int

Address of instruction causing the exception.

TYPE enum

Type of exception.

ArchMsg.Warning.reentrant_vector_catch

TAGS **UNPREDICTABLE** DISPLAY Vector catch debug event on Prefetch or Data abort vector %{ADDR}. Fields:

ADDR unsigned int

vector address.

ArchMsg.Warning.reserved_it_state

TAGS **UNPREDICTABLE** DISPLAY Execution with reserved IT state %{ITSTATE}. Fields:

ITSTATE unsigned int

reserved IT state.

ArchMsg.Warning.tlb_contents_unknown

Translation has been enabled without invalidating the TLB. DISPLAY Translation was enabled in the %{PAS:(Secure|Non-secure|Root|Realm)} EL%{EL:d} translation regime (vmid=%{VMID:x}) without invalidating %{INVALIDITY:(|d-side |i-side |)}entries since power-on: %{INVALIDITY:(|D|I|)}TLB could contain **UNKNOWN** content. Fields:

EL unsigned int

Exception level of translation regime.

INVALIDITY unsigned int

Which TLB may be invalid (if TLBs are separate).

NON_SECURE bool

Whether the translation regime is non-secure.

PAS unsigned int

Physical address space of translation regime.

VMID unsigned int

Current VMID, if applicable.

ArchMsg.Warning.unknown_DLR_DSPSR

Exiting debug state with unknown DLR or DSPSR.

ArchMsg.Warning.unknown_ELR_SPSR

Returning from debug exception with unknown ELR or SPSR.

ArchMsg.Warning.unpred_FAR_on_LRCPC3_non_atomic_mem_access

TAGS **UNPREDICTABLE** Unpredictable FAR/FSTATUS on LRCPC3 pair instructions resulting in two separate memory accesses
DISPLAY Unpredictable FAR on LRCPC3 non atomic memory access at %{ADDR1} with FSTATUS %{FSTATUS1} or %{ADDR2} with FSTATUS %{FSTATUS2}. Fields:

ADDR1 unsigned int

Possible FAR address 1.

ADDR2 unsigned int

Possible FAR address 2.

FSTATUS1 unsigned int

Possible fault status value 1.

FSTATUS2 unsigned int

Possible fault status value 2.

ArchMsg.Warning.unpred_async_tag_check_fail_on_sve_speculative_load_do_update_ffr

Unpredictable set to FALSE of the FFR element associated to SVE speculative load with an async tag check fail TAGS **UNPREDICTABLE**.

ArchMsg.Warning.unpredictable_a32_breakpoint

TAGS **UNPREDICTABLE** DISPLAY Unpredictable breakpoint on a A32 instruction at %{ADDR1} Offset %{OFFSET} BAS %{BAS} for a %{IS_ADDRESS_MISMATCH_BREAKPOINT:(Mismatch|Match)} Breakpoint number %{N:d} Breakpoint address %{ADDR2}. Fields:

ADDR1 unsigned int

Address.

ADDR2 unsigned int

Breakpoint address.

BAS unsigned int

BAS field.

IS_ADDRESS_MISMATCH_BREAKPOINT bool

Is a Mismatch breakpoint, not a Match breakpoint.

N unsigned int

Breakpoint number.

OFFSET unsigned int

Always has a value of zero.

ArchMsg.Warning.unpredictable_banked_register_access

TAGS **UNPREDICTABLE** DISPLAY Access to banked register (R=%{R:d} SYSm=%{SYSm:2x}) is unpredictable. Fields:

R unsigned int

R bit.

SYSm unsigned int

SYSm.

ArchMsg.Warning.unpredictable_ccfail_undef

TAGS **UNPREDICTABLE** DISPLAY An instruction %{OPCODE} at VA=%{VADDR} failed its condition codes check but would otherwise have caused a non-data-dependent trap or undefined exception. Behaviour is implementation-defined choice of exception or nop. Fields:

OPCODE unsigned int

opcode of the instruction.

VADDR unsigned int

Virtual address of the instruction being executed.

ArchMsg.Warning.unpredictable_ici_bits

TAGS **UNPREDICTABLE** DISPLAY ICI bits do not apply or are architecturally unpredictable for instruction.

ArchMsg.Warning.unpredictable_t32_breakpoint

TAGS **UNPREDICTABLE** DISPLAY Unpredictable breakpoint on a T32 instruction at %{ADDR1} Offset %{OFFSET} BAS %{BAS} for a %{IS_ADDRESS_MISMATCH_BREAKPOINT:(Mismatch|Match)} Breakpoint number %{N:d} Breakpoint address %{ADDR2}. Fields:

ADDR1 unsigned int

Address.

ADDR2 unsigned int

Breakpoint address.

BAS unsigned int

BAS field.

IS_ADDRESS_MISMATCH_BREAKPOINT bool

Is a Mismatch breakpoint, not a Match breakpoint.

N unsigned int

Breakpoint number.

OFFSET unsigned int

Indicates which halfword of a 32-bit Thumb instruction the breakpoint is set on. 0 for the first halfword or if the instruction is 16 bits wide, or 1 for the second halfword.

ArchMsg.Warning.unpredictable_t32_vector_catch

TAGS **UNPREDICTABLE** DISPLAY Unpredictable Vector catch exception on a T32 instruction at %{ADDR} Offset %{OFFSET}. Fields:

ADDR unsigned int

Address.

OFFSET unsigned int

Indicates which halfword of a 32-bit Thumb instruction the vector catch exception is set on. 0 for the first halfword or if the instruction is 16 bits wide, or 1 for the second halfword.

ArchMsg.Warning.unpredictable_watchpoint_far

TAGS **UNPREDICTABLE** DISPLAY The value for FAR/EDWAR for hitting this watchpoint may be anywhere in the range %{LOWER_BOUND} to %{UPPER_BOUND}. Fields:

LOWER_BOUND unsigned int

the lowest address accessed by the instruction that triggered the watchpoint.

UPPER_BOUND unsigned int

the highest watchpointed address accessed by that instruction.

ArchMsg.Warning.unpredictable_watchpoint_far_isscmp

DISPLAY The value for FAR/EDWAR for hitting this watchpoint may be anywhere in the range %{LOWER_BOUND} to %{UPPER_BOUND}. Fields:

LOWER_BOUND unsigned int

the lowest address accessed by the instruction that triggered the watchpoint.

UPPER_BOUND unsigned int

the highest watchpointed address accessed by that instruction.

ArchMsg.Warning.warning_AdvSIMDEExpandImmUnexpectedZero

TAGS **UNPREDICTABLE** DISPLAY AdvSIMDEExpandImm may treat this immediate value as **UNPREDICTABLE**.

ArchMsg.Warning.warning_ConditionalSMC

TAGS **UNPREDICTABLE** DISPLAY SMC instruction has **UNPREDICTABLE** effects when conditional and when combined with traps.

ArchMsg.Warning.warning_TLBID_out_of_range

TLBID out of range in TLBI according to the configuration of NIS, NVIS, NOS and NVOS. Fields:

NIS unsigned int

NIS.

NOS unsigned int

NOS.

NVIS unsigned int

NVIS.

NVOS unsigned int

NVOS.

TLBID unsigned int

The TLBID value used in the TLBI instruction.

ArchMsg.Warning.warning_TTL_or_TG_mismatches_current_translation_granule

TTL or TG mismatches current translation granule in TLBI VA/RVA/IPA/RIPA. Fields:

CURRENT_TG unsigned int

Current TG as indicated by TCR_ELx or VTCR_EL2.

TLBI_TG unsigned int

TG (or TTL[3:2]) used in TLBI instruction.

ArchMsg.Warning.warning_access_crosses_page_boundary

TAGS **UNPREDICTABLE** DISPLAY Access crosses page boundary.

ArchMsg.Warning.warning_access_crosses_page_boundary_has_fault

TAGS **UNPREDICTABLE** DISPLAY Access that crosses page boundary has fault(s).

ArchMsg.Warning.warning_access_crosses_page_boundary_has_fault_on_both_pages

TAGS **UNPREDICTABLE** DISPLAY Access that crosses page boundary has a fault on each page. Fields:

ADDR1 unsigned int

FAR address on first page.

ADDR2 unsigned int

FAR address on second page.

FSTATUS1 unsigned int

Fault status on first page.

FSTATUS2 unsigned int

Fault status on second page.

ArchMsg.Warning.warning_access_crosses_page_boundary_spanning_different_memory

TAGS **UNPREDICTABLE** DISPLAY Access at address %{ADDR} crosses page from MEM_TYPE %{MEMTYPE_PAGE1} to %{MEMTYPE_PAGE2}. Fields:

ADDR unsigned int

address of access crossing page.

MEMTYPE_PAGE1 signed int

First Page's memory type (0-Normal, 1-Non-Normal).

MEMTYPE_PAGE2 signed int

Second Page's memory type (0-Normal, 1-Non-Normal).

ArchMsg.Warning.warning_access_wraps_around_memory

TAGS **UNPREDICTABLE** DISPLAY Access wraps around memory in %{WIDTH:d}bit mode.

Fields:

WIDTH unsigned int

Address width.

ArchMsg.Warning.warning_bcr_linking_status

TAGS **UNPREDICTABLE** DISPLAY DBGBCR.{BT2,BT} combination gives unpredictable behaviour.

ArchMsg.Warning.warning_bcr_mask_reserved

TAGS **UNPREDICTABLE** DISPLAY DBGBCR_ELn.MASK is set to a reserved value.

ArchMsg.Warning.warning_bvr_masked_bit_not_zero

TAGS **UNPREDICTABLE** DISPLAY DBGBVRn_EL1.VA contains a bit masked by DBGBCRn_EL1.MASK that is not zero.

ArchMsg.Warning.warning_ccsidr_unimplemented_level

TAGS **UNPREDICTABLE** DISPLAY CCSIDR read while CSSELR %{CSSELR:x} points to an unimplemented cache level. Fields:

CSSELR unsigned int

current effective value of CSSELR.

ArchMsg.Warning.warning_change_to_ns_when_tge_set

TAGS **UNPREDICTABLE** DISPLAY Attempting to change NS to 1 when HCR.TGE = 1.

ArchMsg.Warning.warning_contiguous_bit_check_abort

TAGS **UNPREDICTABLE** DISPLAY An abort occurred whilst attempting to check TLB entry is contiguous at %{ENTRY_ADDR}. Fields:

ENTRY_ADDR unsigned int

address of conflicting TLB entry.

ArchMsg.Warning.warning_contiguous_bit_error

TAGS **UNPREDICTABLE** DISPLAY TLB entry at %{ENTRY_ADDR} was not contiguous with entry at %{CONTIG_ADDR} Contents are %{ENTRY_DATA} but expected %{CONTIG_DATA}. Fields:

CONTIG_ADDR unsigned int

address of the TLB first read, with which this entry is expected to be contiguous.

CONTIG_DATA unsigned int

expected contents based on the entry at CONTIG_ADDR.

ENTRY_ADDR unsigned int

address of conflicting TLB entry.

ENTRY_DATA unsigned int

contents of conflicting TLB entry.

ArchMsg.Warning.warning_contiguous_bit_error_D128

TAGS **UNPREDICTABLE** DISPLAY TLB entry at %{ENTRY_ADDR} was not contiguous with entry at %{CONTIG_ADDR} Contents are %{ENTRY_DATA_HI} %{ENTRY_DATA_LO} but expected %{CONTIG_DATA}. Fields:

CONTIG_ADDR unsigned int

address of the TLB first read, with which this entry is expected to be contiguous.

CONTIG_DATA unsigned int

expected contents based on the entry at CONTIG_ADDR.

ENTRY_ADDR unsigned int

address of conflicting TLB entry.

ENTRY_DATA_HI unsigned int

Higher bit contents of conflicting TLB entry.

ENTRY_DATA_LO unsigned int

Lower bit contents of conflicting TLB entry.

ArchMsg.Warning.warning_cp10_cp11_mismatch

TAGS **UNPREDICTABLE** DISPLAY Unpredictable configuration of CP10 and CP11 controls at EL %{EL:d}. Fields:

EL unsigned int

EL owning the control.

ArchMsg.Warning.warning_cp10_cp11_reserved_value

TAGS **UNPREDICTABLE** DISPLAY Unpredictable configuration of CPACR CP10 (%{CP10}) or CP11 (%{CP11}). Fields:

CP10 unsigned int

CP10 access.

CP11 unsigned int

CP11 access.

ArchMsg.Warning.warning_csselr_level_out_of_range

TAGS **UNPREDICTABLE** DISPLAY CSSELR.Level out of range, written value %{LEVEL} with only %{IMPLEMENTED:d} levels implemented. Fields:

IMPLEMENTED unsigned int

number of cache levels implemented.

LEVEL unsigned int

written value of CSSELR.Level.

TYPE enum

The type of cache selected by CSSELR.InD and TnD.

ArchMsg.Warning.warning_debug_flow_control_bits_not_obeyed

TAGS **UNPREDICTABLE** DISPLAY Flow control bits for %{REGISTER} were not obeyed, causing %{ERROR}. Fields:

ERROR enum

type of error resulting.

REGISTER enum

register under consideration.

ArchMsg.Warning.warning_debug_register_access_during_reset

TAGS **UNPREDICTABLE** DISPLAY Unpredictable %{IS_WRITE:(read|write)} access to debug register offset %{OFFSET} during reset. Fields:

IS_WRITE unsigned int

Write Not Read.

OFFSET unsigned int

Register Offset.

ArchMsg.Warning.warning_decode_cps_inconsistent_fields

TAGS **UNPREDICTABLE** DISPLAY The mode setting fields in the CPS instruction are inconsistent.

ArchMsg.Warning.warning_decode_invalid_state

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable in current exception-level/security state.

ArchMsg.Warning.warning_default_cacheable_mmu_on

TAGS **UNPREDICTABLE** DISPLAY Default cacheable is enabled (HCR.DC = 1) while MMU is enabled.

ArchMsg.Warning.warning_default_cacheable_vmmu_off

TAGS **UNPREDICTABLE** DISPLAY Default cacheable is enabled (HCR.DC = 1) while HCR.VM = 0.

ArchMsg.Warning.warning_deprecated_wvr_bit_2_set

TAGS **UNPREDICTABLE** DISPLAY DBGWVR%{N:d}_EL1=%{WVR:h} has bit 2 set, deprecated in ARMv8. Fields:

N unsigned int

index.

WVR unsigned int

DBGWVR.

ArchMsg.Warning.warning_eret_while_software_step_active_pending

Missing ISB between setting MDSCR_EL1.SS and ERET TAGS **UNPREDICTABLE**.

ArchMsg.Warning.warning_exclusive_to_non_normal

TAGS **UNPREDICTABLE** DISPLAY Exclusive Access to Non-Normal Memory %{ADDR}. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_exclusive_to_non_writeback

TAGS **UNPREDICTABLE** DISPLAY Exclusive Access to Non-Writeback-Cacheable Memory
%{ADDR}. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_execute_from_device_memory

TAGS **UNPREDICTABLE** DISPLAY An attempt was made to execute from Device memory.

ArchMsg.Warning.warning_illegal_cpsr_mode

TAGS **UNPREDICTABLE** DISPLAY Writing an illegal or unimplement mode (%{NEW_MODE:h})
to CPSR was unpredictable. Fields:

NEW_MODE unsigned int

New value of CPSR.M.

ArchMsg.Warning.warning_illegal_srs_mode

TAGS **UNPREDICTABLE** DISPLAY Illegal or **UNPREDICTABLE** mode (%{MODE:h}) used for SRS
instruction. Fields:

MODE unsigned int

mode for Banked SP.

ArchMsg.Warning.warning_implementation_defined_read_debug_register_in_SCS

Read Access to Debug registers from the processor is IMPLEMENTATION DEFINED TAGS
IMP_DEF. Fields:

PPB_OFFSET unsigned int

debug register offset.

ArchMsg.Warning.warning_implementation_defined_sequential_security_transitions_supported

The behavior of sequential instruction fetches that cross from non-secure to secure memory
and contain an SG instruction is CONSTRAINED **UNPREDICTABLE** TAGS **UNPREDICTABLE**.

ArchMsg.Warning.warning_implementation_defined_stack_limit_check_supported

It is IMPLEMENTATION DEFINED whether stack pointer limit checking is performed for this
instruction TAGS IMP_DEF. Fields:

ADDRESS unsigned int

Address of instruction.

OPCODE unsigned int

instruction opcode.

ArchMsg.Warning.warning_implementation_defined_taggednonsharedmemorytype

TAGS IMP_DEF DISPLAY Memory attribute has Tagged, non shared memory type; it is
IMPLEMENTATION DEFINED whether the memory location is treated as Tagged or non-Tagged.

ArchMsg.Warning.warning_implementation_defined_write_debug_register_in_SCS

Write Access to Debug registers from the processor is IMPLEMENTATION DEFINED TAGS IMP_DEF. Fields:

DATA unsigned int

data attempted to be written.

PPB_OFFSET unsigned int

debug register offset.

ArchMsg.Warning.warning_invalid_tcr_granule

TAGS UNPREDICTABLE DISPLAY TCR.TG{%TG_ID:(0|1)} bits have been set to a granule size {%REQUEST:d}K not implemented - using {%SUBSTITUTE:d}K. Fields:

REQUEST signed int

page size requested (or 0 for reserved).

SUBSTITUTE signed int

best guess available page size.

TG_ID bool

bits TG0 or TG1.

ArchMsg.Warning.warning_load_multiple_user_registers_from_user_mode

TAGS UNPREDICTABLE DISPLAY An LDM(user registers) instruction executed from user mode.

ArchMsg.Warning.warning_mte_in_mem_attr_not_supported

TAGS UNPREDICTABLE DISPLAY Memory attribute {%ATTR} has Tagged memory type but Armv8.5-MTE is not implemented. Fields:

ATTR unsigned int

Memory attribute.

ArchMsg.Warning.warning_reserved_breakpoint_state_match

TAGS UNPREDICTABLE DISPLAY {%IS_BREAKPOINT:(Watchpoint|Breakpoint)} programmed with a reserved combination of HMC, SSC and {%IS_BREAKPOINT:(PAC|PMC)}. Fields:

IS_BREAKPOINT unsigned int

Is a breakpoint, not a watchpoint.

ArchMsg.Warning.warning_shareability

TAGS UNPREDICTABLE DISPLAY Unpredictable: combination of the 1st and 2nd stages of translation is Normal Inner Non-Cacheable, Outer Non-Cacheable.

ArchMsg.Warning.warning_software_step_set_while_enabled

TAGS UNPREDICTABLE DISPLAY MDSCR_EL1.SS set to 1 while software step debug exceptions are enabled.

ArchMsg.Warning.warning_thumb_instruction_wraps_around_memory

TAGS UNPREDICTABLE DISPLAY Thumb instruction wraps around memory.

ArchMsg.Warning.warning_ttbr_sbz_bits_are_not_zero

TAGS UNPREDICTABLE DISPLAY %{TTBR}[%{MSB:d}:0] should be zero, but are not. Fields:

MSB unsigned int

x-1.

TTBR enum

which TTBR register.

ArchMsg.Warning.warning_unaligned_address_dbgdttrrx_write

TAGS UNPREDICTABLE DISPLAY Unaligned address %{ADDR} in XO/RO when DBGDTTRRX written in memory access mode. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unaligned_address_dbgdttrtx_read

TAGS UNPREDICTABLE DISPLAY Unaligned address %{ADDR} in XO/RO when DBGDTTRTX read in memory access mode. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unaligned_to_device

TAGS UNPREDICTABLE DISPLAY Unaligned Access to Device Memory %{ADDR}. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unaligned_to_strongly_ordered

TAGS UNPREDICTABLE DISPLAY Unaligned Access to Strongly Ordered Memory %{ADDR}. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unknown_sau_rnr

TAGS UNPREDICTABLE DISPLAY SAU_RNR was set to an unsupported value.

ArchMsg.Warning.warning_unpredictable_AIRCR_PRIS_and_BFHFNMINs

TAGS UNPREDICTABLE DISPLAY The effect of setting both AIRCR.BFHFNMINs and AIRCR.PRIS to 1 is UNPREDICTABLE.

ArchMsg.Warning.warning_unpredictable_AIRCR_VECTCLRACTIVE_when_not_in_debug

TAGS UNPREDICTABLE DISPLAY The effect of writing a 1 to AIRCR.VECTCLRACTIVE if the processor is not halted in Debug state is UNPREDICTABLE.

ArchMsg.Warning.warning_unpredictable_AIRCR_VECTRESET_and_SYSRESETREQ

TAGS UNPREDICTABLE DISPLAY When the processor is halted in Debug state, if a write to the register writes a 1 to both VECTRESET and SYSRESETREQ, the behavior is UNPREDICTABLE.

ArchMsg.Warning.warning_unpredictable_AIRCR_VECTRESET_when_not_in_debug

TAGS **UNPREDICTABLE** DISPLAY The effect of writing a 1 to AIRCR.VECTRESET if the processor is not halted in Debug state is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_AIRCR_incorrect_VKEY

TAGS **UNPREDICTABLE** DISPLAY The value 0x05FA must be written to AIRCR.VKEY, otherwise the register write is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_EXC_RETURN_Reserved_Bit

EXC_RETURN[23:7] are reserved with the special condition that all bits should be written as one. Values other than all 1s are **UNPREDICTABLE**. TAGS **UNPREDICTABLE**. Fields:

EXC_RETURN unsigned int

Special PC value.

SBO_MASK unsigned int

Which bits Should Be One to not UNPRED.

ArchMsg.Warning.warning_unpredictable_PMEVTYPEPn_ELO_VS_reserved_value

TAGS **UNPREDICTABLE** DISPLAY Unpredictable reserved value of PMEVTYPEn<n>_ELO.VS.

ArchMsg.Warning.warning_unpredictable_change_to_DEMCR_MON_STEP_at_insufficient_priority

TAGS **UNPREDICTABLE** DISPLAY The effect of changing DEMCR.MON_STEP at an execution priority that is lower than the priority of the DebugMonitor exception is **CONSTRAINED UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_change_to_priority_of_active_exception

TAGS **UNPREDICTABLE** DISPLAY Changing the priority of an active exception is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_clear_lspact

TAGS **UNPREDICTABLE** DISPLAY **UNPREDICTABLE** state on context switch. Fields:

DRVEXC_TAKEN unsigned int

Derived exception taken, not pended.

REACHED_LAST_STORE unsigned int

Only the last faulted.

ArchMsg.Warning.warning_unpredictable_dbg_exit_unaligned_dlr

TAGS **UNPREDICTABLE** DISPLAY Debug exit to AArch32 with DLR[0] set to 1. PC[0] can be set to 0 or to DLR[0].

ArchMsg.Warning.warning_unpredictable_exception_catch

TAGS **UNPREDICTABLE** DISPLAY Generation of an exception catch event is unpredictable.

ArchMsg.Warning.warning_unpredictable_exception_return_inconsistent_state

TAGS **UNPREDICTABLE** DISPLAY state on exception return is unpredictable.

ArchMsg.Warning.warning_unpredictable_exception_return_instruction

TAGS **UNPREDICTABLE** DISPLAY exception return instruction is unpredictable.

ArchMsg.Warning.warning_unpredictable_in_debug_state

TAGS **UNPREDICTABLE** DISPLAY Instruction is **UNPREDICTABLE** when executed in debug state.

ArchMsg.Warning.warning_unpredictable_mair_encoding

TAGS **UNPREDICTABLE** DISPLAY Memory attribute %{ATTR} has invalid MAIR encoding; this may lead to unpredictable results. Fields:

ATTR unsigned int

Memory attribute.

ArchMsg.Warning.warning_unpredictable_pmu_counter_access

TAGS **UNPREDICTABLE** DISPLAY Access to PMU counters from non-secure EL0 or EL1 is **UNPREDICTABLE** with MDCR_EL2.HPMN set to 0.

ArchMsg.Warning.warning_unpredictable_prioritization_breakpoint_match_vector_catch

TAGS **UNPREDICTABLE** DISPLAY Two events with the same priority occurred at the same instruction: Address Matching Vector Catch debug event, and Breakpoint debug event. It is constrained unpredictable which is taken.

ArchMsg.Warning.warning_unpredictable_stack_selection

TAGS **UNPREDICTABLE** DISPLAY **UNPREDICTABLE** state on context switch.

ArchMsg.Warning.warning_unpredictable_tsize_out_of_range

TAGS **UNPREDICTABLE** DISPLAY %{TCR}.T%{TTBR:d}SZ (%{TSIZE:d} bits) is out of range. Must be between 25 and %{TSIZE_MAX:d}. Fields:

TCR enum

Which TCR.

TSIZE signed int

Value of TSize.

TSIZE_MAX signed int

The maximum allowed value.

TTBR unsigned int

Which TTBR region.

ArchMsg.Warning.warning_unpredictable_unaligned_pc_as_base_register

TAGS **UNPREDICTABLE** DISPLAY The PC value must be word-aligned, otherwise the behavior of the instruction is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_unaligned_pop_stack

TAGS **UNPREDICTABLE** DISPLAY **CONSTRAINED UNPREDICTABLE** unaligned stack frame %{FRAME_ADDR:x} in exception-return. Fields:

FRAME_ADDR unsigned int

unaligned stack frame address.

ArchMsg.Warning.warning_unpredictable_unaligned_to_non_writeback

TAGS **UNPREDICTABLE** DISPLAY Unaligned Access to Non-Writeback-Cacheable Memory.

ArchMsg.Warning.warning_unpredictable_vmsa_memattrib

TAGS **UNPREDICTABLE** DISPLAY Unpredictable vmsa memory attribute with texcb value programmed to %{TEXCB} when tex remap is %{TEX_REMAP}. Fields:

TEXCB unsigned int

texcb value.

TEX_REMAP bool

use tex remap.

ArchMsg.Warning.warning_unpredictable_vtcr_t0sz_sl0

TAGS UNPREDICTABLE DISPLAY The combination of VTCR.TOSZ=%{TOSZ:d} and VTCR.SL0=%{SL0:d} is unpredictable. Fields:

SL0 signed int

VTCR.SL0, starting level for stage 2 translation table walks.

TOSZ signed int

VTCR.TOSZ, required input address range.

ArchMsg.Warning.warning_unpredictable_write_DHCSR

TAGS UNPREDICTABLE DISPLAY UNPREDICTABLE set of bit-changes in DHCSR. Fields:

NEW unsigned int

data attempted to be written.

OLD unsigned int

previous value.

ArchMsg.Warning.warning_unpredictable_write_SHCSR

TAGS UNPREDICTABLE DISPLAY UNPREDICTABLE setting a pending bit to 1 for an exception with priority >= execution priority.

ArchMsg.Warning.warning_unsupported_access_to_memory_mapped_register

TAGS UNPREDICTABLE DISPLAY Access to memory mapped register is unsupported.

ArchMsg.Warning.warning_user_jmcr_access

TAGS UNPREDICTABLE DISPLAY User %{WRITE:(write|read)} access to JMCR is unpredictable. Fields:

WRITE unsigned int

access is write.

ArchMsg.Warning.warning_wcr_mask_and_bas

TAGS UNPREDICTABLE DISPLAY DBGWCR_EL1.MASK is non-zero and BAS is not 0xff.

ArchMsg.Warning.warning_wcr_mask_reserved

TAGS UNPREDICTABLE DISPLAY DBGWCR_ELn.MASK is set to a reserved value.

ArchMsg.Warning.warning_wcr_non_configuous_bas

TAGS UNPREDICTABLE DISPLAY DBGWCR_EL1.BAS has non-contiguous set of ones.

ArchMsg.Warning.warning_wvr_masked_bit_not_zero

TAGS UNPREDICTABLE DISPLAY DBGWVRn_EL1.VA contains a bit masked by DBGWCRn_EL1.MASK that is not zero.

ArchMsg.Why.why_branch_target_exception

A branch target exception has occurred. DISPLAY branch target exception: target pc=%{PC}, target opcode=%{OPCODE}. PRIMARY KEY PC. Fields:

BTTYPE unsigned int

pstate.BTYPE at time branch taken.

OPCODE unsigned int

opcode at branch target.

PC unsigned int

Virtual Address of branch target.

ArchMsg.Why.why_illegal_state

An illegal mode change has occurred. DISPLAY illegal mode change: %{MESSAGE}. Fields:

MESSAGE string

The message.

ArchMsg.warning_invalid_interval_counter_reload_value

Invalid interval counter reload value in PMSIRR_EL1.INTERVAL: software should set this to a value greater than or equal to the minimum indicated by PMSIDR_EL1.Interval.

ArchMsg.warning_invalid_spe_buffer_write_pointer

Invalid value programmed in PMBPTR_EL1.PTR or PMBLIMITR_EL1.LIMIT: Current write pointer must be at least one sample record below the write limit pointer and PMBPTR_EL1.PTR[63:56] must be equal to PMBLIMITR_EL1.LIMIT[63:56].

BRANCH_MISPREDICT

Simulating branch mispredict. Fields:

PC unsigned int

Origin address (or 0 if unavailable).

BRA_DIR

Direct branches, to immediate address. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISET enum

The instruction set of the branch instruction.

IS_COND bool

Indicates if this is a conditional branch.

IS_HINTED bool

Whether the conditional instruction is hinted.

IS_ISB bool

Whether the instruction is ISB.

IS_LINK bool

Indicates if this is a branch with link.

PC unsigned int

The address of the branch instruction.

TARGET_ISET enum

The instruction set after the branch.

TARGET_PC unsigned int

The address the instruction branches to.

BRA_INDIR

Indirect branches, perhaps to a register. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISET enum

The instruction set of the branch instruction.

IS_COND bool

Indicates if this is a conditional branch.

IS_HINTED bool

Whether the conditional instruction is hinted.

IS_ISB bool

Whether the instruction is ISB.

IS_LINK bool

Indicates if this is a branch with link.

PC unsigned int

The address of the branch instruction.

TARGET_ISET enum

The instruction set after the branch.

TARGET_PC unsigned int

The address the instruction branches to.

BRA_INDIR_CCFAIL

Indirect branches fail, perhaps to a register.

BRA_RET_CCFAIL

Branch return fail, perhaps to a register.

CACHE_MAINTENANCE_OP

Cache Maintenance Operation. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

DATA unsigned int

Specified MVA or Set/way.

FUNCTION enum

Clean / Invalidate.

SCOPE enum

Affected region.

SIDE enum

Inst / Data.

CCFAIL

Conditional instruction condition check fail. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CCFAIL_UNC

Unconditional instruction condition check fail. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CCPASS

Conditional instruction condition check pass. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CCPASS_UNC

Unconditional instruction condition check pass. Fields:

COND **enum**

The condition of the conditional instruction.

CORE_NUM **unsigned int**

Core number in a multi processor.

PC **unsigned int**

The address of the conditional instruction.

CODE_CACHE_MAINT

Code cache maintenance. Fields:

END_INCL **unsigned int**

Inclusive end address for VA or PA requests.

NS **bool**

Non-secure world for PA.

START **unsigned int**

Start address for VA or PA requests.

TYPE **enum**

Request type.

COMPILE_BLOCK_END

Last instruction of basic block translated. Fields:

VADDR **unsigned int**

Address of next instruction after this basic block.

COMPILE_BLOCK_START

First instruction of basic block translated. Fields:

VADDR **unsigned int**

Address of first instruction in basic block.

COMPILE_INST

ARM instruction compiled. Fields:

DISASS **string**

Disassembly of the instruction.

ISSET **enum**

The instruction set of this instruction.

ITSTATE **unsigned int**

The ITSTATE current for the instruction.

OPCODE **unsigned int**

The opcode of the instruction.

PC unsigned int

The address of the instruction.

SIZE unsigned int

The size of the instruction in bytes.

CONTEXTIDR

Context ID Register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

NS bool

Secure or nonsecure banked register is accessed.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The new value written.

CONTEXT_SYNC

Called for every context synchronization event. Fields:

ARCH_CSE bool

Architectural Context Synchronization Event.

PC unsigned int

The address of the synchronization instruction.

SYSREG_SYNC bool

System register synchronization event.

CORE_ENDIAN

Core BE8 Big-Endian state changed. Fields:

BE8 bool

Core BE8 Big-Endian state.

CORE_INFO

Static processor attributes. Only triggered by a call to DumpState(). Fields:

ARCH_PROFILE enum

The architecture profile of the core.

CLUSTER_ID unsigned int

The cluster ID of this processor.

CORE_NUM unsigned int

The number of this core in an MP processor.

DETAILED_CORE_FEATURES enum

Detailed core features as a bitfield. Enum values are bitmasks to extract the info from the field.

FPU_VERSION enum

The VFP version implemented by the core.

MEM_ARCH enum

The memory architecture of the core.

MIDR_EL1 unsigned int

MIDR_EL1 value.

NUM_CORES unsigned int

The number of cores in this MP processor.

QUANTUM_SIZE unsigned int

The default quantum size of the core (in ticks).

SECURITY_FEATURES bool

Does the core have security features?.

CORE_LOADS

Processor load accesses. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ACQREL enum

Is this an acquire/release.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DATA unsigned int

The data read or written.

ELEMENT_SIZE unsigned int

Width of each element.

ENDIANNESS enum

Byte ordering of this access.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

LOCK enum

Normal, exclusive or locked access.

MECID unsigned int

Memory Encryption Context ID, if applicable.

MECID2 unsigned int

Memory Encryption Context ID of the second page, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PAGE_SIZE unsigned int

Page size as $\log_2(\text{size in bytes})$.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK, 2=Exclusive Failed, 3=Unsupported.

SIZE unsigned int

Width of the access in bytes. Only required if DATA is not traced.

TRANS bool

Is this a translated access.

VADDR unsigned int

The virtual address of the access.

CORE_REGS

Changes of the core registers R0 to R14. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number, 0 to 14.

MODE enum

Bank of the register accessed.

OLD_VALUE unsigned int

The old value overwritten.

PHYS_ID enum

The physical register accessed.

VALUE unsigned int

The new value written to the register.

CORE_REGS64

Changes of the core registers X0..X30, SP_ELn. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ID enum

The register number, 0-30 for X0-X30, >=32 for SP_ELn.

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

CORE_REGS64_READ

General purpose AArch64 register has been read X0..X30, SP_ELn. Fields:

ID enum

The register number, 0-30 for X0-X30, >=32 for SP_ELn.

VALUE unsigned int

The value read from the register.

CORE_STALLED_TRANSITION

Detect when a core is prevented from progressing PChannel states. Fields:

CURRENT_STATE enum

Current power mode of core.

REASON string

The reason a transition was stalled, and what must be done to unblock it.

TARGET_STATE enum

Desired power mode of core after transition.

CORE_STORES

Processor store accesses. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ACQREL enum

Is this an acquire/release.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DATA unsigned int

The data read or written.

ELEMENT_SIZE unsigned int

Width of each element.

ENDIANNESS enum

Byte ordering of this access.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

LOCK enum

Normal, exclusive or locked access.

MECID unsigned int

Memory Encryption Context ID, if applicable.

MECID2 unsigned int

Memory Encryption Context ID of the second page, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PAGE_SIZE unsigned int

Page size as $\log_2(\text{size in bytes})$.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK, 2=Exclusive Failed, 3=Unsupported.

SIZE unsigned int

Width of the access in bytes. Only required if DATA is not traced.

TRANS bool

Is this a translated access.

VADDR unsigned int

The virtual address of the access.

CP14_READ

System Coprocessor register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP14 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc1 unsigned int

opcode 1.

opc2 unsigned int

opcode 2.

CP14_WRITE

System Coprocessor register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP14 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc1 unsigned int

opcode 1.

opc2 unsigned int

opcode 2.

CP15_READ

System Coprocessor register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP15 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc1 unsigned int

opcode 1.

opc2 unsigned int

opcode 2.

CP15_READ64

System Coprocessor register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP15 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc unsigned int

opcode 1.

CP15_WRITE

System Control Coprocessor register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP15 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc1 unsigned int

opcode 1.

opc2 unsigned int

opcode 2.

CP15_WRITE64

System Coprocessor register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP15 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc unsigned int

opcode 1.

CPSR

CPSR change. (In AArch64 PSTATE represented in SPSR format). Fields:

CORE_NUM unsigned int

Core number in a multi processor.

OLD_VALUE unsigned int

The old CPSR value.

UNKNOWN unsigned int

Bits within the register that have unknown value.

VALUE unsigned int

The new CPSR value.

CRYPTO_SPEC

Every crypto instruction speculatively executed.

CorePowerStateChange

Core power-up state has changed. DISPLAY Core powered %{PU:(up|down)}. Fields:

PU bool

Whether the core powered up.

DATA_CACHE_SET_ALLOC_TAGS

Set allocation tags from DC GVA instruction. Fields:

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

NSDESC unsigned int

The security state of the access.

PADDR unsigned int

The physical address of the access.

PAS enum

The physical address space of the page.

RESPONSE enum

0=Aborted, 1=OK.

SIZE unsigned int

The size of memory in bytes.

TAG unsigned int

The value of the tag being set.

TAGGING_TYPE enum

Type of Memory tagging (Physical or Virtual).

TAG_VADDR unsigned int

The virtual address of the Virtual Tag if Virtual tagging is used; otherwise 0.

VADDR unsigned int

Virtual address used for setting allocation tags.

DATA_CACHE_ZERO

Zero and invalidate cache line from DC ZVA instruction. Fields:

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

NSDESC unsigned int

The security state of the access.

PADDR unsigned int

The physical address of the access.

PADDR_TAG unsigned int

If tag checked, the physical address tag for the access.

PAS enum

The physical address space of the page.

RESPONSE enum

0=Aborted, 1=OK.

SIZE unsigned int

The size of zero'd memory in bytes.

TAG_CHECKED bool

Is this access tag checked.

VADDR unsigned int

Virtual address of the zero'd cache line.

DEBUG_EVENT

Hardware debug support event. Fields:

EVENT enum

Description of event.

VALUE unsigned int

data value.

DEBUG_TARGET_ISET

Hardware debug target ISA. Fields:

TARGET_ISET enum

The target instruction set of the debug state.

DMI_ALLOCATE

DMI allocated. Fields:

ACCESS_DIRECTION enum

READ / WRITE.

PADDR unsigned int

The physical address of the DMI.

PTR unsigned int

DMI range pointer.

SIDE enum

DSIDE / ISIDE.

SIZE unsigned int

The size of DMI.

VADDR unsigned int

The virtual address of the DMI..

DMI_HIT

DMI hit, access used DMI (only fire for small DMI regions i.e. less than a full page). Fields:

ACCESS_DIRECTION enum

READ / WRITE.

PADDR unsigned int

The physical address of the DMI.

SIDE enum

DSIDE / ISIDE.

SIZE unsigned int

The size of the DMI.

VADDR unsigned int

The virtual address of the DMI..

DMI_REVOKE

DMI revoked. Fields:

ACCESS_DIRECTION enum

READ / WRITE.

PADDR unsigned int

The physical address of the DMI.

SIDE enum

DSIDE / ISIDE.

SIZE unsigned int

The size of DMI.

VADDR unsigned int

The virtual address of the DMI..

END_COMPILE

Compilation end. Fields:

END_OF_PAGE_COUNT unsigned int

Number of basic blocks exited due to page boundary since START_COMPILE.

FETCHFAIL_COUNT unsigned int

Number of basic blocks exited due to ifetch failure START_COMPILE.

INST_COUNT unsigned int

Number of instructions compiled since START_COMPILE.

NONSEQ_COUNT unsigned int

Number of basic blocks exited due to non-sequential instructions since START_COMPILE.

PAGE_STRADDLE_COUNT unsigned int

Number of basic blocks exited due to unaligned instructions crossing page since START_COMPILE.

ESB

Error Synchronization Barrier hit. Fields:

ASYNC bool

ESB is async.

IMPLICIT bool

ESB is implicit.

EXCEPTION

Exception taken. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ESR unsigned int

The value assigned to ESR when exception is taken.

ESR_EC unsigned int

The Value assigned to ESR.EC when exception is taken.

IS_PHYSICAL bool

Physical or Virtual exception.

IS_TRAP bool

Exception coming from a trap.

LR unsigned int

The value assigned to the link register.

PC unsigned int

The location where the exception occurred.

PREFERRED_RETURN unsigned int

The preferred return address for the exception.

TAKEN_LOCALLY bool

Whether exception is taken locally(same exception level).

TARGET_ISET enum

The instruction set of the exception handler code.

TARGET_PC unsigned int

The address the exception branches to.

VECTOR enum

The exception vector.

VECTOR_OFFSET unsigned int

The offset of an exception vector from the base address.

EXCEPTION_END

Every exception completed.

EXCEPTION_INFO

Trace human-readable reason of the current exception for data/instruction aborts, floating point exceptions and unknown exceptions. Fields:

DESCRIPTION string

Reason for current exception.

ESR_EC unsigned int

The value of ESR.EC when exception is taken.

EXCEPTION_RAISE

Every exception raised.

EXCEPTION_RETURN

Branches on leaving exception. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISET enum

The instruction set of the branch instruction.

IS_COND bool

Indicates if this is a conditional branch.

IS_HINTED bool

Whether the conditional instruction is hinted.

IS_ISB bool

Whether the instruction is ISB.

IS_LINK bool

Indicates if this is a branch with link.

PC unsigned int

The address of the branch instruction.

TARGET_ISET enum

The instruction set after the branch.

TARGET_PC unsigned int

The address the instruction branches to.

EXCEPTION_RETURN_PREBRANCH

Exception return event that must occur before PC and processor state have updated.

EXCEPTION_START

Every exception started.

FIQ_TAKEN

FIQ taken exception.

FP_EXCEPTIONS_SIGNALLED

Trace which FP exceptions were signalled as a part of the FP operations performed by an instruction. Fields:

FP_EXCEPTIONS_SIGNALLED_MASK unsigned int

Indicates which FP exceptions were signalled by an instruction. This uses the same encoding as the cumulative floating-point exception bitfields in FPSR.

FP_STATE

Floating point state. Fields:

FPSR **unsigned int**

FPSR register value.

MASK **unsigned int**

The mask indicating updated cumulative bits.

FREQ_CHANGED

Core frequency changed. Fields:

CORE_NUM **unsigned int**

Core number in a multi processor.

INST_COUNT **unsigned int**

The instruction count of this core.

NEW_FREQ **signed int**

The new frequency of this core (expressed in Hz).

OLD_FREQ **signed int**

The old frequency of this core (expressed in Hz).

HLT

HLT instruction occurred. Fields:

IMM16 **unsigned int**

The 16-bit immediate value encoded in HLT instruction.

INST

Every instruction executed. Fields:

CORE_NUM **unsigned int**

Core number in a multi processor.

CURRENT_TIME **unsigned int**

The core's current time, as simulated time plus local time.

DEBUG_STATE **bool**

The instruction is executed in debug state.

DISASS **string**

Disassembly of instruction.

INST_COUNT **unsigned int**

The core's instruction counter, starting at 1 for the first instruction.

ISSET **enum**

The current instruction set.

ITSTATE **unsigned int**

The current ITSTATE.

LOCAL_TIME **unsigned int**

The core's local time, relative to the current quantum.

MODE enum

The mode the core is in.

NS unsigned int

The core's non-secure bit.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

OPCODE unsigned int

The opcode of the instruction.

PADDR unsigned int

The physical address of the instruction.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the instruction.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the page.

PC unsigned int

The address of the instruction.

SECURITY_STATE enum

The core's security state.

SIZE unsigned int

The size of the instruction in bytes.

INST_END

Every instruction completed.

INST_START

Every instruction started. Fields:

DEBUG_STATE bool

The instruction is executed in debug state.

ISSET enum

The current instruction set.

MODE enum

The mode the core is in.

NS enum

The current Secure State.

PC unsigned int

The address of the conditional instruction.

SECURITY_STATE enum

The current Security State.

INST_STRADDLE

Instruction straddles boundary. Fields:

DEBUG_STATE bool

The instruction is executed in debug state.

ISET enum

The current instruction set.

MODE enum

The mode the core is in.

NS enum

The current Secure State.

PC unsigned int

The address of the conditional instruction.

SECURITY_STATE enum

The current Security State.

IRQ_TAKEN

IRQ taken exception.

ISB

ISB instruction occurred. Fields:

PC unsigned int

The address of the ISB instruction.

LOAD_MULTIPLE_REGS_LIST

List of destination registers (not source) of a multiple registers load instruction. Fields:

REG string

Register to add into the list.

LOCAL_MONITOR

Local monitor activity. Fields:

PADDR unsigned int

Local Monitor Address.

State enum

State of the monitor (Open/Exclusive).

MAPPING_CREATE

A memory mapping has been created. This is an implementation detail of the model and so may change without notice. Fields:

ASID **unsigned int**

The ASID used during translation, or 0 if no ASID was used.

IS_VALID **bool**

Whether the mapping is valid or not.

PADDR **unsigned int**

Base physical address of region, or 0 if not known.

SIDE **enum**

I-side or D-side.

SIZE **unsigned int**

Size of the page, or 0 if not known.

VADDR **unsigned int**

Base virtual address of region.

VMID **unsigned int**

The VMID used during translation, or 0 if no VMID was used.

MAPPING_DESTROY

A memory mapping has been destroyed. This is an implementation detail of the model and so may change without notice. Fields:

ASID **unsigned int**

The ASID used during translation.

CAUSE **enum**

The cause of invalidation.

PADDR **unsigned int**

Base physical address of page.

SIDE **enum**

I-side or D-side.

SIZE **unsigned int**

Size of the page.

VADDR **unsigned int**

Base virtual address of page.

VMID **unsigned int**

The VMID used during translation.

MEMMAP_DEBUG_READ

Memory mapped reads to the debug registers. Fields:

ADDR unsigned int

address.

CORE_NUM unsigned int

Core number in a multi processor.

EXT bool

Whether access is from an external device (such as the DAP).

NS enum

Secure state of the access.

REG_NAME string

Name of the debug register accessed.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

MEMMAP_DEBUG_WRITE

Memory mapped writes to the debug registers. Fields:

ADDR unsigned int

address.

CORE_NUM unsigned int

Core number in a multi processor.

EXT bool

Whether access is from an external device (such as the DAP).

NS enum

Secure state of the access.

REG_NAME string

Name of the debug register accessed.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The new value written.

MMU_D128_TTB_READ

This event is triggered by reads caused by a D128 translation table walk. Fields:

ABORTED bool

Was the walk successful.

DATA_HIGH unsigned int

The data upper bits read.

DATA_LOW unsigned int

The data lower bits read.

IPA unsigned int

For stage 1, the IPA of the read.

LEVEL unsigned int

Translation table level.

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

OUTADDR unsigned int

The output physical address.

PADDR unsigned int

The physical address of the read.

PASmem enum

Physical Address Space of the accessed memory.

PASreq enum

Physical Address Space of the requested lookup.

REGIME_EL enum

Entry matches in this translation regime.

REQUESTER enum

The requester of this table walk.

SIDE enum

Inst / Data.

STAGE unsigned int

Translation stage.

MMU_D128_TTB_WRITE

This event is triggered by writes caused by a D128 translation table walk. Fields:

ABORTED bool

Was the walk successful.

DATA_HIGH unsigned int

The data upper bits written.

DATA_LOW unsigned int

The data lower bits written.

IPA unsigned int

For stage 1, the IPA of the write.

IS_HACDBS_ACCESS bool

If the access was due to HACDBS_CLEAN.

LEVEL unsigned int

Translation table level.

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

OUTADDR unsigned int

The output physical address.

PADDR unsigned int

The physical address of the write.

PASmem enum

Physical Address Space of the accessed memory.

PASreq enum

Physical Address Space of the requested lookup.

REGIME_EL enum

Entry matches in this translation regime.

SIDE enum

Inst / Data.

STAGE unsigned int

Translation stage.

MMU_TRANS

Address translation information. Fields:

ASID unsigned int

Address space identifier.

CORE_NUM unsigned int

Core number in a multi processor.

Hyp bool

Entry matches in Hyp state only.

INNERCACHE_RA bool

Is the inner cache allocate on read.

INNERCACHE_TRANSIENT bool

Is the inner write-through transient.

INNERCACHE_TYPE enum

Inner Caching scheme (NC/MB/WA).

INNERCACHE_WA bool

Is the inner cache allocate on write.

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MEMTYPE enum

Memory type.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC enum

Is secure side supposed to access secure or nonsecure memory.

OUTERCACHE_RA bool

Is the outer cache allocate on read.

OUTERCACHE_TRANSIENT bool

Is the outer write-through transient.

OUTERCACHE_TYPE enum

Outer Caching scheme (NC/MB/WA).

OUTERCACHE_WA bool

Is the outer cache allocate on write.

PADDR unsigned int

Physical address of the access.

PAGESIZE unsigned int

Page size as $\log_2(\text{size})$.

PAS enum

PAS supposed to be accessed is secure or nonsecure memory or Root/Realm if RME is implemented.

SH enum

Shareability.

SIDE enum

Inst / Data.

STAGE1_PERM unsigned int

Stage 1 permissions, [5:3] Privileged access (XWR) [2:0] User access (XWR).

STAGE2_PERM unsigned int

Stage 2 permission mask [5:3] Privileged access (XWR) [2:0] User access (XWR).

VADDR unsigned int

Virtual address of the access.

VMID unsigned int

Virtual machine identifier.

XS bool

XS Attribute.

nG enum

Flag indicating whether ASID will be matched.

MMU_TTB_READ

This event is triggered by reads caused by a translation table walk. Fields:

ABORTED bool

Was the walk successful.

DATA unsigned int

The data read.

DESC_KIND enum

The encoding scheme of the descriptors used for this walk.

IPA unsigned int

For stage 1, the IPA of the read.

LEVEL unsigned int

Translation table level.

LPAE boolIs this for an LPAE translation (*DEPRECATED*, use DESC_KIND).**MECID unsigned int**

Memory Encryption Context(MEC) ID, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSmem bool

Non secure memory access.

NSreq bool

Non secure request.

OUTADDR unsigned int

The output physical address.

PADDR unsigned int

The physical address of the read.

PASmem enum

Physical Address Space of the accessed memory.

PASreq enum

Physical Address Space of the requested lookup.

REGIME_EL enum

Entry matches in this translation regime.

REQUESTER enum

The requester of this table walk.

SIDE enum

Inst / Data.

STAGE unsigned int

Translation stage.

MMU_TTB_WRITE

This event is triggered by writes caused by a translation table walk. Fields:

ABORTED bool

Was the walk successful.

DATA unsigned int

The data written.

DESC_KIND enum

The encoding scheme of the descriptors used for this walk.

IPA unsigned int

For stage 1, the IPA of the write.

IS_HACDBS_ACCESS bool

If the access was due to HACDBS_CLEAN.

LEVEL unsigned int

Translation table level.

LPAE bool

Is this for an LPAE translation (*DEPRECATED*, use DESC_KIND).

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSmem bool

Non secure memory access.

NSreq bool

Non secure request.

OUTADDR unsigned int

The output physical address.

PADDR unsigned int

The physical address of the write.

PASmem enum

Physical Address Space of the accessed memory.

PASreq enum

Physical Address Space of the requested lookup.

REGIME_EL enum

Entry matches in this translation regime.

SIDE enum

Inst / Data.

STAGE unsigned int

Translation stage.

MODE_CHANGE

Mode change. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

MODE enum

The new mode.

NON_SECURE enum

The core's new non-secure bit.

OLD_MODE enum

The old mode.

SECURITY_STATE enum

The new security state.

NOT_TAG_CHECKED_INST

MTE Tag not checked for current instruction.

PERIODIC

Called for every quantum. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The instruction count of this CPU.

PC unsigned int

The address of the next instruction to be executed on this CPU.

PMU_COUNTER_OVERFLOW

PMU counter overflow. Fields:

COUNTER_GROUP enum

Counter group.

COUNTER_RANGE enum

Counter range.

EVENT_ID unsigned int

Event Identifier, field is valid only for Event counter group.

INDEX unsigned int

Counter index (as selected by PMSELR), field is valid only for Event/Cycle counter groups.

INTERRUPT bool

Is interrupt enabled on overflow for this counter.

POINTER_AUTH

Called when execution hits a pointer authentication check. Fields:

AUTH_STATE bool

Authentication result.

KEY_USED enum

Pointer authentication key.

POTINDEX_UPDATE

Value of POT index for current pc changed. Fields:

NEW_POTINDEX unsigned int

New POTIndex.

OLD_POTINDEX unsigned int

Old POTIndex.

PREFETCH_MEMORY64

Prefetch from PRFM or PRFUM instructions. Fields:

PRFOP enum

Prefetch hint.

VADDR unsigned int

Virtual address of the location that should be prefetched.

PRELOAD64_REQUEST

Preload requests from PRFM or PRFUM instructions. Fields:

PRFOP enum

Prefetch hint.

VADDR unsigned int

Virtual address of the location that should be prefetched.

PRELOAD_DATA

Data preload from PLD instruction. Fields:

VADDR unsigned int

Virtual address of the data that should be preloaded.

PRELOAD_INST

Instruction preload from PLI instruction. Fields:

VADDR unsigned int

Virtual address of the instruction that should be preloaded.

PRE_CORE_LOAD

Trace just before a core load. Fields:

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

PRE_CORE_STORE

Trace just before a core store. Fields:

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

PRE_TTB_READ

This event is triggered before reads caused by a translation table walk. Fields:

ADDR unsigned int

The physical address of the read.

PSTATE

PSTATE fields update. Fields:

ALLINT unsigned int

All interrupt mask bit.

BTYPE unsigned int

Branch target identification bit.

CORE_NUM unsigned int

Core number in a multi processor.

DAIF unsigned int

Exception mask bits.

DIT unsigned int

Data Independent Timing bit.

EL unsigned int

Current Exception level field.

EXLOCK unsigned int

Exception return state lock bit.

IL unsigned int

Illegal Execution state bit.

NZCV unsigned int

Condition flags.

PACM unsigned int

PACM bit.

PAN unsigned int

Privileged Access Never state bit.

PM unsigned int

PMU exception mask bit.

PPEND unsigned int

PMU exception pending bit.

SM unsigned int

StreamingMode status bit.

SP unsigned int

Stack pointer register selection bit.

SS unsigned int

Software Step bit.

SSBS unsigned int

Speculative Store Bypass Safe bit.

TCO unsigned int

Tag Check Override bit.

TOUCHED unsigned int

Mask of the fields that have been touched since this trace has last been emitted. Note that touched does not necessarily mean the value changed. The index of the field is the bit index in the mask plus the index of this field plus one.

UAO unsigned int

User Access Override bit.

UINJ unsigned int

Undefined Instruction exceptions bit.

ZA unsigned int

ZA storage and ZT0 register enable bit.

nRW unsigned int

Current Execution state bit.

PSTATE_PACTIVE_OPMODE_TRANSITION

Trace pactive and Opmode mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Operating mode after the transition.

OLD_STATE enum

Operating mode before the transition.

PSTATE_PACTIVE_OPMODE_TRANSITION_DENIED

Trace the denied pactive and Opmode mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Operating mode after the transition.

OLD_STATE enum

Operating mode before the transition.

REASON string

Reason for the transition being denied.

PSTATE_PACTIVE_POWER_TRANSITION

Trace pactive and power mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Power mode after the transition.

OLD_STATE enum

Power mode before the transition.

PSTATE_PACTIVE_POWER_TRANSITION_DENIED

Trace the denied pactive and power mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Power mode after the transition.

OLD_STATE enum

Power mode before the transition.

REASON string

Reason for the transition being denied.

PSTATE_PACTIVE_THREAD_OPMODE_TRANSITION

Trace pactive and Opmode mode transitions for threads on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Operating mode after the transition.

OLD_STATE enum

Operating mode before the transition.

QUANTUM_START

Called at the start of every quantum.

RANGE_PRELOAD_REQUEST

Preload requests from RPRFM. Fields:

COUNT unsigned int

Number of blocks of data to access.

LENGTH unsigned int

Number of contiguous bytes to be accessed in each block.

OP unsigned int

Range prefetch operation .

REUSE unsigned int

Maximum number of bytes to be accessed before next RPRFM .

STRIDE signed int

Stride to advance after accessing LENGTH bytes.

VADDR unsigned int

Base virtual address of range prefetch request.

RUN_STATE

Run state transition. Fields:

INST_COUNT unsigned int

Ticks count at point of transition.

NEW enum

New run state.

OLD enum

Old run state.

SBP_SAMPLE_COLLISION

DISPLAY Statistical profiling samples collided .

SBP_SAMPLE_DISCARDED

DISPLAY Operation discarded by statistical profiling unit when configured in discard mode.
Fields:

BUFFER_POINTER unsigned int

Value of buffer pointer.

DATA unsigned int

Data discarded.

SIZE unsigned int

Size of sample, in bytes.

SBP_SAMPLE_END

DISPLAY Statistical profiling of an operation ended.

SBP_SAMPLE_POP

DISPLAY Operation which might be sampled by statistical profiling.

SBP_SAMPLE_RECORDED

DISPLAY Operation sampled by statistical profiling has been written to the buffer at
%{BUFFER_ADDRESS} with size %{SIZE} and data %{DATA}. Fields:

BUFFER_ADDRESS unsigned int

Virtual Address of start of buffer write.

DATA unsigned int

Data written to the buffer.

SIZE unsigned int

Size of sample written to buffer, in bytes.

SBP_SAMPLE_START

DISPLAY Statistical profiling of an operation started.

SBP_SAMPLE_TAKEN

DISPLAY Operation sampled by statistical profiling after filtering.

SEMIHOSTING_CALL

Call of a semihost function occurred. Fields:

PC unsigned int

The program counter after the semihosting call.

SEMIHOSTING_PRECALL

About to call semihost. Fields:

EL2 bool

Use EL2 translation regime.

NS bool

Is Non-Secure.

REG_WIDTH unsigned int

The current register width in bytes.

SIGNAL

External signal state change. Fields:

SIGNAL enum

Signal that changed.

STATE bool

Signal asserted state.

SOFTWARE_STEP

Return the Debug Software Step state. Fields:

STATE enum

Software Step state.

SPSR

SPSR change. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

MODE enum

Which of the banked SPSR registers is written.

OLD_VALUE unsigned int

The old SPSR value.

VALUE unsigned int

The new SPSR value.

START_COMPILE

Compilation started. Fields:

VADDR unsigned int

Instruction where compilation begins.

STG_MULTIPLE_MICRO_OPS_ADDRESS_RANGE

Address range for MTE stores with multiple uops. Fields:

ADDR_END unsigned int

Final address.

ADDR_START unsigned int

Starting address.

STORE_MULTIPLE_REGS_ADDRESS_RANGE

Address range for stores with multiple registers. Fields:

ADDR_END unsigned int

Final address.

ADDR_START unsigned int

Starting address.

SUBOPTIMAL_LDST_RETIRED

suboptimal load/store. Fields:

ACCESS_SIZE unsigned int

Log2 of the access width in bytes used for alignment checking (1: half-word, 2: word ...).

OPTIMAL_ALIGN_SELECTOR unsigned int

Selector to decide the optimal alignment size (1: access width, 2: 4-bytes, 3: 8-bytes, ..., 12: 4-kbytes).

VADDR unsigned int

The virtual address of the access.

SYNC

Called for every synchronization, that is every quantum break. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The instruction count of this CPU.

LOCAL_QUANTUM unsigned int

The local quantum of this CPU.

LOCAL_TIME unsigned int

The local time of this CPU.

SYNC_REASON enum

Reason for sync.

SYSCALL

System call instruction executed. Fields:

IMM unsigned int

Immediate value of the system call instruction.

TYPE enum

System call type.

VADDR unsigned int

Instruction that caused the system call.

SYSREG_READ128

System Coprocessor 128 bit register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

REGNUM enum

Internal register number.

REGNUM_PHYS enum

Internal register number physically accessed (e.g. in presense of VHE redirection).

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_READ64

System Coprocessor register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REGNUM enum

Internal register number.

REGNUM_OPERAND enum

Register number corresponding to the instruction operand.

REGNUM_PHYS enum

Internal register number physically accessed (e.g. in presense of VHE redirection).

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_UPDATE128

Triggers when the system updates a register. Fields:

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG enum

Register number.

UNKNOWN unsigned int

Bits of the register which became unknown.

VALUE unsigned int

Value written to the register.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_UPDATE32

Triggers when the system updates a register. Fields:

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG enum

Register number.

UNKNOWN unsigned int

Bits of the register which became unknown.

VALUE unsigned int

Value written to the register.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_UPDATE64

Triggers when the system updates a register. Fields:

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG enum

Register number.

UNKNOWN unsigned int

Bits of the register which became unknown.

VALUE unsigned int

Value written to the register.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_WRITE128

System Coprocessor 128 bit register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

REGNUM enum

Internal register number.

REGNUM_PHYS enum

Internal register number physically accessed (e.g. in presense of VHE redirection).

UNDEF bool

The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int
opcode 2.

SYSREG_WRITE64

System Coprocessor register write. Fields:

CORE_NUM unsigned int
Core number in a multi processor.

CRm unsigned int
CRm.

CRn unsigned int
CRn.

NS bool
Secure or nonsecure banked register is accessed.

REGNUM enum
Internal register number.

REGNUM_OPERAND enum
Register number corresponding to the instruction operand.

REGNUM_PHYS enum
Internal register number physically accessed (e.g. in presense of VHE redirection).

UNDEF bool
The register accessed is undefined.

UPDATED_VALUE unsigned int
Updated value of the register now it has been written.

VALUE unsigned int
The new value written.

opc unsigned int
opcode 1.

opc0 unsigned int
opcode 0.

opc2 unsigned int
opcode 2.

UNALIGNED_LDST_RETIRED

Processor unaligned load/store. Fields:

ACCESS_SIZE unsigned int
Log2 of the access width in bytes used for alignment checking (1=>half-word, 2->word ...).

NUMBER_OF_BEATS unsigned int

Number of accesses (beats) in this burst (Total size in bytes is ACCESS_SIZE * NUMBER_OF_BEATS).

VADDR unsigned int

The virtual address of the access.

VFP_D_REGS

VFP/NEON D 64 bit register write. Fields:

ALIASING enum

Type of register aliasing used.

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a write to the corresponding bit in the register. For example, 0x1 would indicate a write to VALUE[0].

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

VFP_D_REGS_READ

VFP/NEON D 64 bit register read. Fields:

ID unsigned int

The register number.

VALUE unsigned int

The current value of the register.

VFP_Q_REGS

VFP/NEON Q 128 bit register write. Fields:

ALIASING enum

Type of register aliasing used.

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a write to the corresponding byte in the register. For example, 0x1 would indicate a write to VALUE[7:0].

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

VFP_Q_REGS_READ

VFP/NEON Q 128 bit register read. Fields:

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a read to the corresponding byte in the register. For example, 0x1 would indicate a read to VALUE[7:0].

VALUE unsigned int

The current value of the register.

VFP_SYS_REGS

Writes to the VFP/NEON units system registers. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ID enum

Which VFP system register is written.

MASK unsigned int

The mask indicating updated cumulative bits.

OLD_VALUE unsigned int

The register's old value overwritten.

VALUE unsigned int

The new value written to the VFP system register.

VFP_S_REGS

VFP/NEON S 32 bit register write. Fields:

ALIASING enum

Type of register aliasing used.

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a write to the corresponding bit in the register. For example, 0x1 would indicate a write to VALUE[0].

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

VFP_S_REGS_READ

VFP/NEON S 32 bit register read. Fields:

ID unsigned int

The register number.

VALUE unsigned int

The current value of the register.

VIRTUAL_TAG_ALLOCATED

Virtual tag information when allocated. Fields:

TAG_VADDR unsigned int

Virtual tag virtual address.

VADDR unsigned int

Data Virtual address.

VIRTUAL_TAG unsigned int

Virtual tag value.

WAYPOINT

Signals end of basic block execution. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ISET enum

Origin instruction set.

IS_COND bool

Indicates if this is a conditional waypoint.

PC unsigned int

Origin address (or 0 if unavailable).

TAKEN bool

Indicates if this waypoint was taken.

TARGET unsigned int

Destination address.

TARGET_ISET enum

Destination instruction set.

WFE_END

WFE ended. Fields:

INST_COUNT unsigned int

Ticks count when leaving WFE.

WFE_EVENT_REGISTER

WFE event register status: set/clear, reason. Fields:

INST_COUNT unsigned int

Ticks count.

REASON enum

Reason for set/clear. Only REASON==0 (Cleared by WFE) clears the bit, all other reasons set it.

WFE_IGNORED

WFE ignored. Fields:

EVENT bool

This WFE was ignored because the event register was set.

INST_COUNT unsigned int

Ticks count when ignoring WFE.

TRAPPED bool

This WFE was trapped.

WFE_START

WFE entered. Fields:

INST_COUNT unsigned int

Ticks count when entering WFE.

WFI_END

WFI ended. Fields:

INST_COUNT unsigned int

Ticks count when leaving WFI.

WFI_IGNORED

WFI ignored. Fields:

DISABLED bool

This WFI was ignored because WFI is disabled.

INST_COUNT unsigned int

Ticks count when ignoring WFI.

REASON enum

specifies reason why WFI trace was ignored.

TRAPPED bool

This WFI was trapped.

WFI_START

WFI entered. Fields:

INST_COUNT unsigned int

Ticks count when entering WFI.

WFI_WAKEUP

WFI wakeup. Fields:

INST_COUNT unsigned int

Ticks count when WFI wakeup occurred.

REASON enum

Reason for wakeup.

2.38 ARM_Cortex-M0

This section describes the trace sources.

ASYNC_MEMORY_FAULT

Context ID Register write. Fields:

FAULT unsigned int

Fault status present in ESR.ISS encoding for Error in ARCH64 and present in DFSR encoding for external abort format in ARCH32.

PADDR unsigned int

Physical Address (or 0 if unavailable).

ATOMIC_END_ACCESS

Bus trace access for atomics. Fields:

ACCESS_FAIL bool

Memory access failed.

ADDR unsigned int

The virtual address of the access.

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

COMPARE_VALUE unsigned int

Compare value for CAS.

LOAD_VALUE unsigned int

Loaded values.

NSDESC unsigned int

The security state of the access.

OPERAND_VALUE unsigned int

Operation's operand.

OPERATION enum

Operation type.

PADDR unsigned int

The physical address of the access.

PRIV bool

Is this a privileged access?.

ATOMIC_START_ACCESS

Bus trace access for atomics. Fields:

ADDR unsigned int

The virtual address of the access.

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

COMPARE_VALUE unsigned int

Compare value for CAS.

NSDESC unsigned int

The security state of the access.

OPERAND_VALUE unsigned int

Operation's operand.

OPERATION enum

Operation type.

PADDR unsigned int

The physical address of the access.

PRIV bool

Is this a privileged access?.

ArchMsg.Error.exit_code

DISPLAY exit code: %{EXITCODE} component kind: %{KIND} reason: %{REASON}. Fields:

EXITCODE signed int

Exit code to be returned.

KIND string

Component kind that invoked the exit code trace.

REASON string

Reason why the exit code trace was invoked (optional).

ArchMsg.Warning.Unpredictably Indexed PM Event Register

TAGS **UNPREDICTABLE** DISPLAY Event %{IsDirect:(Selector value|Register)} %{SEL} is >= %{N}.

Fields:

IsDirect unsigned int

Direct Access rather than indirect.

N unsigned int

Number of Accessible Registers.

SEL unsigned int

Selector.

ArchMsg.Warning.dap_csw_bad_size

A write to CM3DAP CSW has an invalid size field. DISPLAY Write %{DATA} to CM3DAP CSW has invalid size field. Fields:

DATA unsigned int

bits[2:0] are size.

ArchMsg.Warning.dcimvac_matches_watchpoint

TAGS **UNPREDICTABLE** DISPLAY Watchpoints matching an AArch32 DCIMVA are implementation defined.

ArchMsg.Warning.decode_fieldmismatch

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with different fields %{FIELD1} and %{FIELD2}. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_fracbitsnegative

TAGS **UNPREDICTABLE** DISPLAY VCVT Instruction is unpredictable with a negative number of fraction bits.

ArchMsg.Warning.decode_initblock

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable in an IT block.

ArchMsg.Warning.decode_initblocknotlast

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable in an IT block when not the last.

ArchMsg.Warning.decode_invalid_condition

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable when the condition is not AL.

ArchMsg.Warning.decode_invalidfieldcombination

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with the values in %{FIELD1} and %{FIELD2}. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_invalidvalue

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with the value in %{FIELD}. Fields:

FIELD enum

Field name.

ArchMsg.Warning.decode_mem_hint_unallocated

TAGS **UNPREDICTABLE** DISPLAY There is no memory hint allocated to this bit pattern.

ArchMsg.Warning.decode_registermatch

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with %{FIELD1} and %{FIELD2} as the same register. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_registermismatch

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with %{FIELD1} and %{FIELD2} as different registers. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_registeroutofrange

TAGS **UNPREDICTABLE** DISPLAY Floating-point instruction is unpredictable indexing beyond the end of the register bank.

ArchMsg.Warning.decode_sbzsbo

TAGS **UNPREDICTABLE** DISPLAY Reserved bits in the instruction are not canonical.

ArchMsg.Warning.decode_transactiontoobig

TAGS **UNPREDICTABLE** DISPLAY Attempt to perform a memory transaction of more than 128 bytes.

ArchMsg.Warning.decode_unpred_other

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable.

ArchMsg.Warning.decode_unpreduseofpc

TAGS UNPREDICTABLE DISPLAY Instruction is unpredictable when %{FIELD} is PC. Fields:

FIELD enum

Field name.

ArchMsg.Warning.decode_unpreduseofr13

TAGS UNPREDICTABLE DISPLAY Instruction is unpredictable when %{FIELD} is R13. Fields:

FIELD enum

Field name.

ArchMsg.Warning.decode_unpreduseofr13inreglist

TAGS UNPREDICTABLE DISPLAY Instruction is unpredictable with R13 in register-list.

ArchMsg.Warning.decode_useofd16orhigher

TAGS UNPREDICTABLE DISPLAY Attempt to use register D16 or higher where it is not allowed.

ArchMsg.Warning.decode_writebackandbaseinlist

TAGS UNPREDICTABLE DISPLAY Instruction is unpredictable with write back when the base register is in the transfer list.

ArchMsg.Warning.decode_zeroregistersinlist

TAGS UNPREDICTABLE DISPLAY Instruction is unpredictable when no registers are to be transferred.

ArchMsg.Warning.recursive_branch

An instruction is performing a branch that targets the same instruction. If this was intended then a WFI might be more effective DISPLAY recursive branch detected at PC=%{PC}. PRIMARY KEY PC. Fields:

PC unsigned int

Address of instruction causing the branch.

ArchMsg.Warning.recursive_exception

An instruction has generated an exception that targets the same instruction. DISPLAY recursive %{TYPE} exception detected at PC=%{PC} (CPSR=%{CPSR}). PRIMARY KEY PC. Fields:

CPSR unsigned int

Processor state.

PC unsigned int

Address of instruction causing the exception.

TYPE enum

Type of exception.

ArchMsg.Warning.reentrant_vector_catch

TAGS UNPREDICTABLE DISPLAY Vector catch debug event on Prefetch or Data abort vector %{ADDR}. Fields:

ADDR unsigned int
vector address.

ArchMsg.Warning.reserved_it_state

TAGS **UNPREDICTABLE** DISPLAY Execution with reserved IT state %{ITSTATE}. Fields:

ITSTATE unsigned int
reserved IT state.

ArchMsg.Warning.secure_vector_fetch_from_nonsecure

DISPLAY Secure vector table is being fetched from Non-secure memory.

ArchMsg.Warning.sg_in_it_block

Arm recommends that software does not place SG (security gate instruction) inside an IT instruction block.

ArchMsg.Warning.unpred_FAR_on_LRCPC3_non_atomic_mem_access

TAGS **UNPREDICTABLE** Unpredictable FAR/FSTATUS on LRCPC3 pair instructions resulting in two separate memory accesses
DISPLAY Unpredictable FAR on LRCPC3 non atomic memory access at %{ADDR1} with FSTATUS %{FSTATUS1} or %{ADDR2} with FSTATUS %{FSTATUS2}. Fields:

ADDR1 unsigned int
Possible FAR address 1.

ADDR2 unsigned int
Possible FAR address 2.

FSTATUS1 unsigned int
Possible fault status value 1.

FSTATUS2 unsigned int
Possible fault status value 2.

ArchMsg.Warning.unpred_async_tag_check_fail_on_sve_speculative_load_do_update_ffr

Unpredictable set to FALSE of the FFR element associated to SVE speculative load with an async tag check fail TAGS **UNPREDICTABLE**.

ArchMsg.Warning.unpredictable_a32_breakpoint

TAGS **UNPREDICTABLE** DISPLAY Unpredictable breakpoint on a A32 instruction at %{ADDR1} Offset %{OFFSET} BAS %{BAS} for a %{IS_ADDRESS_MISMATCH_BREAKPOINT:(Mismatch|Match)} Breakpoint number %{N:d} Breakpoint address %{ADDR2}. Fields:

ADDR1 unsigned int
Address.

ADDR2 unsigned int
Breakpoint address.

BAS unsigned int
BAS field.

IS_ADDRESS_MISMATCH_BREAKPOINT bool

Is a Mismatch breakpoint, not a Match breakpoint.

N unsigned int

Breakpoint number.

OFFSET unsigned int

Always has a value of zero.

ArchMsg.Warning.unpredictable_banked_register_access

TAGS **UNPREDICTABLE** DISPLAY Access to banked register (R=%{R:d} SYSm=%{SYSm:2x}) is unpredictable. Fields:

R unsigned int

R bit.

SYSm unsigned int

SYSm.

ArchMsg.Warning.unpredictable_ccfail_undef

TAGS **UNPREDICTABLE** DISPLAY An instruction %{OPCODE} at VA=%{VADDR} failed its condition codes check but would otherwise have caused a non-data-dependent trap or undefined exception. Behaviour is implementation-defined choice of exception or nop. Fields:

OPCODE unsigned int

opcode of the instruction.

VADDR unsigned int

Virtual address of the instruction being executed.

ArchMsg.Warning.unpredictable_ici_bits

TAGS **UNPREDICTABLE** DISPLAY ICI bits do not apply or are architecturally unpredictable for instruction.

ArchMsg.Warning.unpredictable_t32_breakpoint

TAGS **UNPREDICTABLE** DISPLAY Unpredictable breakpoint on a T32 instruction at %{ADDR1} Offset %{OFFSET} BAS %{BAS} for a %{IS_ADDRESS_MISMATCH_BREAKPOINT:(Mismatch|Match)} Breakpoint number %{N:d} Breakpoint address %{ADDR2}. Fields:

ADDR1 unsigned int

Address.

ADDR2 unsigned int

Breakpoint address.

BAS unsigned int

BAS field.

IS_ADDRESS_MISMATCH_BREAKPOINT bool

Is a Mismatch breakpoint, not a Match breakpoint.

N unsigned int

Breakpoint number.

OFFSET unsigned int

Indicates which halfword of a 32-bit Thumb instruction the breakpoint is set on. 0 for the first halfword or if the instruction is 16 bits wide, or 1 for the second halfword.

ArchMsg.Warning.unpredictable_t32_vector_catch

TAGS **UNPREDICTABLE** DISPLAY Unpredictable Vector catch exception on a T32 instruction at **%{ADDR}** Offset **%{OFFSET}**. Fields:

ADDR unsigned int

Address.

OFFSET unsigned int

Indicates which halfword of a 32-bit Thumb instruction the vector catch exception is set on. 0 for the first halfword or if the instruction is 16 bits wide, or 1 for the second halfword.

ArchMsg.Warning.unpredictable_watchpoint_far

TAGS **UNPREDICTABLE** DISPLAY The value for FAR/EDWAR for hitting this watchpoint may be anywhere in the range **%{LOWER_BOUND}** to **%{UPPER_BOUND}**. Fields:

LOWER_BOUND unsigned int

the lowest address accessed by the instruction that triggered the watchpoint.

UPPER_BOUND unsigned int

the highest watchpointed address accessed by that instruction.

ArchMsg.Warning.warning_AdvSIMDExpandImmUnexpectedZero

TAGS **UNPREDICTABLE** DISPLAY AdvSIMDExpandImm may treat this immediate value as **UNPREDICTABLE**.

ArchMsg.Warning.warning_ConditionalSMC

TAGS **UNPREDICTABLE** DISPLAY SMC instruction has **UNPREDICTABLE** effects when conditional and when combined with traps.

ArchMsg.Warning.warning_access_crosses_page_boundary

TAGS **UNPREDICTABLE** DISPLAY Access crosses page boundary.

ArchMsg.Warning.warning_access_crosses_page_boundary_has_fault

TAGS **UNPREDICTABLE** DISPLAY Access that crosses page boundary has fault(s).

ArchMsg.Warning.warning_access_crosses_page_boundary_has_fault_on_both_pages

TAGS **UNPREDICTABLE** DISPLAY Access that crosses page boundary has a fault on each page. Fields:

ADDR1 unsigned int

FAR address on first page.

ADDR2 unsigned int

FAR address on second page.

FSTATUS1 unsigned int

Fault status on first page.

FSTATUS2 unsigned int

Fault status on second page.

ArchMsg.Warning.warning_access_crosses_page_boundary_spanning_different_memory

TAGS **UNPREDICTABLE** DISPLAY Access at address %{ADDR} crosses page from MEM_TYPE %{MEMTYPE_PAGE1} to %{MEMTYPE_PAGE2}. Fields:

ADDR unsigned int

address of access crossing page.

MEMTYPE_PAGE1 signed int

First Page's memory type (0-Normal, 1-Non-Normal).

MEMTYPE_PAGE2 signed int

Second Page's memory type (0-Normal, 1-Non-Normal).

ArchMsg.Warning.warning_access_wraps_around_memory

TAGS **UNPREDICTABLE** DISPLAY Access wraps around memory in %{WIDTH:d}bit mode. Fields:

WIDTH unsigned int

Address width.

ArchMsg.Warning.warning_bcr_linking_status

TAGS **UNPREDICTABLE** DISPLAY DBGBCR.{BT2,BT} combination gives unpredictable behaviour.

ArchMsg.Warning.warning_bcr_mask_reserved

TAGS **UNPREDICTABLE** DISPLAY DBGBCR_ELn.MASK is set to a reserved value.

ArchMsg.Warning.warning_bvr_masked_bit_not_zero

TAGS **UNPREDICTABLE** DISPLAY DBGBVRn_EL1.VA contains a bit masked by DBGBCRn_EL1.MASK that is not zero.

ArchMsg.Warning.warning_ccsidr_unimplemented_level

TAGS **UNPREDICTABLE** DISPLAY CCSIDR read while CSSELR %{CSSELR:x} points to an unimplemented cache level. Fields:

CSSELR unsigned int

current effective value of CSSELR.

ArchMsg.Warning.warning_change_to_ns_when_tge_set

TAGS **UNPREDICTABLE** DISPLAY Attempting to change NS to 1 when HCR.TGE = 1.

ArchMsg.Warning.warning_contiguous_bit_check_abort

TAGS **UNPREDICTABLE** DISPLAY An abort occurred whilst attempting to check TLB entry is contiguous at %{ENTRY_ADDR}. Fields:

ENTRY_ADDR unsigned int

address of conflicting TLB entry.

ArchMsg.Warning.warning_contiguous_bit_error

TAGS **UNPREDICTABLE** DISPLAY TLB entry at %{ENTRY_ADDR} was not contiguous with entry at %{CONTIG_ADDR} Contents are %{ENTRY_DATA} but expected %{CONTIG_DATA}. Fields:

CONTIG_ADDR unsigned int

address of the TLB first read, with which this entry is expected to be contiguous.

CONTIG_DATA unsigned int

expected contents based on the entry at CONTIG_ADDR.

ENTRY_ADDR unsigned int

address of conflicting TLB entry.

ENTRY_DATA unsigned int

contents of conflicting TLB entry.

ArchMsg.Warning.warning_contiguous_bit_error_D128

TAGS **UNPREDICTABLE** DISPLAY TLB entry at %{ENTRY_ADDR} was not contiguous with entry at %{CONTIG_ADDR} Contents are %{ENTRY_DATA_HI} %{ENTRY_DATA_LO} but expected %{CONTIG_DATA}. Fields:

CONTIG_ADDR unsigned int

address of the TLB first read, with which this entry is expected to be contiguous.

CONTIG_DATA unsigned int

expected contents based on the entry at CONTIG_ADDR.

ENTRY_ADDR unsigned int

address of conflicting TLB entry.

ENTRY_DATA_HI unsigned int

Higher bit contents of conflicting TLB entry.

ENTRY_DATA_LO unsigned int

Lower bit contents of conflicting TLB entry.

ArchMsg.Warning.warning_cp10_cp11_mismatch

TAGS **UNPREDICTABLE** DISPLAY Unpredictable configuration of CP10 and CP11 controls at EL %{EL:d}. Fields:

EL unsigned int

EL owning the control.

ArchMsg.Warning.warning_cp10_cp11_reserved_value

TAGS **UNPREDICTABLE** DISPLAY Unpredictable configuration of CPACR CP10 (%{CP10}) or CP11 (%{CP11}). Fields:

CP10 unsigned int

CP10 access.

CP11 unsigned int

CP11 access.

ArchMsg.Warning.warning_csselr_level_out_of_range

TAGS **UNPREDICTABLE** DISPLAY CSSELR.Level out of range, written value %{LEVEL} with only %{IMPLEMENTED:d} levels implemented. Fields:

IMPLEMENTED unsigned int

number of cache levels implemented.

LEVEL unsigned int

written value of CSSELR.Level.

TYPE enum

The type of cache selected by CSSELR.InD and TnD.

ArchMsg.Warning.warning_debug_flow_control_bits_not_obeyed

TAGS **UNPREDICTABLE** DISPLAY Flow control bits for %{REGISTER} were not obeyed, causing %{ERROR}. Fields:

ERROR enum

type of error resulting.

REGISTER enum

register under consideration.

ArchMsg.Warning.warning_debug_register_access_during_reset

TAGS **UNPREDICTABLE** DISPLAY Unpredictable %{IS_WRITE:(read|write)} access to debug register offset %{OFFSET} during reset. Fields:

IS_WRITE unsigned int

Write Not Read.

OFFSET unsigned int

Register Offset.

ArchMsg.Warning.warning_decode_cps_inconsistent_fields

TAGS **UNPREDICTABLE** DISPLAY The mode setting fields in the CPS instruction are inconsistent.

ArchMsg.Warning.warning_decode_invalid_state

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable in current exception-level/security state.

ArchMsg.Warning.warning_default_cacheable_mmu_on

TAGS **UNPREDICTABLE** DISPLAY Default cacheable is enabled (HCR.DC = 1) while MMU is enabled.

ArchMsg.Warning.warning_default_cacheable_vmmu_off

TAGS **UNPREDICTABLE** DISPLAY Default cacheable is enabled (HCR.DC = 1) while HCR.VM = 0.

ArchMsg.Warning.warning_deprecated_wvr_bit_2_set

TAGS **UNPREDICTABLE** DISPLAY DBGWVR%{N:d}_EL1=%{WVR:h} has bit 2 set, deprecated in ARMv8. Fields:

N unsigned int
index.

WVR unsigned int
DBGWVR.

ArchMsg.Warning.warning_eret_while_software_step_active_pending

Missing ISB between setting MDSCR_EL1.SS and ERET TAGS UNPREDICTABLE.

ArchMsg.Warning.warning_exclusive_to_non_normal

TAGS UNPREDICTABLE DISPLAY Exclusive Access to Non-Normal Memory %{ADDR}. Fields:

ADDR unsigned int
Address.

ArchMsg.Warning.warning_exclusive_to_non_writeback

TAGS UNPREDICTABLE DISPLAY Exclusive Access to Non-Writeback-Cacheable Memory %{ADDR}. Fields:

ADDR unsigned int
Address.

ArchMsg.Warning.warning_execute_from_device_memory

TAGS UNPREDICTABLE DISPLAY An attempt was made to execute from Device memory.

ArchMsg.Warning.warning_illegal_cpsr_mode

TAGS UNPREDICTABLE DISPLAY Writing an illegal or unimplement mode (%{NEW_MODE:h}) to CPSR was unpredictable. Fields:

NEW_MODE unsigned int
New value of CPSR.M.

ArchMsg.Warning.warning_illegal_srs_mode

TAGS UNPREDICTABLE DISPLAY Illegal or UNPREDICTABLE mode (%{MODE:h}) used for SRS instruction. Fields:

MODE unsigned int
mode for Banked SP.

ArchMsg.Warning.warning_implementation_defined_read_debug_register_in_SCS

Read Access to Debug registers from the processor is IMPLEMENTATION DEFINED TAGS IMP_DEF. Fields:

PPB_OFFSET unsigned int
debug register offset.

ArchMsg.Warning.warning_implementation_defined_sequential_security_transitions_supported

The behavior of sequential instruction fetches that cross from non-secure to secure memory and contain an SG instruction is CONSTRAINED UNPREDICTABLE TAGS UNPREDICTABLE.

ArchMsg.Warning.warning_implementation_defined_stack_limit_check_supported

It is **IMPLEMENTATION DEFINED** whether stack pointer limit checking is performed for this instruction TAGS IMP_DEF. Fields:

ADDRESS unsigned int

Address of instruction.

OPCODE unsigned int

instruction opcode.

ArchMsg.Warning.warning_implementation_defined_taggednonsharedmemorytype

TAGS IMP_DEF DISPLAY Memory attribute has Tagged, non shared memory type; it is **IMPLEMENTATION DEFINED** whether the memory location is treated as Tagged or non-Tagged.

ArchMsg.Warning.warning_implementation_defined_write_debug_register_in_SCS

Write Access to Debug registers from the processor is **IMPLEMENTATION DEFINED** TAGS IMP_DEF. Fields:

DATA unsigned int

data attempted to be written.

PPB_OFFSET unsigned int

debug register offset.

ArchMsg.Warning.warning_invalid_tcr_granule

TAGS **UNPREDICTABLE** DISPLAY TCR.TG{%TG_ID:(0|1)} bits have been set to a granule size {%REQUEST:d}K not implemented - using {%SUBSTITUTE:d}K. Fields:

REQUEST signed int

page size requested (or 0 for reserved).

SUBSTITUTE signed int

best guess available page size.

TG_ID bool

bits TG0 or TG1.

ArchMsg.Warning.warning_load_multiple_user_registers_from_user_mode

TAGS **UNPREDICTABLE** DISPLAY An LDM(user registers) instruction executed from user mode.

ArchMsg.Warning.warning_mte_in_mem_attr_not_supported

TAGS **UNPREDICTABLE** DISPLAY Memory attribute {%ATTR} has Tagged memory type but Armv8.5-MTE is not implemented. Fields:

ATTR unsigned int

Memory attribute.

ArchMsg.Warning.warning_reserved_breakpoint_state_match

TAGS **UNPREDICTABLE** DISPLAY {%IS_BREAKPOINT:(Watchpoint|Breakpoint)} programmed with a reserved combination of HMC, SSC and {%IS_BREAKPOINT:(PAC|PMC)}. Fields:

IS_BREAKPOINT unsigned int

Is a breakpoint, not a watchpoint.

ArchMsg.Warning.warning_shareability

TAGS **UNPREDICTABLE** DISPLAY Unpredictable: combination of the 1st and 2nd stages of translation is Normal Inner Non-Cacheable, Outer Non-Cacheable.

ArchMsg.Warning.warning_software_step_set_while_enabled

TAGS **UNPREDICTABLE** DISPLAY MDSCR_EL1.SS set to 1 while software step debug exceptions are enabled.

ArchMsg.Warning.warning_thumb_instruction_wraps_around_memory

TAGS **UNPREDICTABLE** DISPLAY Thumb instruction wraps around memory.

ArchMsg.Warning.warning_ttbr_sbz_bits_are_not_zero

TAGS **UNPREDICTABLE** DISPLAY %{TTBR}[%{MSB:d}:0] should be zero, but are not. Fields:

MSB unsigned int

x-1.

TTBR enum

which TTBR register.

ArchMsg.Warning.warning_unaligned_address_dbgdtrrx_write

TAGS **UNPREDICTABLE** DISPLAY Unaligned address %{ADDR} in X0/R0 when DBGDTRRX written in memory access mode. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unaligned_address_dbgdtrtx_read

TAGS **UNPREDICTABLE** DISPLAY Unaligned address %{ADDR} in X0/R0 when DBGDTRTX read in memory access mode. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unaligned_to_device

TAGS **UNPREDICTABLE** DISPLAY Unaligned Access to Device Memory %{ADDR}. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unaligned_to_strongly_ordered

TAGS **UNPREDICTABLE** DISPLAY Unaligned Access to Strongly Ordered Memory %{ADDR}. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unknown_sau_rnr

TAGS UNPREDICTABLE DISPLAY SAU_RNR was set to an unsupported value.

ArchMsg.Warning.warning_unpredictable_AIRCR_PRIS_and_BFHFNMINs

TAGS UNPREDICTABLE DISPLAY The effect of setting both AIRCR.BFHFNMINs and AIRCR.PRIS to 1 is UNPREDICTABLE.

ArchMsg.Warning.warning_unpredictable_AIRCR_VECTCLRACTIVE_when_not_in_debug

TAGS UNPREDICTABLE DISPLAY The effect of writing a 1 to AIRCR.VECTCLRACTIVE if the processor is not halted in Debug state is UNPREDICTABLE.

ArchMsg.Warning.warning_unpredictable_AIRCR_VECTRESET_and_SYSRESETREQ

TAGS UNPREDICTABLE DISPLAY When the processor is halted in Debug state, if a write to the register writes a 1 to both VECTRESET and SYSRESETREQ, the behavior is UNPREDICTABLE.

ArchMsg.Warning.warning_unpredictable_AIRCR_VECTRESET_when_not_in_debug

TAGS UNPREDICTABLE DISPLAY The effect of writing a 1 to AIRCR.VECTRESET if the processor is not halted in Debug state is UNPREDICTABLE.

ArchMsg.Warning.warning_unpredictable_AIRCR_incorrect_VKEY

TAGS UNPREDICTABLE DISPLAY The value 0x05FA must be written to AIRCR.VKEY, otherwise the register write is UNPREDICTABLE.

ArchMsg.Warning.warning_unpredictable_EXC_RETURN_Reserved_Bit

EXC_RETURN[23:7] are reserved with the special condition that all bits should be written as one. Values other than all 1s are UNPREDICTABLE. TAGS UNPREDICTABLE. Fields:

EXC_RETURN unsigned int

Special PC value.

SBO_MASK unsigned int

Which bits Should Be One to not UNPRED.

ArchMsg.Warning.warning_unpredictable_PMEVTYPEPn_ELO_VS_reserved_value

TAGS UNPREDICTABLE DISPLAY Unpredictable reserved value of PMEVTYPEn<n>_ELO.VS.

ArchMsg.Warning.warning_unpredictable_change_to_DEMCR_MON_STEP_at_insufficient_priority

TAGS UNPREDICTABLE DISPLAY The effect of changing DEMCR.MON_STEP at an execution priority that is lower than the priority of the DebugMonitor exception is **CONSTRAINED** UNPREDICTABLE.

ArchMsg.Warning.warning_unpredictable_change_to_priority_of_active_exception

TAGS UNPREDICTABLE DISPLAY Changing the priority of an active exception is UNPREDICTABLE.

ArchMsg.Warning.warning_unpredictable_clear_lspact

TAGS UNPREDICTABLE DISPLAY UNPREDICTABLE state on context switch. Fields:

DRVEXC_TAKEN unsigned int

Derived exception taken, not pended.

REACHED_LAST_STORE unsigned int

Only the last faulted.

ArchMsg.Warning.warning_unpredictable_dbg_exit_unaligned_dlr

TAGS **UNPREDICTABLE** DISPLAY Debug exit to AArch32 with DLR[0] set to 1. PC[0] can be set to 0 or to DLR[0].

ArchMsg.Warning.warning_unpredictable_exception_catch

TAGS **UNPREDICTABLE** DISPLAY Generation of an exception catch event is unpredictable.

ArchMsg.Warning.warning_unpredictable_exception_return_inconsistent_state

TAGS **UNPREDICTABLE** DISPLAY state on exception return is unpredictable.

ArchMsg.Warning.warning_unpredictable_exception_return_instruction

TAGS **UNPREDICTABLE** DISPLAY exception return instruction is unpredictable.

ArchMsg.Warning.warning_unpredictable_in_debug_state

TAGS **UNPREDICTABLE** DISPLAY Instruction is **UNPREDICTABLE** when executed in debug state.

ArchMsg.Warning.warning_unpredictable_mair_encoding

TAGS **UNPREDICTABLE** DISPLAY Memory attribute %{ATTR} has invalid MAIR encoding; this may lead to unpredictable results. Fields:

ATTR unsigned int

Memory attribute.

ArchMsg.Warning.warning_unpredictable_pmu_counter_access

TAGS **UNPREDICTABLE** DISPLAY Access to PMU counters from non-secure EL0 or EL1 is **UNPREDICTABLE** with MDCR_EL2.HPMN set to 0.

ArchMsg.Warning.warning_unpredictable_prioritization_breakpoint_match_vector_catch

TAGS **UNPREDICTABLE** DISPLAY Two events with the same priority occurred at the same instruction: Address Matching Vector Catch debug event, and Breakpoint debug event. It is constrained unpredictable which is taken.

ArchMsg.Warning.warning_unpredictable_stack_selection

TAGS **UNPREDICTABLE** DISPLAY **UNPREDICTABLE** state on context switch.

ArchMsg.Warning.warning_unpredictable_tsize_out_of_range

TAGS **UNPREDICTABLE** DISPLAY %{TCR}.T%{TTBR:d}SZ (%{TSIZE:d} bits) is out of range. Must be between 25 and %{TSIZE_MAX:d}. Fields:

TCR enum

Which TCR.

TSIZE signed int

Value of TSize.

TSIZE_MAX signed int

The maximum allowed value.

TTBR unsigned int

Which TTBR region.

ArchMsg.Warning.warning_unpredictable_unaligned_pc_as_base_register

TAGS **UNPREDICTABLE** DISPLAY The PC value must be word-aligned, otherwise the behavior of the instruction is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_unaligned_pop_stack

TAGS **UNPREDICTABLE** DISPLAY **CONSTRAINED UNPREDICTABLE** unaligned stack frame %{FRAME_ADDR:x} in exception-return. Fields:

FRAME_ADDR unsigned int

unaligned stack frame address.

ArchMsg.Warning.warning_unpredictable_unaligned_to_non_writeback

TAGS **UNPREDICTABLE** DISPLAY Unaligned Access to Non-Writeback-Cacheable Memory.

ArchMsg.Warning.warning_unpredictable_vmsa_memattrib

TAGS **UNPREDICTABLE** DISPLAY Unpredictable vmsa memory attribute with texcb value programmed to %{TEXCB} when tex remap is %{TEX_REMAP}. Fields:

TEXCB unsigned int

texcb value.

TEX_REMAP bool

use tex remap.

ArchMsg.Warning.warning_unpredictable_vtcr_t0sz_sl0

TAGS **UNPREDICTABLE** DISPLAY The combination of VTCR.T0SZ=%{T0SZ:d} and VTCR.SL0=%{SL0:d} is unpredictable. Fields:

SL0 signed int

VTCR.SL0, starting level for stage 2 translation table walks.

T0SZ signed int

VTCR.T0SZ, required input address range.

ArchMsg.Warning.warning_unpredictable_write_DHCSR

TAGS **UNPREDICTABLE** DISPLAY **UNPREDICTABLE** set of bit-changes in DHCSR. Fields:

NEW unsigned int

data attempted to be written.

OLD unsigned int

previous value.

ArchMsg.Warning.warning_unpredictable_write_SHCSR

TAGS **UNPREDICTABLE** DISPLAY **UNPREDICTABLE** setting a pending bit to 1 for an exception with priority >= execution priority.

ArchMsg.Warning.warning_unsupported_access_to_memory_mapped_register

TAGS **UNPREDICTABLE** DISPLAY Access to memory mapped register is unsupported.

ArchMsg.Warning.warning_user_jmcr_access

TAGS UNPREDICTABLE DISPLAY User %{WRITE:(write|read)} access to JMCR is unpredictable.
Fields:

WRITE unsigned int
access is write.

ArchMsg.Warning.warning_wcr_mask_and_bas

TAGS UNPREDICTABLE DISPLAY DBGWCR_EL1.MASK is non-zero and BAS is not 0xff.

ArchMsg.Warning.warning_wcr_mask_reserved

TAGS UNPREDICTABLE DISPLAY DBGWCR_ELn.MASK is set to a reserved value.

ArchMsg.Warning.warning_wcr_non_configuous_bas

TAGS UNPREDICTABLE DISPLAY DBGWCR_EL1.BAS has non-contiguous set of ones.

ArchMsg.Warning.warning_wvr_masked_bit_not_zero

TAGS UNPREDICTABLE DISPLAY DBGWVRn_EL1.VA contains a bit masked by
DBGWCRn_EL1.MASK that is not zero.

ArchMsg.warning_unpred_system_register_access

DISPLAY Unpredictable %{IS_WRITE:(read|write)} access to system register offset
%{OFFSET}. Fields:

IS_WRITE unsigned int
Write Not Read.

OFFSET unsigned int
Register Offset.

BRANCH_MISPREDICT

Simulating branch mispredict. Fields:

PC unsigned int
Origin address (or 0 if unavailable).

BRA_DIR

Direct branches, to immediate address. Fields:

CORE_NUM unsigned int
Core number in a multi processor.

INST_COUNT unsigned int
The core's instruction counter, starting at 1 for the first instruction.

ISET enum
The instruction set of the branch instruction.

IS_COND bool
Indicates if this is a conditional branch.

IS_HINTED bool
Whether the conditional instruction is hinted.

IS_ISB bool

Whether the instruction is ISB.

IS_LINK bool

Indicates if this is a branch with link.

PC unsigned int

The address of the branch instruction.

TARGET_ISET enum

The instruction set after the branch.

TARGET_PC unsigned int

The address the instruction branches to.

BRA_INDIR

Indirect branches, perhaps to a register. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISET enum

The instruction set of the branch instruction.

IS_COND bool

Indicates if this is a conditional branch.

IS_HINTED bool

Whether the conditional instruction is hinted.

IS_ISB bool

Whether the instruction is ISB.

IS_LINK bool

Indicates if this is a branch with link.

PC unsigned int

The address of the branch instruction.

TARGET_ISET enum

The instruction set after the branch.

TARGET_PC unsigned int

The address the instruction branches to.

BRA_INDIR_CCFAIL

Indirect branches fail, perhaps to a register.

BRA_RET_CCFAIL

Branch return fail, perhaps to a register.

CCFAIL

Conditional instruction condition check fail. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CCFAIL_UNC

Unconditional instruction condition check fail. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CCPASS

Conditional instruction condition check pass. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CCPASS_UNC

Unconditional instruction condition check pass. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CODE_CACHE_MAINT

Code cache maintenance. Fields:

END_INCL unsigned int

Inclusive end address for VA or PA requests.

NS bool

Non-secure world for PA.

START unsigned int

Start address for VA or PA requests.

TYPE enum

Request type.

COMPILE_BLOCK_END

Last instruction of basic block translated. Fields:

VADDR unsigned int

Address of next instruction after this basic block.

COMPILE_BLOCK_START

First instruction of basic block translated. Fields:

VADDR unsigned int

Address of first instruction in basic block.

COMPILE_INST

ARM instruction compiled. Fields:

DISASS string

Disassembly of the instruction.

ISSET enum

The instruction set of this instruction.

ITSTATE unsigned int

The ITSTATE current for the instruction.

OPCODE unsigned int

The opcode of the instruction.

PC unsigned int

The address of the instruction.

SIZE unsigned int

The size of the instruction in bytes.

CONTEXTIDR

Context ID Register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

NS bool

Secure or nonsecure banked register is accessed.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The new value written.

CONTEXT_SYNC

Called for every context synchronization event. Fields:

ARCH_CSE bool

Architectural Context Synchronization Event.

PC unsigned int

The address of the synchronization instruction.

SYSREG_SYNC bool

System register synchronization event.

CORE_ENDIAN

Core BE8 Big-Endian state changed. Fields:

BE8 bool

Core BE8 Big-Endian state.

CORE_INFO

Static processor attributes. Only triggered by a call to DumpState(). Fields:

ARCH_PROFILE enum

The architecture profile of the core.

CLUSTER_ID unsigned int

The cluster ID of this processor.

CORE_NUM unsigned int

The number of this core in an MP processor.

DETAILED_CORE_FEATURES enum

Detailed core features as a bitfield. Enum values are bitmasks to extract the info from the field.

FPU_VERSION enum

The VFP version implemented by the core.

MEM_ARCH enum

The memory architecture of the core.

MIDR_EL1 unsigned int

MIDR_EL1 value.

NUM_CORES unsigned int

The number of cores in this MP processor.

QUANTUM_SIZE unsigned int

The default quantum size of the core (in ticks).

SECURITY_FEATURES bool

Does the core have security features?.

CORE_LOADS

Processor load accesses. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ACQREL enum

Is this an acquire/release.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DATA unsigned int

The data read or written.

ELEMENT_SIZE unsigned int

Width of each element.

ENDIANNESS enum

Byte ordering of this access.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

LOCK enum

Normal, exclusive or locked access.

MECID unsigned int

Memory Encryption Context ID, if applicable.

MECID2 unsigned int

Memory Encryption Context ID of the second page, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PAGE_SIZE unsigned int

Page size as $\log_2(\text{size in bytes})$.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK, 2=Exclusive Failed, 3=Unsupported.

SIZE unsigned int

Width of the access in bytes. Only required if DATA is not traced.

TRANS bool

Is this a translated access.

VADDR unsigned int

The virtual address of the access.

CORE_REGS

Changes of the core registers R0 to R14. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number, 0 to 14.

OLD_VALUE unsigned int

The old value overwritten.

PHYS_ID enum

The physical register accessed.

VALUE unsigned int

The new value written to the register.

CORE_STALLED_TRANSITION

Detect when a core is prevented from progressing PChannel states. Fields:

CURRENT_STATE enum

Current power mode of core.

REASON string

The reason a transition was stalled, and what must be done to unblock it.

TARGET_STATE enum

Desired power mode of core after transition.

CORE_STORES

Processor store accesses. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ACQREL enum

Is this an acquire/release.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DATA unsigned int

The data read or written.

ELEMENT_SIZE unsigned int

Width of each element.

ENDIANNESS enum

Byte ordering of this access.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

LOCK enum

Normal, exclusive or locked access.

MECID unsigned int

Memory Encryption Context ID, if applicable.

MECID2 unsigned int

Memory Encryption Context ID of the second page, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PAGE_SIZE unsigned int

Page size as $\log_2(\text{size in bytes})$.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK, 2=Exclusive Failed, 3=Unsupported.

SIZE unsigned int

Width of the access in bytes. Only required if DATA is not traced.

TRANS bool

Is this a translated access.

VADDR unsigned int

The virtual address of the access.

CRYPTO_SPEC

Every crypto instruction speculatively executed.

DATA_CACHE_SET_ALLOC_TAGS

Set allocation tags from DC GVA instruction. Fields:

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

NSDESC unsigned int

The security state of the access.

PADDR unsigned int

The physical address of the access.

PAS enum

The physical address space of the page.

RESPONSE enum

0=Aborted, 1=OK.

SIZE unsigned int

The size of memory in bytes.

TAG unsigned int

The value of the tag being set.

TAGGING_TYPE enum

Type of Memory tagging (Physical or Virtual).

TAG_VADDR unsigned int

The virtual address of the Virtual Tag if Virtual tagging is used; otherwise 0.

VADDR unsigned int

Virtual address used for setting allocation tags.

DATA_CACHE_ZERO

Zero and invalidate cache line from DC ZVA instruction. Fields:

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

NSDESC unsigned int

The security state of the access.

PADDR unsigned int

The physical address of the access.

PADDR_TAG unsigned int

If tag checked, the physical address tag for the access.

PAS enum

The physical address space of the page.

RESPONSE enum

0=Aborted, 1=OK.

SIZE unsigned int

The size of zero'd memory in bytes.

TAG_CHECKED bool

Is this access tag checked.

VADDR unsigned int

Virtual address of the zero'd cache line.

DEBUG_EVENT

Hardware debug support event. Fields:

EVENT enum

Description of event.

VALUE unsigned int

data value.

DMI_ALLOCATE

DMI allocated. Fields:

ACCESS_DIRECTION enum

READ / WRITE.

PADDR unsigned int

The physical address of the DMI.

PTR unsigned int

DMI range pointer.

SIDE enum

DSIDE / ISIDE.

SIZE unsigned int

The size of DMI.

VADDR unsigned int

The virtual address of the DMI..

DMI_HIT

DMI hit, access used DMI (only fire for small DMI regions i.e. less than a full page). Fields:

ACCESS_DIRECTION enum

READ / WRITE.

PADDR unsigned int

The physical address of the DMI.

SIDE enum

DSIDE / ISIDE.

SIZE unsigned int

The size of the DMI.

VADDR unsigned int

The virtual address of the DMI..

DMI_REVOKE

DMI revoked. Fields:

ACCESS_DIRECTION enum

READ / WRITE.

PADDR unsigned int

The physical address of the DMI.

SIDE enum

DSIDE / ISIDE.

SIZE unsigned int

The size of DMI.

VADDR unsigned int

The virtual address of the DMI..

DWT_MATCH

DWT comparator matches. Fields:

NUM unsigned int

DWT comparator number.

TYPE enum

DWT comparator configuration.

END_COMPILE

Compilation end. Fields:

END_OF_PAGE_COUNT unsigned int

Number of basic blocks exited due to page boundary since START_COMPILE.

FETCHFAIL_COUNT unsigned int

Number of basic blocks exited due to ifetch failure START_COMPILE.

INST_COUNT unsigned int

Number of instructions compiled since START_COMPILE.

NONSEQ_COUNT unsigned int

Number of basic blocks exited due to non-sequential instructions since START_COMPILE.

PAGE_STRADDLE_COUNT unsigned int

Number of basic blocks exited due to unaligned instructions crossing page since START_COMPILE.

EXCEPTION

Exceptions that are taken. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ISSET enum

The instruction set of the processor when the exception occurred.

LR unsigned int

The value assigned to the link register.

PC unsigned int

The location where the exception occurred.

TARGET_ISET enum

The instruction set of the exception handler code.

TARGET_PC unsigned int

The address the exception branches to.

VECTOR enum

The exception vector.

EXCEPTION_END

Every exception completed.

EXCEPTION_ENTRY

Event marking the entry of an exception. All pushes of registers to the stack will follow this event. Fields:

PC unsigned int

The location where the exception occurred.

VECTOR enum

The exception that occurred.

EXCEPTION_RAISE

Every exception raised.

EXCEPTION_RETURN

Marks the end of an exception, but on an M core does not indicate a program flow change (branch). In most situations it will be followed by an indirect branch. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISSET enum

The instruction set of the branch instruction.

PC unsigned int

The address of the branch instruction.

EXCEPTION_RETURN_PREBRANCH

Exception return event that must occur before PC and processor state have updated.

EXCEPTION_START

Every exception started.

EXCEPTION_VECTOR_FETCH

I-side vector fetch for M-class exception. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

DATA unsigned int

Result of fetch if OK.

RESPONSE enum

0=Aborted, 1=OK.

VADDR unsigned int

Where the vector is fetched from (including VTOR).

FP_STATE

Floating point state. Fields:

FPSR unsigned int

FPSR register value.

MASK unsigned int

The mask indicating updated cumulative bits.

FREQ_CHANGED

Core frequency changed. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The instruction count of this core.

NEW_FREQ signed int

The new frequency of this core (expressed in Hz).

OLD_FREQ signed int

The old frequency of this core (expressed in Hz).

HLT

HLT instruction occurred. Fields:

IMM16 unsigned int

The 16-bit immediate value encoded in HLT instruction.

INFO_EXCEPTION_REASON

Provide information of real cause of exception. Fields:

FaultCause enum

Which bit of which V7M/V8M-mainline FSR/HFSR would have been set.

PC unsigned int

The location where the exception occurred.

PHASE enum

What we are doing with the exception.

REASONS enum

Why we are doing it.

VECTOR enum

The exception that occurred.

INFO_STACKING

Provide information of real cause of exception. Fields:

PHASE enum

Are we starting or ending a sequence of stack writes.

REASONS enum

Why we are doing it.

INST

Every instruction executed. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DEBUG_STATE bool

The instruction is executed in debug state.

DISASS string

Disassembly of instruction.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISSET enum

The current instruction set.

ITSTATE unsigned int

The current ITSTATE.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

MODE enum

The mode the core is in.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

OPCODE unsigned int

The opcode of the instruction.

PADDR unsigned int

The physical address of the instruction.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the instruction.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the page.

PC unsigned int

The address of the instruction.

SECURITY_STATE enum

The core's security state.

SIZE unsigned int

The size of the instruction in bytes.

INST_END

Every instruction completed.

INST_START

Every instruction started. Fields:

DEBUG_STATE bool

The instruction is executed in debug state.

ISSET enum

The current instruction set.

MODE enum

The mode the core is in.

NS enum

The current Secure State.

PC unsigned int

The address of the conditional instruction.

SECURITY_STATE enum

The current Security State.

INST_STRADDLE

Instruction straddles boundary. Fields:

DEBUG_STATE bool

The instruction is executed in debug state.

ISET enum

The current instruction set.

MODE enum

The mode the core is in.

NS enum

The current Secure State.

PC unsigned int

The address of the conditional instruction.

SECURITY_STATE enum

The current Security State.

ISB

ISB instruction occurred. Fields:

PC unsigned int

The address of the ISB instruction.

LOAD_MULTIPLE_REGS_LIST

List of destination registers (not source) of a multiple registers load instruction. Fields:

REG string

Register to add into the list.

LOCAL_MONITOR

Local monitor activity. Fields:

PADDR unsigned int

Local Monitor Address.

State enum

State of the monitor (Open/Exclusive).

LOCKUP_CYCLE

This event is triggered if the core enters or stays in the lock-up state. Fields:

CAUSE enum

What causes the lockup?.

LOCKUP_ENTRY

This event is triggered if the core enters the lock-up state. Fields:

PC unsigned int

The PC when the lockup state is entered.

LOCKUP_EXIT

This event is triggered if the core leaves the lock-up state.

MODE_CHANGE

Mode change. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

MODE enum

The new mode.

OLD_MODE enum

The old mode.

MPU_TRANS

Address translation information. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

EXEC_PERM enum

Execution Permission.

INNERCACHE_RA bool

Is the inner cache allocate on read.

INNERCACHE_TYPE enum

Inner Caching scheme (NC/MB/WA).

INNERCACHE_WA bool

Is the inner cache allocate on write.

MEMTYPE enum

Memory type.

OUTERCACHE_RA bool

Is the outer cache allocate on read.

OUTERCACHE_TYPE enum

Outer Caching scheme (NC/MB/WA).

OUTERCACHE_WA bool

Is the outer cache allocate on write.

PADDR unsigned int

Address of the access.

PAGESIZE unsigned int

Page size as $\log_2(\text{size})$.

READ_PERM enum

Read Permission.

REG_NUM unsigned int

MPU register based region-number 0-15 (else 0xFF=none, 0xBB/0xDD=background/default).

SH enum

Shareability.

SIDE enum

Inst / Data.

WRITE_PERM enum

Write Permission.

PERIODIC

Called for every quantum. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The instruction count of this CPU.

PC unsigned int

The address of the next instruction to be executed on this CPU.

POTINDEX_UPDATE

Value of POT index for current pc changed. Fields:

NEW_POTINDEX unsigned int

New POTIndex.

OLD_POTINDEX unsigned int

Old POTIndex.

PPB_READ_ACCESS

Trace reads from the core to memory mapped registers in the PPB (private peripheral bus) address range. Fields:

ACCESS_SIZE unsigned int

Logarithm of the access size: 0=byte, 1=halfword, 2=word, 3=doubleword, ...

BURST_LEN unsigned int

The number of data transfers (beats) in this burst.

DATA unsigned int

The data transferred.

INNER_ATTRIBUTES unsigned int

The inner attributes: bit0=write allocate, bit1=read allocate, bit2=cachable, bit3=bufferable, bit4=shareable.

INST bool

Is an instruction fetch.

LOCK enum

Normal / Locked / Exclusive.

MANAGER_ID unsigned int

The AXI manager ID.

NS bool

Is a nonsecure access.

OUTER_ATTRIBUTES unsigned int

The outer attributes: bit0=write allocate, bit1=read allocate, bit2=cachable, bit3=bufferable, bit4=shareable.

PADDR unsigned int

Physical address of access.

PRIV bool

Is privileged access.

RESPONSE enum

Whether the transaction was successful, or an error occurred.

USER_FLAGS unsigned int

Core specific additional signals.

PPB_WRITE_ACCESS

Trace writes from the core to memory mapped registers in the PPB (private peripheral bus) address range. Fields:

ACCESS_SIZE unsigned int

Logarithm of the access size: 0=byte, 1=halfword, 2=word, 3=doubleword, ...

BURST_LEN unsigned int

The number of data transfers (beats) in this burst.

DATA unsigned int

The data transferred.

INNER_ATTRIBUTES unsigned int

The inner attributes: bit0=write allocate, bit1=read allocate, bit2=cachable, bit3=bufferable, bit4=shareable.

INST bool

Is an instruction fetch.

LOCK enum

Normal / Locked / Exclusive.

MANAGER_ID unsigned int

The AXI manager ID.

NS bool

Is a nonsecure access.

OUTER_ATTRIBUTES unsigned int

The outer attributes: bit0=write allocate, bit1=read allocate, bit2=cachable, bit3=bufferable, bit4=shareable.

PADDR unsigned int

Physical address of access.

PRIV bool

Is privileged access.

RESPONSE enum

Whether the transaction was successful, or an error occurred.

USER_FLAGS unsigned int

Core specific additional signals.

PRELOAD_DATA

Data preload from PLD instruction. Fields:

VADDR unsigned int

Virtual address of the data that should be preloaded.

PRELOAD_INST

Instruction preload from PLI instruction. Fields:

VADDR unsigned int

Virtual address of the instruction that should be preloaded.

PRE_CORE_LOAD

Trace just before a core load. Fields:

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

PRE_CORE_STORE

Trace just before a core store. Fields:

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

PSTATE_PACTIVE_OPMODE_TRANSITION

Trace pactive and Opmode mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Operating mode after the transition.

OLD_STATE enum

Operating mode before the transition.

PSTATE_PACTIVE_OPMODE_TRANSITION_DENIED

Trace the denied pactive and Opmode mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Operating mode after the transition.

OLD_STATE enum

Operating mode before the transition.

REASON string

Reason for the transition being denied.

PSTATE_PACTIVE_POWER_TRANSITION

Trace pactive and power mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Power mode after the transition.

OLD_STATE enum

Power mode before the transition.

PSTATE_PACTIVE_POWER_TRANSITION_DENIED

Trace the denied pactive and power mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Power mode after the transition.

OLD_STATE enum

Power mode before the transition.

REASON string

Reason for the transition being denied.

PSTATE_PACTIVE_THREAD_OPMODE_TRANSITION

Trace pactive and Opmode mode transitions for threads on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Operating mode after the transition.

OLD_STATE enum

Operating mode before the transition.

QUANTUM_START

Called at the start of every quantum.

RUN_STATE

Run state transition. Fields:

INST_COUNT unsigned int

Ticks count at point of transition.

NEW enum

New run state.

OLD enum

Old run state.

SEMIHOSTING_CALL

Call of a semihost function occurred. Fields:

PC unsigned int

The program counter after the semihosting call.

SEMIHOSTING_PRECALL

About to call semihost. Fields:

EL2 bool

Use EL2 translation regime.

NS bool

Is Non-Secure.

REG_WIDTH unsigned int

The current register width in bytes.

SIGNAL

External signal state change. Fields:

SIGNAL enum

Signal that changed.

STATE bool

Signal asserted state.

START_COMPILE

Compilation started. Fields:

VADDR unsigned int

Instruction where compilation begins.

STG_MULTIPLE_MICRO_OPS_ADDRESS_RANGE

Address range for MTE stores with multiple uops. Fields:

ADDR_END unsigned int

Final address.

ADDR_START unsigned int

Starting address.

STORE_MULTIPLE_REGS_ADDRESS_RANGE

Address range for stores with multiple registers. Fields:

ADDR_END unsigned int

Final address.

ADDR_START unsigned int

Starting address.

SUBOPTIMAL_LDST_RETIRED

suboptimal load/store. Fields:

ACCESS_SIZE unsigned int

Log2 of the access width in bytes used for alignment checking (1: half-word, 2: word ...).

OPTIMAL_ALIGN_SELECTOR unsigned int

Selector to decide the optimal alignment size (1: access width, 2: 4-bytes, 3: 8-bytes, ..., 12: 4-kbytes).

VADDR unsigned int

The virtual address of the access.

SYNC

Called for every synchronization, that is every quantum break. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The instruction count of this CPU.

LOCAL_QUANTUM unsigned int

The local quantum of this CPU.

LOCAL_TIME unsigned int

The local time of this CPU.

SYNC_REASON enum

Reason for sync.

SYSCALL

System call instruction executed. Fields:

IMM unsigned int

Immediate value of the system call instruction.

TYPE enum

System call type.

VADDR unsigned int

Instruction that caused the system call.

SYSREG_READ

Trace reads from system registers. Fields:

REG_NAME enum

Register number.

VALUE unsigned int

The value read.

SYSREG_UPDATE128

Triggers when the system updates a register. Fields:

REG enum

Register number.

UNKNOWN unsigned int

Bits of the register which became unknown.

VALUE unsigned int

Value written to the register.

SYSREG_UPDATE32

Triggers when the system updates a register. Fields:

REG enum

Register number.

UNKNOWN unsigned int

Bits of the register which became unknown.

VALUE unsigned int

Value written to the register.

SYSREG_UPDATE64

Triggers when the system updates a register. Fields:

REG enum

Register number.

UNKNOWN unsigned int

Bits of the register which became unknown.

VALUE unsigned int

Value written to the register.

SYSREG_WRITE

Trace write to system registers. Fields:

REG_ACCESS enum

Register access status.

REG_NAME enum

Register number.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

Value written to the register.

UNALIGNED_LDST_RETIRED

Processor unaligned load/store. Fields:

ACCESS_SIZE unsigned int

Log2 of the access width in bytes used for alignment checking (1=>half-word, 2->word ...).

NUMBER_OF_BEATS unsigned int

Number of accesses (beats) in this burst (Total size in bytes is ACCESS_SIZE * NUMBER_OF_BEATS).

VADDR unsigned int

The virtual address of the access.

WAYPOINT

Signals end of basic block execution. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ISET enum

Origin instruction set.

IS_COND bool

Indicates if this is a conditional waypoint.

PC unsigned int

Origin address (or 0 if unavailable).

TAKEN bool

Indicates if this waypoint was taken.

TARGET unsigned int

Destination address.

TARGET_ISET enum

Destination instruction set.

WFE_END

WFE ended. Fields:

INST_COUNT unsigned int

Ticks count when leaving WFE.

WFE_EVENT_REGISTER

WFE event register status: set/clear, reason. Fields:

INST_COUNT unsigned int

Ticks count.

REASON enum

Reason for set/clear. Only REASON==0 (Cleared by WFE) clears the bit, all other reasons set it.

WFE_IGNORED

WFE ignored. Fields:

EVENT bool

This WFE was ignored because the event register was set.

INST_COUNT unsigned int

Ticks count when ignoring WFE.

TRAPPED bool

This WFE was trapped.

WFE_START

WFE entered. Fields:

INST_COUNT unsigned int

Ticks count when entering WFE.

WFI_END

WFI ended. Fields:

INST_COUNT unsigned int

Ticks count when leaving WFI.

WFI_IGNORED

WFI ignored. Fields:

DISABLED bool

This WFI was ignored because WFI is disabled.

INST_COUNT unsigned int

Ticks count when ignoring WFI.

REASON enum

specifies reason why WFI trace was ignored.

TRAPPED bool

This WFI was trapped.

WFI_START

WFI entered. Fields:

INST_COUNT unsigned int

Ticks count when entering WFI.

WFI_WAKEUP

WFI wakeup. Fields:

INST_COUNT unsigned int

Ticks count when WFI wakeup occurred.

REASON enum

Reason for wakeup.

XPSR

Changes to the xPSR register. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

OLD_VALUE unsigned int

The old xPSR value.

UNKNOWN unsigned int

Bits within the register that have unknown value.

VALUE unsigned int

The new xPSR value.

2.39 ARM_Cortex-M0+

This section describes the trace sources.

ASYNC_MEMORY_FAULT

Context ID Register write. Fields:

FAULT unsigned int

Fault status present in ESR.ISS encoding for Serror in ARCH64 and present in DFSR encoding for external abort format in ARCH32.

PADDR unsigned int

Physical Address (or 0 if unavailable).

ATOMIC_END_ACCESS

Bus trace access for atomics. Fields:

ACCESS_FAIL bool

Memory access failed.

ADDR unsigned int

The virtual address of the access.

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

COMPARE_VALUE unsigned int

Compare value for CAS.

LOAD_VALUE unsigned int

Loaded values.

NSDESC unsigned int

The security state of the access.

OPERAND_VALUE unsigned int

Operation's operand.

OPERATION enum

Operation type.

PADDR unsigned int

The physical address of the access.

PRIV bool

Is this a privileged access?.

ATOMIC_START_ACCESS

Bus trace access for atomics. Fields:

ADDR unsigned int

The virtual address of the access.

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

COMPARE_VALUE unsigned int

Compare value for CAS.

NSDESC unsigned int

The security state of the access.

OPERAND_VALUE unsigned int

Operation's operand.

OPERATION enum

Operation type.

PADDR unsigned int

The physical address of the access.

PRIV bool

Is this a privileged access?.

ArchMsg.Error.exit_code

DISPLAY exit code: %{EXITCODE} component kind: %{KIND} reason: %{REASON}. Fields:

EXITCODE signed int

Exit code to be returned.

KIND string

Component kind that invoked the exit code trace.

REASON string

Reason why the exit code trace was invoked (optional).

ArchMsg.Warning.Unpredictably Indexed PM Event Register

TAGS **UNPREDICTABLE** DISPLAY Event %{IsDirect:(Selector value|Register)} %{SEL} is >= %{N}. Fields:

IsDirect unsigned int

Direct Access rather than indirect.

N unsigned int

Number of Accessible Registers.

SEL unsigned int

Selector.

ArchMsg.Warning.dap_csw_bad_size

A write to CM3DAP CSW has an invalid size field. DISPLAY Write %{DATA} to CM3DAP CSW has invalid size field. Fields:

DATA unsigned int

bits[2:0] are size.

ArchMsg.Warning.dcimvac_matches_watchpoint

TAGS **UNPREDICTABLE** DISPLAY Watchpoints matching an AArch32 DCIMVA are implementation defined.

ArchMsg.Warning.decode_fieldmismatch

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with different fields %{FIELD1} and %{FIELD2}. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_fracbitsnegative

TAGS **UNPREDICTABLE** DISPLAY VCVT Instruction is unpredictable with a negative number of fraction bits.

ArchMsg.Warning.decode_initblock

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable in an IT block.

ArchMsg.Warning.decode_initblocknotlast

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable in an IT block when not the last.

ArchMsg.Warning.decode_invalid_condition

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable when the condition is not AL.

ArchMsg.Warning.decode_invalidfieldcombination

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with the values in %{FIELD1} and %{FIELD2}. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_invalidvalue

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with the value in %{FIELD}. Fields:

FIELD enum

Field name.

ArchMsg.Warning.decode_mem_hint_unallocated

TAGS **UNPREDICTABLE** DISPLAY There is no memory hint allocated to this bit pattern.

ArchMsg.Warning.decode_registermatch

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with %{FIELD1} and %{FIELD2} as the same register. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_registermismatch

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with %{FIELD1} and %{FIELD2} as different registers. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_registeroutofrange

TAGS UNPREDICTABLE DISPLAY Floating-point instruction is unpredictable indexing beyond the end of the register bank.

ArchMsg.Warning.decode_sbzsbo

TAGS UNPREDICTABLE DISPLAY Reserved bits in the instruction are not canonical.

ArchMsg.Warning.decode_transactiontoobig

TAGS UNPREDICTABLE DISPLAY Attempt to perform a memory transaction of more than 128 bytes.

ArchMsg.Warning.decode_unpred_other

TAGS UNPREDICTABLE DISPLAY Instruction is unpredictable.

ArchMsg.Warning.decode_unpreduseofpc

TAGS UNPREDICTABLE DISPLAY Instruction is unpredictable when %{FIELD} is PC. Fields:

FIELD enum

Field name.

ArchMsg.Warning.decode_unpreduseofr13

TAGS UNPREDICTABLE DISPLAY Instruction is unpredictable when %{FIELD} is R13. Fields:

FIELD enum

Field name.

ArchMsg.Warning.decode_unpreduseofr13inreglist

TAGS UNPREDICTABLE DISPLAY Instruction is unpredictable with R13 in register-list.

ArchMsg.Warning.decode_useofd16orhigher

TAGS UNPREDICTABLE DISPLAY Attempt to use register D16 or higher where it is not allowed.

ArchMsg.Warning.decode_writebackandbaseinlist

TAGS UNPREDICTABLE DISPLAY Instruction is unpredictable with write back when the base register is in the transfer list.

ArchMsg.Warning.decode_zeroregistersinlist

TAGS UNPREDICTABLE DISPLAY Instruction is unpredictable when no registers are to be transferred.

ArchMsg.Warning.recursive_branch

An instruction is performing a branch that targets the same instruction. If this was intended then a WFI might be more effective DISPLAY recursive branch detected at PC=%{PC}.

PRIMARY KEY PC. Fields:

PC unsigned int

Address of instruction causing the branch.

ArchMsg.Warning.recursive_exception

An instruction has generated an exception that targets the same instruction. DISPLAY recursive %{TYPE} exception detected at PC=%{PC} (CPSR=%{CPSR}). PRIMARY KEY PC. Fields:

CPSR unsigned int

Processor state.

PC unsigned int

Address of instruction causing the exception.

TYPE enum

Type of exception.

ArchMsg.Warning.reentrant_vector_catch

TAGS **UNPREDICTABLE** DISPLAY Vector catch debug event on Prefetch or Data abort vector %{ADDR}. Fields:

ADDR unsigned int

vector address.

ArchMsg.Warning.reserved_it_state

TAGS **UNPREDICTABLE** DISPLAY Execution with reserved IT state %{ITSTATE}. Fields:

ITSTATE unsigned int

reserved IT state.

ArchMsg.Warning.secure_vector_fetch_from_nonsecure

DISPLAY Secure vector table is being fetched from Non-secure memory.

ArchMsg.Warning.sg_in_it_block

Arm recommends that software does not place SG (security gate instruction) inside an IT instruction block.

ArchMsg.Warning.unpred_FAR_on_LRCPC3_non_atomic_mem_access

TAGS **UNPREDICTABLE** Unpredictable FAR/FSTATUS on LRCPC3 pair instructions resulting in two separate memory accessesDISPLAY Unpredictable FAR on LRCPC3 non atomic memory access at %{ADDR1} with FSTATUS %{FSTATUS1} or %{ADDR2} with FSTATUS %{FSTATUS2}. Fields:

ADDR1 unsigned int

Possible FAR address 1.

ADDR2 unsigned int

Possible FAR address 2.

FSTATUS1 unsigned int

Possible fault status value 1.

FSTATUS2 unsigned int

Possible fault status value 2.

ArchMsg.Warning.unpred_async_tag_check_fail_on_sve_speculative_load_do_update_ffr

Unpredictable set to FALSE of the FFR element associated to SVE speculative load with an async tag check fail TAGS **UNPREDICTABLE**.

ArchMsg.Warning.unpredictable_a32_breakpoint

TAGS **UNPREDICTABLE** DISPLAY Unpredictable breakpoint on a A32 instruction at %{ADDR1} Offset %{OFFSET} BAS %{BAS} for a %{IS_ADDRESS_MISMATCH_BREAKPOINT:(Mismatch|Match)} Breakpoint number %{N:d} Breakpoint address %{ADDR2}. Fields:

ADDR1 unsigned int

Address.

ADDR2 unsigned int

Breakpoint address.

BAS unsigned int

BAS field.

IS_ADDRESS_MISMATCH_BREAKPOINT bool

Is a Mismatch breakpoint, not a Match breakpoint.

N unsigned int

Breakpoint number.

OFFSET unsigned int

Always has a value of zero.

ArchMsg.Warning.unpredictable_banked_register_access

TAGS **UNPREDICTABLE** DISPLAY Access to banked register (R=%{R:d} SYSm=%{SYSM:2x}) is unpredictable. Fields:

R unsigned int

R bit.

SYSM unsigned int

SYSm.

ArchMsg.Warning.unpredictable_ccfail_undef

TAGS **UNPREDICTABLE** DISPLAY An instruction %{OPCODE} at VA=%{VADDR} failed its condition codes check but would otherwise have caused a non-data-dependent trap or undefined exception. Behaviour is implementation-defined choice of exception or nop. Fields:

OPCODE unsigned int

opcode of the instruction.

VADDR unsigned int

Virtual address of the instruction being executed.

ArchMsg.Warning.unpredictable_ici_bits

TAGS **UNPREDICTABLE** DISPLAY ICI bits do not apply or are architecturally unpredictable for instruction.

ArchMsg.Warning.unpredictable_t32_breakpoint

TAGS **UNPREDICTABLE** DISPLAY Unpredictable breakpoint on a T32 instruction at %{ADDR1} Offset %{OFFSET} BAS %{BAS} for a %{IS_ADDRESS_MISMATCH_BREAKPOINT:(Mismatch|Match)} Breakpoint number %{N:d} Breakpoint address %{ADDR2}. Fields:

ADDR1 unsigned int

Address.

ADDR2 unsigned int

Breakpoint address.

BAS unsigned int

BAS field.

IS_ADDRESS_MISMATCH_BREAKPOINT bool

Is a Mismatch breakpoint, not a Match breakpoint.

N unsigned int

Breakpoint number.

OFFSET unsigned int

Indicates which halfword of a 32-bit Thumb instruction the breakpoint is set on. 0 for the first halfword or if the instruction is 16 bits wide, or 1 for the second halfword.

ArchMsg.Warning.unpredictable_t32_vector_catch

TAGS **UNPREDICTABLE** DISPLAY Unpredictable Vector catch exception on a T32 instruction at %{ADDR} Offset %{OFFSET}. Fields:

ADDR unsigned int

Address.

OFFSET unsigned int

Indicates which halfword of a 32-bit Thumb instruction the vector catch exception is set on. 0 for the first halfword or if the instruction is 16 bits wide, or 1 for the second halfword.

ArchMsg.Warning.unpredictable_watchpoint_far

TAGS **UNPREDICTABLE** DISPLAY The value for FAR/EDWAR for hitting this watchpoint may be anywhere in the range %{LOWER_BOUND} to %{UPPER_BOUND}. Fields:

LOWER_BOUND unsigned int

the lowest address accessed by the instruction that triggered the watchpoint.

UPPER_BOUND unsigned int

the highest watchpointed address accessed by that instruction.

ArchMsg.Warning.warning_AdvSIMDExpandImmUnexpectedZero

TAGS **UNPREDICTABLE** DISPLAY AdvSIMDExpandImm may treat this immediate value as **UNPREDICTABLE**.

ArchMsg.Warning.warning_ConditionalSMC

TAGS **UNPREDICTABLE** DISPLAY SMC instruction has **UNPREDICTABLE** effects when conditional and when combined with traps.

ArchMsg.Warning.warning_access_crosses_page_boundary

TAGS **UNPREDICTABLE** DISPLAY Access crosses page boundary.

ArchMsg.Warning.warning_access_crosses_page_boundary_has_fault

TAGS **UNPREDICTABLE** DISPLAY Access that crosses page boundary has fault(s).

ArchMsg.Warning.warning_access_crosses_page_boundary_has_fault_on_both_pages

TAGS **UNPREDICTABLE** DISPLAY Access that crosses page boundary has a fault on each page.
Fields:

ADDR1 unsigned int

FAR address on first page.

ADDR2 unsigned int

FAR address on second page.

FSTATUS1 unsigned int

Fault status on first page.

FSTATUS2 unsigned int

Fault status on second page.

ArchMsg.Warning.warning_access_crosses_page_boundary_spanning_different_memory

TAGS **UNPREDICTABLE** DISPLAY Access at address %{ADDR} crosses page from MEM_TYPE %{MEMTYPE_PAGE1} to %{MEMTYPE_PAGE2}. Fields:

ADDR unsigned int

address of access crossing page.

MEMTYPE_PAGE1 signed int

First Page's memory type (0-Normal, 1-Non-Normal).

MEMTYPE_PAGE2 signed int

Second Page's memory type (0-Normal, 1-Non-Normal).

ArchMsg.Warning.warning_access_wraps_around_memory

TAGS **UNPREDICTABLE** DISPLAY Access wraps around memory in %{WIDTH:d}bit mode.
Fields:

WIDTH unsigned int

Address width.

ArchMsg.Warning.warning_bcr_linking_status

TAGS **UNPREDICTABLE** DISPLAY DBGBCR.{BT2,BT} combination gives unpredictable behaviour.

ArchMsg.Warning.warning_bcr_mask_reserved

TAGS **UNPREDICTABLE** DISPLAY DBGBCR_ELn.MASK is set to a reserved value.

ArchMsg.Warning.warning_bvr_masked_bit_not_zero

TAGS **UNPREDICTABLE** DISPLAY DBGBVRn_EL1.VA contains a bit masked by DBGBCRn_EL1.MASK that is not zero.

ArchMsg.Warning.warning_ccsidr_unimplemented_level

TAGS **UNPREDICTABLE** DISPLAY CCSIDR read while CSSELR %{CSSELR:x} points to an unimplemented cache level. Fields:

CSSELR unsigned int

current effective value of CSSELR.

ArchMsg.Warning.warning_change_to_ns_when_tge_set

TAGS **UNPREDICTABLE** DISPLAY Attempting to change NS to 1 when HCR.TGE = 1.

ArchMsg.Warning.warning_contiguous_bit_check_abort

TAGS **UNPREDICTABLE** DISPLAY An abort occurred whilst attempting to check TLB entry is contiguous at %{ENTRY_ADDR}. Fields:

ENTRY_ADDR unsigned int

address of conflicting TLB entry.

ArchMsg.Warning.warning_contiguous_bit_error

TAGS **UNPREDICTABLE** DISPLAY TLB entry at %{ENTRY_ADDR} was not contiguous with entry at %{CONTIG_ADDR} Contents are %{ENTRY_DATA} but expected %{CONTIG_DATA}. Fields:

CONTIG_ADDR unsigned int

address of the TLB first read, with which this entry is expected to be contiguous.

CONTIG_DATA unsigned int

expected contents based on the entry at CONTIG_ADDR.

ENTRY_ADDR unsigned int

address of conflicting TLB entry.

ENTRY_DATA unsigned int

contents of conflicting TLB entry.

ArchMsg.Warning.warning_contiguous_bit_error_D128

TAGS **UNPREDICTABLE** DISPLAY TLB entry at %{ENTRY_ADDR} was not contiguous with entry at %{CONTIG_ADDR} Contents are %{ENTRY_DATA_HI} %{ENTRY_DATA_LO} but expected %{CONTIG_DATA}. Fields:

CONTIG_ADDR unsigned int

address of the TLB first read, with which this entry is expected to be contiguous.

CONTIG_DATA unsigned int

expected contents based on the entry at CONTIG_ADDR.

ENTRY_ADDR unsigned int

address of conflicting TLB entry.

ENTRY_DATA_HI unsigned int

Higher bit contents of conflicting TLB entry.

ENTRY_DATA_LO unsigned int

Lower bit contents of conflicting TLB entry.

ArchMsg.Warning.warning_cp10_cp11_mismatch

TAGS **UNPREDICTABLE** DISPLAY Unpredictable configuration of CP10 and CP11 controls at EL
%{EL:d}. Fields:

EL unsigned int

EL owning the control.

ArchMsg.Warning.warning_cp10_cp11_reserved_value

TAGS **UNPREDICTABLE** DISPLAY Unpredictable configuration of CPACR CP10 (%{CP10}) or
CP11 (%{CP11}). Fields:

CP10 unsigned int

CP10 access.

CP11 unsigned int

CP11 access.

ArchMsg.Warning.warning_csselr_level_out_of_range

TAGS **UNPREDICTABLE** DISPLAY CSSELR.Level out of range, written value %{LEVEL} with only
%{IMPLEMENTED:d} levels implemented. Fields:

IMPLEMENTED unsigned int

number of cache levels implemented.

LEVEL unsigned int

written value of CSSELR.Level.

TYPE enum

The type of cache selected by CSSELR.InD and TnD.

ArchMsg.Warning.warning_debug_flow_control_bits_not_obeyed

TAGS **UNPREDICTABLE** DISPLAY Flow control bits for %{REGISTER} were not obeyed, causing
%{ERROR}. Fields:

ERROR enum

type of error resulting.

REGISTER enum

register under consideration.

ArchMsg.Warning.warning_debug_register_access_during_reset

TAGS **UNPREDICTABLE** DISPLAY Unpredictable %{IS_WRITE:(read|write)} access to debug
register offset %{OFFSET} during reset. Fields:

IS_WRITE unsigned int

Write Not Read.

OFFSET unsigned int

Register Offset.

ArchMsg.Warning.warning_decode_cps_inconsistent_fields

TAGS **UNPREDICTABLE** DISPLAY The mode setting fields in the CPS instruction are inconsistent.

ArchMsg.Warning.warning_decode_invalid_state

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable in current exception-level/security state.

ArchMsg.Warning.warning_default_cacheable_mmu_on

TAGS **UNPREDICTABLE** DISPLAY Default cacheable is enabled (HCR.DC = 1) while MMU is enabled.

ArchMsg.Warning.warning_default_cacheable_vmmu_off

TAGS **UNPREDICTABLE** DISPLAY Default cacheable is enabled (HCR.DC = 1) while HCR.VM = 0.

ArchMsg.Warning.warning_deprecated_wvr_bit_2_set

TAGS **UNPREDICTABLE** DISPLAY DBGWVR%{N:d}_EL1=%{WVR:h} has bit 2 set, deprecated in ARMv8. Fields:

N unsigned int

index.

WVR unsigned int

DBGWVR.

ArchMsg.Warning.warning_eret_while_software_step_active_pending

Missing ISB between setting MDSCR_EL1.SS and ERET TAGS **UNPREDICTABLE**.

ArchMsg.Warning.warning_exclusive_to_non_normal

TAGS **UNPREDICTABLE** DISPLAY Exclusive Access to Non-Normal Memory %{ADDR}. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_exclusive_to_non_writeback

TAGS **UNPREDICTABLE** DISPLAY Exclusive Access to Non-Writeback-Cacheable Memory %{ADDR}. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_execute_from_device_memory

TAGS **UNPREDICTABLE** DISPLAY An attempt was made to execute from Device memory.

ArchMsg.Warning.warning_illegal_cpsr_mode

TAGS **UNPREDICTABLE** DISPLAY Writing an illegal or unimplement mode (%{NEW_MODE:h}) to CPSR was unpredictable. Fields:

NEW_MODE unsigned int

New value of CPSR.M.

ArchMsg.Warning.warning_illegal_srs_mode

TAGS **UNPREDICTABLE** DISPLAY Illegal or **UNPREDICTABLE** mode (%{MODE:h}) used for SRS instruction. Fields:

MODE unsigned int

mode for Banked SP.

ArchMsg.Warning.warning_implementation_defined_read_debug_register_in_SCS

Read Access to Debug registers from the processor is **IMPLEMENTATION DEFINED** TAGS **IMP_DEF**. Fields:

PPB_OFFSET unsigned int

debug register offset.

ArchMsg.Warning.warning_implementation_defined_sequential_security_transitions_supported

The behavior of sequential instruction fetches that cross from non-secure to secure memory and contain an SG instruction is **CONSTRAINED UNPREDICTABLE** TAGS **UNPREDICTABLE**.

ArchMsg.Warning.warning_implementation_defined_stack_limit_check_supported

It is **IMPLEMENTATION DEFINED** whether stack pointer limit checking is performed for this instruction TAGS **IMP_DEF**. Fields:

ADDRESS unsigned int

Address of instruction.

OPCODE unsigned int

instruction opcode.

ArchMsg.Warning.warning_implementation_defined_taggednonsharedmemorytype

TAGS **IMP_DEF** DISPLAY Memory attribute has Tagged, non shared memory type; it is **IMPLEMENTATION DEFINED** whether the memory location is treated as Tagged or non-Tagged.

ArchMsg.Warning.warning_implementation_defined_write_debug_register_in_SCS

Write Access to Debug registers from the processor is **IMPLEMENTATION DEFINED** TAGS **IMP_DEF**. Fields:

DATA unsigned int

data attempted to be written.

PPB_OFFSET unsigned int

debug register offset.

ArchMsg.Warning.warning_invalid_tcr_granule

TAGS **UNPREDICTABLE** DISPLAY TCR.TG%{TG_ID:(0|1)} bits have been set to a granule size %{REQUEST:d}K not implemented - using %{SUBSTITUTE:d}K. Fields:

REQUEST signed int

page size requested (or 0 for reserved).

SUBSTITUTE signed int

best guess available page size.

TG_ID bool

bits TG0 or TG1.

ArchMsg.Warning.warning_load_multiple_user_registers_from_user_mode

TAGS UNPREDICTABLE DISPLAY An LDM(user registers) instruction executed from user mode.

ArchMsg.Warning.warning_mte_in_mem_attr_not_supported

TAGS UNPREDICTABLE DISPLAY Memory attribute %{ATTR} has Tagged memory type but Armv8.5-MTE is not implemented. Fields:

ATTR unsigned int

Memory attribute.

ArchMsg.Warning.warning_reserved_breakpoint_state_match

TAGS UNPREDICTABLE DISPLAY %{IS_BREAKPOINT:(Watchpoint|Breakpoint)} programmed with a reserved combination of HMC, SSC and %{IS_BREAKPOINT:(PAC|PMC)}. Fields:

IS_BREAKPOINT unsigned int

Is a breakpoint, not a watchpoint.

ArchMsg.Warning.warning_shareability

TAGS UNPREDICTABLE DISPLAY Unpredictable: combination of the 1st and 2nd stages of translation is Normal Inner Non-Cacheable, Outer Non-Cacheable.

ArchMsg.Warning.warning_software_step_set_while_enabled

TAGS UNPREDICTABLE DISPLAY MDSCR_EL1.SS set to 1 while software step debug exceptions are enabled.

ArchMsg.Warning.warning_thumb_instruction_wraps_around_memory

TAGS UNPREDICTABLE DISPLAY Thumb instruction wraps around memory.

ArchMsg.Warning.warning_ttbr_sbz_bits_are_not_zero

TAGS UNPREDICTABLE DISPLAY %{TTBR}[%{MSB:d};0] should be zero, but are not. Fields:

MSB unsigned int

x-1.

TTBR enum

which TTBR register.

ArchMsg.Warning.warning_unaligned_address_dbgdtrrx_write

TAGS UNPREDICTABLE DISPLAY Unaligned address %{ADDR} in X0/R0 when DBGDTRRX written in memory access mode. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unaligned_address_dbgdtrtx_read

TAGS **UNPREDICTABLE** DISPLAY Unaligned address %{ADDR} in XO/RO when DBGDTRTX read in memory access mode. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unaligned_to_device

TAGS **UNPREDICTABLE** DISPLAY Unaligned Access to Device Memory %{ADDR}. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unaligned_to_strongly_ordered

TAGS **UNPREDICTABLE** DISPLAY Unaligned Access to Strongly Ordered Memory %{ADDR}. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unknown_sau_rnr

TAGS **UNPREDICTABLE** DISPLAY SAU_RNR was set to an unsupported value.

ArchMsg.Warning.warning_unpredictable_AIRCR_PRIS_and_BFHFNMINs

TAGS **UNPREDICTABLE** DISPLAY The effect of setting both AIRCR.BFHFNMINs and AIRCR.PRIS to 1 is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_AIRCR_VECTCLRACTIVE_when_not_in_debug

TAGS **UNPREDICTABLE** DISPLAY The effect of writing a 1 to AIRCR.VECTCLRACTIVE if the processor is not halted in Debug state is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_AIRCR_VECTRESET_and_SYSRESETREQ

TAGS **UNPREDICTABLE** DISPLAY When the processor is halted in Debug state, if a write to the register writes a 1 to both VECTRESET and SYSRESETREQ, the behavior is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_AIRCR_VECTRESET_when_not_in_debug

TAGS **UNPREDICTABLE** DISPLAY The effect of writing a 1 to AIRCR.VECTRESET if the processor is not halted in Debug state is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_AIRCR_incorrect_VKEY

TAGS **UNPREDICTABLE** DISPLAY The value 0x05FA must be written to AIRCR.VKEY, otherwise the register write is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_EXC_RETURN_Reserved_Bit

EXC_RETURN[23:7] are reserved with the special condition that all bits should be written as one. Values other than all 1s are **UNPREDICTABLE**. TAGS **UNPREDICTABLE**. Fields:

EXC_RETURN unsigned int

Special PC value.

SBO_MASK unsigned int

Which bits Should Be One to not UNPRED.

ArchMsg.Warning.warning_unpredictable_PMEVTYPErn_ELO_VS_reserved_value

TAGS UNPREDICTABLE DISPLAY Unpredictable reserved value of PMEVTYPER<n>_ELO.VS.

ArchMsg.Warning.warning_unpredictable_change_to_DEMCR_MON_STEP_at_insufficient_priority

TAGS UNPREDICTABLE DISPLAY The effect of changing DEMCR.MON_STEP at an execution priority that is lower than the priority of the DebugMonitor exception is **CONSTRAINED UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_change_to_priority_of_active_exception

TAGS UNPREDICTABLE DISPLAY Changing the priority of an active exception is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_clear_lspact

TAGS UNPREDICTABLE DISPLAY **UNPREDICTABLE** state on context switch. Fields:

DRVEXC_TAKEN unsigned int

Derived exception taken, not pended.

REACHED_LAST_STORE unsigned int

Only the last faulted.

ArchMsg.Warning.warning_unpredictable_dbg_exit_unaligned_dlr

TAGS UNPREDICTABLE DISPLAY Debug exit to AArch32 with DLR[0] set to 1. PC[0] can be set to 0 or to DLR[0].

ArchMsg.Warning.warning_unpredictable_exception_catch

TAGS UNPREDICTABLE DISPLAY Generation of an exception catch event is unpredictable.

ArchMsg.Warning.warning_unpredictable_exception_return_inconsistent_state

TAGS UNPREDICTABLE DISPLAY state on exception return is unpredictable.

ArchMsg.Warning.warning_unpredictable_exception_return_instruction

TAGS UNPREDICTABLE DISPLAY exception return instruction is unpredictable.

ArchMsg.Warning.warning_unpredictable_in_debug_state

TAGS UNPREDICTABLE DISPLAY Instruction is **UNPREDICTABLE** when executed in debug state.

ArchMsg.Warning.warning_unpredictable_mair_encoding

TAGS UNPREDICTABLE DISPLAY Memory attribute %{ATTR} has invalid MAIR encoding; this may lead to unpredictable results. Fields:

ATTR unsigned int

Memory attribute.

ArchMsg.Warning.warning_unpredictable_pmu_counter_access

TAGS UNPREDICTABLE DISPLAY Access to PMU counters from non-secure ELO or EL1 is **UNPREDICTABLE** with MDCR_EL2.HPMN set to 0.

ArchMsg.Warning.warning_unpredictable_prioritization_breakpoint_match_vector_catch

TAGS UNPREDICTABLE DISPLAY Two events with the same priority occurred at the same instruction: Address Matching Vector Catch debug event, and Breakpoint debug event. It is constrained unpredictable which is taken.

ArchMsg.Warning.warning_unpredictable_stack_selection

TAGS **UNPREDICTABLE** DISPLAY **UNPREDICTABLE** state on context switch.

ArchMsg.Warning.warning_unpredictable_tsize_out_of_range

TAGS **UNPREDICTABLE** DISPLAY **%{TCR}.T%{TTBR:d}SZ (%{TSIZE:d} bits)** is out of range. Must be between 25 and **%{TSIZE_MAX:d}**. Fields:

TCR enum

Which TCR.

TSIZE signed int

Value of TSize.

TSIZE_MAX signed int

The maximum allowed value.

TTBR unsigned int

Which TTBR region.

ArchMsg.Warning.warning_unpredictable_unaligned_pc_as_base_register

TAGS **UNPREDICTABLE** DISPLAY The PC value must be word-aligned, otherwise the behavior of the instruction is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_unaligned_pop_stack

TAGS **UNPREDICTABLE** DISPLAY **CONSTRAINED UNPREDICTABLE** unaligned stack frame **%{FRAME_ADDR:x}** in exception-return. Fields:

FRAME_ADDR unsigned int

unaligned stack frame address.

ArchMsg.Warning.warning_unpredictable_unaligned_to_non_writeback

TAGS **UNPREDICTABLE** DISPLAY Unaligned Access to Non-Writeback-Cacheable Memory.

ArchMsg.Warning.warning_unpredictable_vmsa_memattrib

TAGS **UNPREDICTABLE** DISPLAY Unpredictable vmsa memory attribute with texcb value programmed to **%{TEXCB}** when tex remap is **%{TEX_REMAP}**. Fields:

TEXCB unsigned int

texcb value.

TEX_REMAP bool

use tex remap.

ArchMsg.Warning.warning_unpredictable_vtcr_t0sz_sl0

TAGS **UNPREDICTABLE** DISPLAY The combination of **VTCR.TOSZ=%{TOSZ:d}** and **VTCR.SL0=%{SL0:d}** is unpredictable. Fields:

SL0 signed int

VTCR.SL0, starting level for stage 2 translation table walks.

TOSZ signed int

VTCR.TOSZ, required input address range.

ArchMsg.Warning.warning_unpredictable_write_DHCSR

TAGS UNPREDICTABLE DISPLAY UNPREDICTABLE set of bit-changes in DHCSR. Fields:

NEW unsigned int

data attempted to be written.

OLD unsigned int

previous value.

ArchMsg.Warning.warning_unpredictable_write_SHCSR

TAGS UNPREDICTABLE DISPLAY UNPREDICTABLE setting a pending bit to 1 for an exception with priority >= execution priority.

ArchMsg.Warning.warning_unsupported_access_to_memory_mapped_register

TAGS UNPREDICTABLE DISPLAY Access to memory mapped register is unsupported.

ArchMsg.Warning.warning_user_jmcr_access

TAGS UNPREDICTABLE DISPLAY User %{WRITE:(write|read)} access to JMCR is unpredictable. Fields:

WRITE unsigned int

access is write.

ArchMsg.Warning.warning_wcr_mask_and_bas

TAGS UNPREDICTABLE DISPLAY DBGWCR_EL1.MASK is non-zero and BAS is not 0xff.

ArchMsg.Warning.warning_wcr_mask_reserved

TAGS UNPREDICTABLE DISPLAY DBGWCR_ELn.MASK is set to a reserved value.

ArchMsg.Warning.warning_wcr_non_configuous_bas

TAGS UNPREDICTABLE DISPLAY DBGWCR_EL1.BAS has non-contiguous set of ones.

ArchMsg.Warning.warning_wvr_masked_bit_not_zero

TAGS UNPREDICTABLE DISPLAY DBGWVRn_EL1.VA contains a bit masked by DBGWCRn_EL1.MASK that is not zero.

ArchMsg.warning_unpred_system_register_access

DISPLAY Unpredictable %{IS_WRITE:(read|write)} access to system register offset %{OFFSET}. Fields:

IS_WRITE unsigned int

Write Not Read.

OFFSET unsigned int

Register Offset.

BRANCH_MISPREDICT

Simulating branch mispredict. Fields:

PC unsigned int

Origin address (or 0 if unavailable).

BRA_DIR

Direct branches, to immediate address. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISSET enum

The instruction set of the branch instruction.

IS_COND bool

Indicates if this is a conditional branch.

IS_HINTED bool

Whether the conditional instruction is hinted.

IS_ISB bool

Whether the instruction is ISB.

IS_LINK bool

Indicates if this is a branch with link.

PC unsigned int

The address of the branch instruction.

TARGET_ISET enum

The instruction set after the branch.

TARGET_PC unsigned int

The address the instruction branches to.

BRA_INDIR

Indirect branches, perhaps to a register. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISSET enum

The instruction set of the branch instruction.

IS_COND bool

Indicates if this is a conditional branch.

IS_HINTED bool

Whether the conditional instruction is hinted.

IS_ISB bool

Whether the instruction is ISB.

IS_LINK bool

Indicates if this is a branch with link.

PC unsigned int

The address of the branch instruction.

TARGET_ISET enum

The instruction set after the branch.

TARGET_PC unsigned int

The address the instruction branches to.

BRA_INDIR_CCFAIL

Indirect branches fail, perhaps to a register.

BRA_RET_CCFAIL

Branch return fail, perhaps to a register.

CCFAIL

Conditional instruction condition check fail. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CCFAIL_UNC

Unconditional instruction condition check fail. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CCPASS

Conditional instruction condition check pass. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CCPASS_UNC

Unconditional instruction condition check pass. Fields:

COND **enum**

The condition of the conditional instruction.

CORE_NUM **unsigned int**

Core number in a multi processor.

PC **unsigned int**

The address of the conditional instruction.

CODE_CACHE_MAINT

Code cache maintenance. Fields:

END_INCL **unsigned int**

Inclusive end address for VA or PA requests.

NS **bool**

Non-secure world for PA.

START **unsigned int**

Start address for VA or PA requests.

TYPE **enum**

Request type.

COMPILE_BLOCK_END

Last instruction of basic block translated. Fields:

VADDR **unsigned int**

Address of next instruction after this basic block.

COMPILE_BLOCK_START

First instruction of basic block translated. Fields:

VADDR **unsigned int**

Address of first instruction in basic block.

COMPILE_INST

ARM instruction compiled. Fields:

DISASS **string**

Disassembly of the instruction.

ISSET **enum**

The instruction set of this instruction.

ITSTATE **unsigned int**

The ITSTATE current for the instruction.

OPCODE **unsigned int**

The opcode of the instruction.

PC unsigned int

The address of the instruction.

SIZE unsigned int

The size of the instruction in bytes.

CONTEXTIDR

Context ID Register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

NS bool

Secure or nonsecure banked register is accessed.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The new value written.

CONTEXT_SYNC

Called for every context synchronization event. Fields:

ARCH_CSE bool

Architectural Context Synchronization Event.

PC unsigned int

The address of the synchronization instruction.

SYSREG_SYNC bool

System register synchronization event.

CORE_ENDIAN

Core BE8 Big-Endian state changed. Fields:

BE8 bool

Core BE8 Big-Endian state.

CORE_INFO

Static processor attributes. Only triggered by a call to DumpState(). Fields:

ARCH_PROFILE enum

The architecture profile of the core.

CLUSTER_ID unsigned int

The cluster ID of this processor.

CORE_NUM unsigned int

The number of this core in an MP processor.

DETAILED_CORE_FEATURES enum

Detailed core features as a bitfield. Enum values are bitmasks to extract the info from the field.

FPU_VERSION enum

The VFP version implemented by the core.

MEM_ARCH enum

The memory architecture of the core.

MIDR_EL1 unsigned int

MIDR_EL1 value.

NUM_CORES unsigned int

The number of cores in this MP processor.

QUANTUM_SIZE unsigned int

The default quantum size of the core (in ticks).

SECURITY_FEATURES bool

Does the core have security features?.

CORE_LOADS

Processor load accesses. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ACQREL enum

Is this an acquire/release.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DATA unsigned int

The data read or written.

ELEMENT_SIZE unsigned int

Width of each element.

ENDIANNESS enum

Byte ordering of this access.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

LOCK enum

Normal, exclusive or locked access.

MECID unsigned int

Memory Encryption Context ID, if applicable.

MECID2 unsigned int

Memory Encryption Context ID of the second page, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PAGE_SIZE unsigned int

Page size as $\log_2(\text{size in bytes})$.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK, 2=Exclusive Failed, 3=Unsupported.

SIZE unsigned int

Width of the access in bytes. Only required if DATA is not traced.

TRANS bool

Is this a translated access.

VADDR unsigned int

The virtual address of the access.

CORE_REGS

Changes of the core registers R0 to R14. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number, 0 to 14.

OLD_VALUE unsigned int

The old value overwritten.

PHYS_ID enum

The physical register accessed.

VALUE unsigned int

The new value written to the register.

CORE_STALLED_TRANSITION

Detect when a core is prevented from progressing PChannel states. Fields:

CURRENT_STATE enum

Current power mode of core.

REASON string

The reason a transition was stalled, and what must be done to unblock it.

TARGET_STATE enum

Desired power mode of core after transition.

CORE_STORES

Processor store accesses. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ACQREL enum

Is this an acquire/release.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DATA unsigned int

The data read or written.

ELEMENT_SIZE unsigned int

Width of each element.

ENDIANNESS enum

Byte ordering of this access.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

LOCK enum

Normal, exclusive or locked access.

MECID unsigned int

Memory Encryption Context ID, if applicable.

MECID2 unsigned int

Memory Encryption Context ID of the second page, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PAGE_SIZE unsigned int

Page size as $\log_2(\text{size in bytes})$.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK, 2=Exclusive Failed, 3=Unsupported.

SIZE unsigned int

Width of the access in bytes. Only required if DATA is not traced.

TRANS bool

Is this a translated access.

VADDR unsigned int

The virtual address of the access.

CRYPTO_SPEC

Every crypto instruction speculatively executed.

DATA_CACHE_SET_ALLOC_TAGS

Set allocation tags from DC GVA instruction. Fields:

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

NSDESC unsigned int

The security state of the access.

PADDR unsigned int

The physical address of the access.

PAS enum

The physical address space of the page.

RESPONSE enum

0=Aborted, 1=OK.

SIZE unsigned int

The size of memory in bytes.

TAG unsigned int

The value of the tag being set.

TAGGING_TYPE enum

Type of Memory tagging (Physical or Virtual).

TAG_VADDR unsigned int

The virtual address of the Virtual Tag if Virtual tagging is used; otherwise 0.

VADDR unsigned int

Virtual address used for setting allocation tags.

DATA_CACHE_ZERO

Zero and invalidate cache line from DC ZVA instruction. Fields:

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

NSDESC unsigned int

The security state of the access.

PADDR unsigned int

The physical address of the access.

PADDR_TAG unsigned int

If tag checked, the physical address tag for the access.

PAS enum

The physical address space of the page.

RESPONSE enum

0=Aborted, 1=OK.

SIZE unsigned int

The size of zero'd memory in bytes.

TAG_CHECKED bool

Is this access tag checked.

VADDR unsigned int

Virtual address of the zero'd cache line.

DEBUG_EVENT

Hardware debug support event. Fields:

EVENT enum

Description of event.

VALUE unsigned int

data value.

DMI_ALLOCATE

DMI allocated. Fields:

ACCESS_DIRECTION enum

READ / WRITE.

PADDR unsigned int

The physical address of the DMI.

PTR unsigned int

DMI range pointer.

SIDE enum

DSIDE / ISIDE.

SIZE unsigned int

The size of DMI.

VADDR unsigned int

The virtual address of the DMI..

DMI_HIT

DMI hit, access used DMI (only fire for small DMI regions i.e. less than a full page). Fields:

ACCESS_DIRECTION enum

READ / WRITE.

PADDR unsigned int

The physical address of the DMI.

SIDE enum

DSIDE / ISIDE.

SIZE unsigned int

The size of the DMI.

VADDR unsigned int

The virtual address of the DMI..

DMI_REVOKE

DMI revoked. Fields:

ACCESS_DIRECTION enum

READ / WRITE.

PADDR unsigned int

The physical address of the DMI.

SIDE enum

DSIDE / ISIDE.

SIZE unsigned int

The size of DMI.

VADDR unsigned int

The virtual address of the DMI..

DWT_MATCH

DWT comparator matches. Fields:

NUM unsigned int

DWT comparator number.

TYPE enum

DWT comparator configuration.

END_COMPILE

Compilation end. Fields:

END_OF_PAGE_COUNT unsigned int

Number of basic blocks exited due to page boundary since START_COMPILE.

FETCHFAIL_COUNT unsigned int

Number of basic blocks exited due to ifetch failure START_COMPILE.

INST_COUNT unsigned int

Number of instructions compiled since START_COMPILE.

NONSEQ_COUNT unsigned int

Number of basic blocks exited due to non-sequential instructions since START_COMPILE.

PAGE_STRADDLE_COUNT unsigned int

Number of basic blocks exited due to unaligned instructions crossing page since START_COMPILE.

EXCEPTION

Exceptions that are taken. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ISSET enum

The instruction set of the processor when the exception occurred.

LR unsigned int

The value assigned to the link register.

PC unsigned int

The location where the exception occurred.

TARGET_ISSET enum

The instruction set of the exception handler code.

TARGET_PC unsigned int

The address the exception branches to.

VECTOR enum

The exception vector.

EXCEPTION_END

Every exception completed.

EXCEPTION_ENTRY

Event marking the entry of an exception. All pushes of registers to the stack will follow this event. Fields:

PC unsigned int

The location where the exception occurred.

VECTOR enum

The exception that occurred.

EXCEPTION_RAISE

Every exception raised.

EXCEPTION_RETURN

Marks the end of an exception, but on an M core does not indicate a program flow change (branch). In most situations it will be followed by an indirect branch. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISSET enum

The instruction set of the branch instruction.

PC unsigned int

The address of the branch instruction.

EXCEPTION_RETURN_PREBRANCH

Exception return event that must occur before PC and processor state have updated.

EXCEPTION_START

Every exception started.

EXCEPTION_VECTOR_FETCH

I-side vector fetch for M-class exception. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

DATA unsigned int

Result of fetch if OK.

RESPONSE enum

0=Aborted, 1=OK.

VADDR unsigned int

Where the vector is fetched from (including VTOR).

FP_STATE

Floating point state. Fields:

FPSR unsigned int

FPSR register value.

MASK unsigned int

The mask indicating updated cumulative bits.

FREQ_CHANGED

Core frequency changed. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The instruction count of this core.

NEW_FREQ signed int

The new frequency of this core (expressed in Hz).

OLD_FREQ signed int

The old frequency of this core (expressed in Hz).

HLT

HLT instruction occurred. Fields:

IMM16 unsigned int

The 16-bit immediate value encoded in HLT instruction.

INFO_EXCEPTION_REASON

Provide information of real cause of exception. Fields:

FaultCause enum

Which bit of which V7M/V8M-mainline FSR/HFSR would have been set.

PC unsigned int

The location where the exception occurred.

PHASE enum

What we are doing with the exception.

REASONS enum

Why we are doing it.

VECTOR enum

The exception that occurred.

INFO_STACKING

Provide information of real cause of exception. Fields:

PHASE enum

Are we starting or ending a sequence of stack writes.

REASONS enum

Why we are doing it.

INST

Every instruction executed. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DEBUG_STATE bool

The instruction is executed in debug state.

DISASS string

Disassembly of instruction.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISET enum

The current instruction set.

ITSTATE unsigned int

The current ITSTATE.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

MODE enum

The mode the core is in.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

OPCODE unsigned int

The opcode of the instruction.

PADDR unsigned int

The physical address of the instruction.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the instruction.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the page.

PC unsigned int

The address of the instruction.

SECURITY_STATE enum

The core's security state.

SIZE unsigned int

The size of the instruction in bytes.

INST_END

Every instruction completed.

INST_START

Every instruction started. Fields:

DEBUG_STATE bool

The instruction is executed in debug state.

ISET enum

The current instruction set.

MODE enum

The mode the core is in.

NS enum

The current Secure State.

PC unsigned int

The address of the conditional instruction.

SECURITY_STATE enum

The current Security State.

INST_STRADDLE

Instruction straddles boundary. Fields:

DEBUG_STATE bool

The instruction is executed in debug state.

ISET enum

The current instruction set.

MODE enum

The mode the core is in.

NS enum

The current Secure State.

PC unsigned int

The address of the conditional instruction.

SECURITY_STATE enum

The current Security State.

ISB

ISB instruction occurred. Fields:

PC unsigned int

The address of the ISB instruction.

LOAD_MULTIPLE_REGS_LIST

List of destination registers (not source) of a multiple registers load instruction. Fields:

REG string

Register to add into the list.

LOCAL_MONITOR

Local monitor activity. Fields:

PADDR unsigned int

Local Monitor Address.

State enum

State of the monitor (Open/Exclusive).

LOCKUP_CYCLE

This event is triggered if the core enters or stays in the lock-up state. Fields:

CAUSE enum

What causes the lockup?.

LOCKUP_ENTRY

This event is triggered if the core enters the lock-up state. Fields:

PC unsigned int

The PC when the lockup state is entered.

LOCKUP_EXIT

This event is triggered if the core leaves the lock-up state.

MODE_CHANGE

Mode change. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

MODE enum

The new mode.

OLD_MODE enum

The old mode.

MPU_TRANS

Address translation information. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

EXEC_PERM enum

Execution Permission.

INNERCACHE_RA bool

Is the inner cache allocate on read.

INNERCACHE_TYPE enum

Inner Caching scheme (NC/MB/WA).

INNERCACHE_WA bool

Is the inner cache allocate on write.

MEMTYPE enum

Memory type.

OUTERCACHE_RA bool

Is the outer cache allocate on read.

OUTERCACHE_TYPE enum

Outer Caching scheme (NC/MB/WA).

OUTERCACHE_WA bool

Is the outer cache allocate on write.

PADDR unsigned int

Address of the access.

PAGESIZE unsigned int

Page size as $\log_2(\text{size})$.

READ_PERM enum

Read Permission.

REG_NUM unsigned int

MPU register based region-number 0-15 (else 0xFF=none, 0xBB/0xDD=background/default).

SH enum

Shareability.

SIDE enum

Inst / Data.

WRITE_PERM enum

Write Permission.

PERIODIC

Called for every quantum. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The instruction count of this CPU.

PC unsigned int

The address of the next instruction to be executed on this CPU.

POTINDEX_UPDATE

Value of POT index for current pc changed. Fields:

NEW_POTINDEX unsigned int

New POTIndex.

OLD_POTINDEX unsigned int

Old POTIndex.

PPB_READ_ACCESS

Trace reads from the core to memory mapped registers in the PPB (private peripheral bus) address range. Fields:

ACCESS_SIZE unsigned int

Logarithm of the access size: 0=byte, 1=halfword, 2=word, 3=doubleword, ...

BURST_LEN unsigned int

The number of data transfers (beats) in this burst.

DATA unsigned int

The data transferred.

INNER_ATTRIBUTES unsigned int

The inner attributes: bit0=write allocate, bit1=read allocate, bit2=cachable, bit3=bufferable, bit4=shareable.

INST bool

Is an instruction fetch.

LOCK enum

Normal / Locked / Exclusive.

MANAGER_ID unsigned int

The AXI manager ID.

NS bool

Is a nonsecure access.

OUTER_ATTRIBUTES unsigned int

The outer attributes: bit0=write allocate, bit1=read allocate, bit2=cachable, bit3=bufferable, bit4=shareable.

PADDR unsigned int

Physical address of access.

PRIV bool

Is privileged access.

RESPONSE enum

Whether the transaction was successful, or an error occurred.

USER_FLAGS unsigned int

Core specific additional signals.

PPB_WRITE_ACCESS

Trace writes from the core to memory mapped registers in the PPB (private peripheral bus) address range. Fields:

ACCESS_SIZE unsigned int

Logarithm of the access size: 0=byte, 1=halfword, 2=word, 3=doubleword, ...

BURST_LEN unsigned int

The number of data transfers (beats) in this burst.

DATA unsigned int

The data transferred.

INNER_ATTRIBUTES unsigned int

The inner attributes: bit0=write allocate, bit1=read allocate, bit2=cachable, bit3=bufferable, bit4=shareable.

INST bool

Is an instruction fetch.

LOCK enum

Normal / Locked / Exclusive.

MANAGER_ID unsigned int

The AXI manager ID.

NS bool

Is a nonsecure access.

OUTER_ATTRIBUTES unsigned int

The outer attributes: bit0=write allocate, bit1=read allocate, bit2=cachable, bit3=bufferable, bit4=shareable.

PADDR unsigned int

Physical address of access.

PRIV bool

Is privileged access.

RESPONSE enum

Whether the transaction was successful, or an error occurred.

USER_FLAGS unsigned int

Core specific additional signals.

PRELOAD_DATA

Data preload from PLD instruction. Fields:

VADDR unsigned int

Virtual address of the data that should be preloaded.

PRELOAD_INST

Instruction preload from PLI instruction. Fields:

VADDR unsigned int

Virtual address of the instruction that should be preloaded.

PRE_CORE_LOAD

Trace just before a core load. Fields:

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

PRE_CORE_STORE

Trace just before a core store. Fields:

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

PSTATE_PACTIVE_OPMODE_TRANSITION

Trace pactive and Opmode mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Operating mode after the transition.

OLD_STATE enum

Operating mode before the transition.

PSTATE_PACTIVE_OPMODE_TRANSITION_DENIED

Trace the denied pactive and Opmode mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Operating mode after the transition.

OLD_STATE enum

Operating mode before the transition.

REASON string

Reason for the transition being denied.

PSTATE_PACTIVE_POWER_TRANSITION

Trace pactive and power mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Power mode after the transition.

OLD_STATE enum

Power mode before the transition.

PSTATE_PACTIVE_POWER_TRANSITION_DENIED

Trace the denied pactive and power mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Power mode after the transition.

OLD_STATE enum

Power mode before the transition.

REASON string

Reason for the transition being denied.

PSTATE_PACTIVE_THREAD_OPMODE_TRANSITION

Trace pactive and Opmode mode transitions for threads on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Operating mode after the transition.

OLD_STATE enum

Operating mode before the transition.

QUANTUM_START

Called at the start of every quantum.

RUN_STATE

Run state transition. Fields:

INST_COUNT unsigned int

Ticks count at point of transition.

NEW enum

New run state.

OLD enum

Old run state.

SEMIHOSTING_CALL

Call of a semihost function occurred. Fields:

PC unsigned int

The program counter after the semihosting call.

SEMIHOSTING_PRECALL

About to call semihost. Fields:

EL2 bool

Use EL2 translation regime.

NS bool

Is Non-Secure.

REG_WIDTH unsigned int

The current register width in bytes.

SIGNAL

External signal state change. Fields:

SIGNAL enum

Signal that changed.

STATE bool

Signal asserted state.

START_COMPILE

Compilation started. Fields:

VADDR unsigned int

Instruction where compilation begins.

STG_MULTIPLE_MICRO_OPS_ADDRESS_RANGE

Address range for MTE stores with multiple uops. Fields:

ADDR_END unsigned int

Final address.

ADDR_START unsigned int

Starting address.

STORE_MULTIPLE_REGS_ADDRESS_RANGE

Address range for stores with multiple registers. Fields:

ADDR_END unsigned int

Final address.

ADDR_START unsigned int

Starting address.

SUBOPTIMAL_LDST_RETIRED

suboptimal load/store. Fields:

ACCESS_SIZE unsigned int

Log2 of the access width in bytes used for alignment checking (1: half-word, 2: word ...).

OPTIMAL_ALIGN_SELECTOR unsigned int

Selector to decide the optimal alignment size (1: access width, 2: 4-bytes, 3: 8-bytes, ..., 12: 4-kbytes).

VADDR unsigned int

The virtual address of the access.

SYNC

Called for every synchronization, that is every quantum break. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The instruction count of this CPU.

LOCAL_QUANTUM unsigned int

The local quantum of this CPU.

LOCAL_TIME unsigned int

The local time of this CPU.

SYNC_REASON enum

Reason for sync.

SYSCALL

System call instruction executed. Fields:

IMM unsigned int

Immediate value of the system call instruction.

TYPE enum

System call type.

VADDR unsigned int

Instruction that caused the system call.

SYSREG_READ

Trace reads from system registers. Fields:

REG_NAME enum

Register number.

VALUE unsigned int

The value read.

SYSREG_UPDATE128

Triggers when the system updates a register. Fields:

REG enum

Register number.

UNKNOWN unsigned int

Bits of the register which became unknown.

VALUE unsigned int

Value written to the register.

SYSREG_UPDATE32

Triggers when the system updates a register. Fields:

REG enum

Register number.

UNKNOWN unsigned int

Bits of the register which became unknown.

VALUE unsigned int

Value written to the register.

SYSREG_UPDATE64

Triggers when the system updates a register. Fields:

REG enum

Register number.

UNKNOWN unsigned int

Bits of the register which became unknown.

VALUE unsigned int

Value written to the register.

SYSREG_WRITE

Trace write to system registers. Fields:

REG_ACCESS enum

Register access status.

REG_NAME enum

Register number.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

Value written to the register.

UNALIGNED_LDST_RETIRED

Processor unaligned load/store. Fields:

ACCESS_SIZE unsigned int

Log2 of the access width in bytes used for alignment checking (1=>half-word, 2->word ...).

NUMBER_OF_BEATS unsigned int

Number of accesses (beats) in this burst (Total size in bytes is ACCESS_SIZE * NUMBER_OF_BEATS).

VADDR unsigned int

The virtual address of the access.

WAYPOINT

Signals end of basic block execution. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ISET enum

Origin instruction set.

IS_COND bool

Indicates if this is a conditional waypoint.

PC unsigned int

Origin address (or 0 if unavailable).

TAKEN bool

Indicates if this waypoint was taken.

TARGET unsigned int

Destination address.

TARGET_ISET enum

Destination instruction set.

WFE_END

WFE ended. Fields:

INST_COUNT unsigned int

Ticks count when leaving WFE.

WFE_EVENT_REGISTER

WFE event register status: set/clear, reason. Fields:

INST_COUNT unsigned int

Ticks count.

REASON enum

Reason for set/clear. Only REASON==0 (Cleared by WFE) clears the bit, all other reasons set it.

WFE_IGNORED

WFE ignored. Fields:

EVENT bool

This WFE was ignored because the event register was set.

INST_COUNT unsigned int

Ticks count when ignoring WFE.

TRAPPED bool

This WFE was trapped.

WFE_START

WFE entered. Fields:

INST_COUNT unsigned int

Ticks count when entering WFE.

WFI_END

WFI ended. Fields:

INST_COUNT unsigned int

Ticks count when leaving WFI.

WFI_IGNORED

WFI ignored. Fields:

DISABLED bool

This WFI was ignored because WFI is disabled.

INST_COUNT unsigned int

Ticks count when ignoring WFI.

REASON enum

specifies reason why WFI trace was ignored.

TRAPPED bool

This WFI was trapped.

WFI_START

WFI entered. Fields:

INST_COUNT unsigned int

Ticks count when entering WFI.

WFI_WAKEUP

WFI wakeup. Fields:

INST_COUNT unsigned int

Ticks count when WFI wakeup occurred.

REASON enum

Reason for wakeup.

XPSR

Changes to the xPSR register. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

OLD_VALUE unsigned int

The old xPSR value.

UNKNOWN unsigned int

Bits within the register that have unknown value.

VALUE unsigned int

The new xPSR value.

2.40 ARM_Cortex-M23

This section describes the trace sources.

ASYNC_MEMORY_FAULT

Context ID Register write. Fields:

FAULT unsigned int

Fault status present in ESR.ISS encoding for Serror in ARCH64 and present in DFSR encoding for external abort format in ARCH32.

PADDR unsigned int

Physical Address (or 0 if unavailable).

ATOMIC_END_ACCESS

Bus trace access for atomics. Fields:

ACCESS_FAIL bool

Memory access failed.

ADDR unsigned int

The virtual address of the access.

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

COMPARE_VALUE unsigned int

Compare value for CAS.

LOAD_VALUE unsigned int

Loaded values.

NSDESC unsigned int

The security state of the access.

OPERAND_VALUE unsigned int

Operation's operand.

OPERATION enum

Operation type.

PADDR unsigned int

The physical address of the access.

PRIV bool

Is this a privileged access?.

ATOMIC_START_ACCESS

Bus trace access for atomics. Fields:

ADDR unsigned int

The virtual address of the access.

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

COMPARE_VALUE unsigned int

Compare value for CAS.

NSDESC unsigned int

The security state of the access.

OPERAND_VALUE unsigned int

Operation's operand.

OPERATION enum

Operation type.

PADDR unsigned int

The physical address of the access.

PRIV bool

Is this a privileged access?.

ArchMsg.Error.exit_code

DISPLAY exit code: %{EXITCODE} component kind: %{KIND} reason: %{REASON}. Fields:

EXITCODE signed int

Exit code to be returned.

KIND string

Component kind that invoked the exit code trace.

REASON string

Reason why the exit code trace was invoked (optional).

ArchMsg.Warning.Unpredictably Indexed PM Event Register

TAGS **UNPREDICTABLE** DISPLAY Event %{IsDirect:(Selector value|Register)} %{SEL} is >= %{N}. Fields:

IsDirect unsigned int

Direct Access rather than indirect.

N unsigned int

Number of Accessible Registers.

SEL unsigned int

Selector.

ArchMsg.Warning.dap_csw_bad_size

A write to CM3DAP CSW has an invalid size field. DISPLAY Write %{DATA} to CM3DAP CSW has invalid size field. Fields:

DATA unsigned int

bits[2:0] are size.

ArchMsg.Warning.dcimvac_matches_watchpoint

TAGS UNPREDICTABLE DISPLAY Watchpoints matching an AArch32 DCIMVA are implementation defined.

ArchMsg.Warning.decode_fieldmismatch

TAGS UNPREDICTABLE DISPLAY Instruction is unpredictable with different fields %{FIELD1} and %{FIELD2}. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_frachitsnegative

TAGS UNPREDICTABLE DISPLAY VCVT Instruction is unpredictable with a negative number of fraction bits.

ArchMsg.Warning.decode_initblock

TAGS UNPREDICTABLE DISPLAY Instruction is unpredictable in an IT block.

ArchMsg.Warning.decode_initblocknotlast

TAGS UNPREDICTABLE DISPLAY Instruction is unpredictable in an IT block when not the last.

ArchMsg.Warning.decode_invalid_condition

TAGS UNPREDICTABLE DISPLAY Instruction is unpredictable when the condition is not AL.

ArchMsg.Warning.decode_invalidfieldcombination

TAGS UNPREDICTABLE DISPLAY Instruction is unpredictable with the values in %{FIELD1} and %{FIELD2}. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_invalidvalue

TAGS UNPREDICTABLE DISPLAY Instruction is unpredictable with the value in %{FIELD}. Fields:

FIELD enum

Field name.

ArchMsg.Warning.decode_mem_hint_unallocated

TAGS UNPREDICTABLE DISPLAY There is no memory hint allocated to this bit pattern.

ArchMsg.Warning.decode_registermatch

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with %{FIELD1} and %{FIELD2} as the same register. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_registermismatch

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with %{FIELD1} and %{FIELD2} as different registers. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_registeroutofrange

TAGS **UNPREDICTABLE** DISPLAY Floating-point instruction is unpredictable indexing beyond the end of the register bank.

ArchMsg.Warning.decode_sbzsbo

TAGS **UNPREDICTABLE** DISPLAY Reserved bits in the instruction are not canonical.

ArchMsg.Warning.decode_transactiontoobig

TAGS **UNPREDICTABLE** DISPLAY Attempt to perform a memory transaction of more than 128 bytes.

ArchMsg.Warning.decode_unpred_other

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable.

ArchMsg.Warning.decode_unpreduseofpc

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable when %{FIELD} is PC. Fields:

FIELD enum

Field name.

ArchMsg.Warning.decode_unpreduseofr13

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable when %{FIELD} is R13. Fields:

FIELD enum

Field name.

ArchMsg.Warning.decode_unpreduseofr13inreglist

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with R13 in register-list.

ArchMsg.Warning.decode_useofd16orhigher

TAGS **UNPREDICTABLE** DISPLAY Attempt to use register D16 or higher where it is not allowed.

ArchMsg.Warning.decode_writebackandbaseinlist

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with write back when the base register is in the transfer list.

ArchMsg.Warning.decode_zeroregistersinlist

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable when no registers are to be transferred.

ArchMsg.Warning.recursive_branch

An instruction is performing a branch that targets the same instruction. If this was intended then a WFI might be more effective DISPLAY recursive branch detected at PC=%{PC}. PRIMARY KEY PC. Fields:

PC unsigned int

Address of instruction causing the branch.

ArchMsg.Warning.recursive_exception

An instruction has generated an exception that targets the same instruction. DISPLAY recursive %{TYPE} exception detected at PC=%{PC} (CPSR=%{CPSR}). PRIMARY KEY PC. Fields:

CPSR unsigned int

Processor state.

PC unsigned int

Address of instruction causing the exception.

TYPE enum

Type of exception.

ArchMsg.Warning.reentrant_vector_catch

TAGS **UNPREDICTABLE** DISPLAY Vector catch debug event on Prefetch or Data abort vector %{ADDR}. Fields:

ADDR unsigned int

vector address.

ArchMsg.Warning.reserved_it_state

TAGS **UNPREDICTABLE** DISPLAY Execution with reserved IT state %{ITSTATE}. Fields:

ITSTATE unsigned int

reserved IT state.

ArchMsg.Warning.secure_vector_fetch_from_nonsecure

DISPLAY Secure vector table is being fetched from Non-secure memory.

ArchMsg.Warning.sg_in_it_block

Arm recommends that software does not place SG (security gate instruction) inside an IT instruction block.

ArchMsg.Warning.unpred_FAR_on_LRCPC3_non_atomic_mem_access

TAGS **UNPREDICTABLE** Unpredictable FAR/FSTATUS on LRCPC3 pair instructions resulting in two separate memory accesses
 DISPLAY Unpredictable FAR on LRCPC3 non atomic memory access at %{ADDR1} with FSTATUS %{FSTATUS1} or %{ADDR2} with FSTATUS %{FSTATUS2}. Fields:

ADDR1 unsigned int

Possible FAR address 1.

ADDR2 unsigned int

Possible FAR address 2.

FSTATUS1 unsigned int

Possible fault status value 1.

FSTATUS2 unsigned int

Possible fault status value 2.

ArchMsg.Warning.unpred_async_tag_check_fail_on_sve_speculative_load_do_update_ffr

Unpredictable set to FALSE of the FFR element associated to SVE speculative load with an async tag check fail TAGS **UNPREDICTABLE**.

ArchMsg.Warning.unpredictable_a32_breakpoint

TAGS **UNPREDICTABLE** DISPLAY Unpredictable breakpoint on a A32 instruction at %{ADDR1} Offset %{OFFSET} BAS %{BAS} for a %{IS_ADDRESS_MISMATCH_BREAKPOINT:(Mismatch|Match)} Breakpoint number %{N:d} Breakpoint address %{ADDR2}. Fields:

ADDR1 unsigned int

Address.

ADDR2 unsigned int

Breakpoint address.

BAS unsigned int

BAS field.

IS_ADDRESS_MISMATCH_BREAKPOINT bool

Is a Mismatch breakpoint, not a Match breakpoint.

N unsigned int

Breakpoint number.

OFFSET unsigned int

Always has a value of zero.

ArchMsg.Warning.unpredictable_banked_register_access

TAGS **UNPREDICTABLE** DISPLAY Access to banked register (R=%{R:d} SYSm=%{SYSM:2x}) is unpredictable. Fields:

R unsigned int

R bit.

SYSM unsigned int
SYSm.

ArchMsg.Warning.unpredictable_ccfail_undef

TAGS **UNPREDICTABLE** DISPLAY An instruction %{OPCODE} at VA=%{VADDR} failed its condition codes check but would otherwise have caused a non-data-dependent trap or undefined exception. Behaviour is implementation-defined choice of exception or nop. Fields:

OPCODE unsigned int
opcode of the instruction.

VADDR unsigned int
Virtual address of the instruction being executed.

ArchMsg.Warning.unpredictable_ici_bits

TAGS **UNPREDICTABLE** DISPLAY ICI bits do not apply or are architecturally unpredictable for instruction.

ArchMsg.Warning.unpredictable_t32_breakpoint

TAGS **UNPREDICTABLE** DISPLAY Unpredictable breakpoint on a T32 instruction at %{ADDR1} Offset %{OFFSET} BAS %{BAS} for a %{IS_ADDRESS_MISMATCH_BREAKPOINT:(Mismatch|Match)} Breakpoint number %{N:d} Breakpoint address %{ADDR2}. Fields:

ADDR1 unsigned int
Address.

ADDR2 unsigned int
Breakpoint address.

BAS unsigned int
BAS field.

IS_ADDRESS_MISMATCH_BREAKPOINT bool
Is a Mismatch breakpoint, not a Match breakpoint.

N unsigned int
Breakpoint number.

OFFSET unsigned int
Indicates which halfword of a 32-bit Thumb instruction the breakpoint is set on. 0 for the first halfword or if the instruction is 16 bits wide, or 1 for the second halfword.

ArchMsg.Warning.unpredictable_t32_vector_catch

TAGS **UNPREDICTABLE** DISPLAY Unpredictable Vector catch exception on a T32 instruction at %{ADDR} Offset %{OFFSET}. Fields:

ADDR unsigned int
Address.

OFFSET unsigned int

Indicates which halfword of a 32-bit Thumb instruction the vector catch exception is set on. 0 for the first halfword or if the instruction is 16 bits wide, or 1 for the second halfword.

ArchMsg.Warning.unpredictable_watchpoint_far

TAGS **UNPREDICTABLE** DISPLAY The value for FAR/EDWAR for hitting this watchpoint may be anywhere in the range %{LOWER_BOUND} to %{UPPER_BOUND}. Fields:

LOWER_BOUND unsigned int

the lowest address accessed by the instruction that triggered the watchpoint.

UPPER_BOUND unsigned int

the highest watchpointed address accessed by that instruction.

ArchMsg.Warning.warning_AdvSIMDExpandImmUnexpectedZero

TAGS **UNPREDICTABLE** DISPLAY AdvSIMDExpandImm may treat this immediate value as **UNPREDICTABLE**.

ArchMsg.Warning.warning_ConditionalSMC

TAGS **UNPREDICTABLE** DISPLAY SMC instruction has **UNPREDICTABLE** effects when conditional and when combined with traps.

ArchMsg.Warning.warning_access_crosses_page_boundary

TAGS **UNPREDICTABLE** DISPLAY Access crosses page boundary.

ArchMsg.Warning.warning_access_crosses_page_boundary_has_fault

TAGS **UNPREDICTABLE** DISPLAY Access that crosses page boundary has fault(s).

ArchMsg.Warning.warning_access_crosses_page_boundary_has_fault_on_both_pages

TAGS **UNPREDICTABLE** DISPLAY Access that crosses page boundary has a fault on each page. Fields:

ADDR1 unsigned int

FAR address on first page.

ADDR2 unsigned int

FAR address on second page.

FSTATUS1 unsigned int

Fault status on first page.

FSTATUS2 unsigned int

Fault status on second page.

ArchMsg.Warning.warning_access_crosses_page_boundary_spanning_different_memory

TAGS **UNPREDICTABLE** DISPLAY Access at address %{ADDR} crosses page from MEM_TYPE %{MEMTYPE_PAGE1} to %{MEMTYPE_PAGE2}. Fields:

ADDR unsigned int

address of access crossing page.

MEMTYPE_PAGE1 signed int

First Page's memory type (0-Normal, 1-Non-Normal).

MEMTYPE_PAGE2 signed int

Second Page's memory type (0-Normal, 1-Non-Normal).

ArchMsg.Warning.warning_access_wraps_around_memory

TAGS UNPREDICTABLE DISPLAY Access wraps around memory in %{WIDTH:d}bit mode.
Fields:

WIDTH unsigned int

Address width.

ArchMsg.Warning.warning_bcr_linking_status

TAGS UNPREDICTABLE DISPLAY DBGBCR.{BT2,BT} combination gives unpredictable behaviour.

ArchMsg.Warning.warning_bcr_mask_reserved

TAGS UNPREDICTABLE DISPLAY DBGBCR_ELn.MASK is set to a reserved value.

ArchMsg.Warning.warning_bvr_masked_bit_not_zero

TAGS UNPREDICTABLE DISPLAY DBGBVRn_EL1.VA contains a bit masked by DBGBCRn_EL1.MASK that is not zero.

ArchMsg.Warning.warning_ccsidr_unimplemented_level

TAGS UNPREDICTABLE DISPLAY CCSIDR read while CSSELR %{CSSELR:x} points to an unimplemented cache level. Fields:

CSSELR unsigned int

current effective value of CSSELR.

ArchMsg.Warning.warning_change_to_ns_when_tge_set

TAGS UNPREDICTABLE DISPLAY Attempting to change NS to 1 when HCR.TGE = 1.

ArchMsg.Warning.warning_contiguous_bit_check_abort

TAGS UNPREDICTABLE DISPLAY An abort occurred whilst attempting to check TLB entry is contiguous at %{ENTRY_ADDR}. Fields:

ENTRY_ADDR unsigned int

address of conflicting TLB entry.

ArchMsg.Warning.warning_contiguous_bit_error

TAGS UNPREDICTABLE DISPLAY TLB entry at %{ENTRY_ADDR} was not contiguous with entry at %{CONTIG_ADDR} Contents are %{ENTRY_DATA} but expected %{CONTIG_DATA}.
Fields:

CONTIG_ADDR unsigned int

address of the TLB first read, with which this entry is expected to be contiguous.

CONTIG_DATA unsigned int

expected contents based on the entry at CONTIG_ADDR.

ENTRY_ADDR unsigned int

address of conflicting TLB entry.

ENTRY_DATA unsigned int

contents of conflicting TLB entry.

ArchMsg.Warning.warning_contiguous_bit_error_D128

TAGS **UNPREDICTABLE** DISPLAY TLB entry at %{ENTRY_ADDR} was not contiguous with entry at %{CONTIG_ADDR} Contents are %{ENTRY_DATA_HI} %{ENTRY_DATA_LO} but expected %{CONTIG_DATA}. Fields:

CONTIG_ADDR unsigned int

address of the TLB first read, with which this entry is expected to be contiguous.

CONTIG_DATA unsigned int

expected contents based on the entry at CONTIG_ADDR.

ENTRY_ADDR unsigned int

address of conflicting TLB entry.

ENTRY_DATA_HI unsigned int

Higher bit contents of conflicting TLB entry.

ENTRY_DATA_LO unsigned int

Lower bit contents of conflicting TLB entry.

ArchMsg.Warning.warning_cp10_cp11_mismatch

TAGS **UNPREDICTABLE** DISPLAY Unpredictable configuration of CP10 and CP11 controls at EL %{EL:d}. Fields:

EL unsigned int

EL owning the control.

ArchMsg.Warning.warning_cp10_cp11_reserved_value

TAGS **UNPREDICTABLE** DISPLAY Unpredictable configuration of CPACR CP10 (%{CP10}) or CP11 (%{CP11}). Fields:

CP10 unsigned int

CP10 access.

CP11 unsigned int

CP11 access.

ArchMsg.Warning.warning_csselr_level_out_of_range

TAGS **UNPREDICTABLE** DISPLAY CSSELR.Level out of range, written value %{LEVEL} with only %{IMPLEMENTED:d} levels implemented. Fields:

IMPLEMENTED unsigned int

number of cache levels implemented.

LEVEL unsigned int

written value of CSSELR.Level.

TYPE enum

The type of cache selected by CSSELR.InD and TnD.

ArchMsg.Warning.warning_debug_flow_control_bits_not_obeyed

TAGS **UNPREDICTABLE** DISPLAY Flow control bits for %{REGISTER} were not obeyed, causing %{ERROR}. Fields:

ERROR enum

type of error resulting.

REGISTER enum

register under consideration.

ArchMsg.Warning.warning_debug_register_access_during_reset

TAGS **UNPREDICTABLE** DISPLAY Unpredictable %{IS_WRITE:(read|write)} access to debug register offset %{OFFSET} during reset. Fields:

IS_WRITE unsigned int

Write Not Read.

OFFSET unsigned int

Register Offset.

ArchMsg.Warning.warning_decode_cps_inconsistent_fields

TAGS **UNPREDICTABLE** DISPLAY The mode setting fields in the CPS instruction are inconsistent.

ArchMsg.Warning.warning_decode_invalid_state

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable in current exception-level/security state.

ArchMsg.Warning.warning_default_cacheable_mmu_on

TAGS **UNPREDICTABLE** DISPLAY Default cacheable is enabled (HCR.DC = 1) while MMU is enabled.

ArchMsg.Warning.warning_default_cacheable_vmmu_off

TAGS **UNPREDICTABLE** DISPLAY Default cacheable is enabled (HCR.DC = 1) while HCR.VM = 0.

ArchMsg.Warning.warning_deprecated_wvr_bit_2_set

TAGS **UNPREDICTABLE** DISPLAY DBGWVR%{N:d}_EL1=%{WVR:h} has bit 2 set, deprecated in ARMv8. Fields:

N unsigned int

index.

WVR unsigned int

DBGWVR.

ArchMsg.Warning.warning_eret_while_software_step_active_pending

Missing ISB between setting MDSCR_EL1.SS and ERET TAGS **UNPREDICTABLE**.

ArchMsg.Warning.warning_exclusive_to_non_normal

TAGS **UNPREDICTABLE** DISPLAY Exclusive Access to Non-Normal Memory %{ADDR}. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_exclusive_to_non_writeback

TAGS **UNPREDICTABLE** DISPLAY Exclusive Access to Non-Writeback-Cacheable Memory %{ADDR}. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_execute_from_device_memory

TAGS **UNPREDICTABLE** DISPLAY An attempt was made to execute from Device memory.

ArchMsg.Warning.warning_illegal_cpsr_mode

TAGS **UNPREDICTABLE** DISPLAY Writing an illegal or unimplement mode (%{NEW_MODE:h}) to CPSR was unpredictable. Fields:

NEW_MODE unsigned int

New value of CPSR.M.

ArchMsg.Warning.warning_illegal_srs_mode

TAGS **UNPREDICTABLE** DISPLAY Illegal or **UNPREDICTABLE** mode (%{MODE:h}) used for SRS instruction. Fields:

MODE unsigned int

mode for Banked SP.

ArchMsg.Warning.warning_implementation_defined_read_debug_register_in_SCS

Read Access to Debug registers from the processor is **IMPLEMENTATION DEFINED** TAGS **IMP_DEF**. Fields:

PPB_OFFSET unsigned int

debug register offset.

ArchMsg.Warning.warning_implementation_defined_sequential_security_transitions_supported

The behavior of sequential instruction fetches that cross from non-secure to secure memory and contain an SG instruction is **CONSTRAINED UNPREDICTABLE** TAGS **UNPREDICTABLE**.

ArchMsg.Warning.warning_implementation_defined_stack_limit_check_supported

It is **IMPLEMENTATION DEFINED** whether stack pointer limit checking is performed for this instruction TAGS **IMP_DEF**. Fields:

ADDRESS unsigned int

Address of instruction.

OPCODE unsigned int

instruction opcode.

ArchMsg.Warning.warning_implementation_defined_taggednonsharedmemorytype

TAGS IMP_DEF DISPLAY Memory attribute has Tagged, non shared memory type; it is **IMPLEMENTATION DEFINED** whether the memory location is treated as Tagged or non-Tagged.

ArchMsg.Warning.warning_implementation_defined_write_debug_register_in_SCS

Write Access to Debug registers from the processor is **IMPLEMENTATION DEFINED** TAGS IMP_DEF. Fields:

DATA unsigned int

data attempted to be written.

PPB_OFFSET unsigned int

debug register offset.

ArchMsg.Warning.warning_invalid_tcr_granule

TAGS **UNPREDICTABLE** DISPLAY TCR.TG%{TG_ID:(0|1)} bits have been set to a granule size %{\$REQUEST:d}K not implemented - using %{\$SUBSTITUTE:d}K. Fields:

REQUEST signed int

page size requested (or 0 for reserved).

SUBSTITUTE signed int

best guess available page size.

TG_ID bool

bits TG0 or TG1.

ArchMsg.Warning.warning_load_multiple_user_registers_from_user_mode

TAGS **UNPREDICTABLE** DISPLAY An LDM(user registers) instruction executed from user mode.

ArchMsg.Warning.warning_mte_in_mem_attr_not_supported

TAGS **UNPREDICTABLE** DISPLAY Memory attribute %{\$ATTR} has Tagged memory type but Armv8.5-MTE is not implemented. Fields:

ATTR unsigned int

Memory attribute.

ArchMsg.Warning.warning_reserved_breakpoint_state_match

TAGS **UNPREDICTABLE** DISPLAY %{\$IS_BREAKPOINT:(Watchpoint|Breakpoint)} programmed with a reserved combination of HMC, SSC and %{\$IS_BREAKPOINT:(PAC|PMC)}. Fields:

IS_BREAKPOINT unsigned int

Is a breakpoint, not a watchpoint.

ArchMsg.Warning.warning_shareability

TAGS **UNPREDICTABLE** DISPLAY Unpredictable: combination of the 1st and 2nd stages of translation is Normal Inner Non-Cacheable, Outer Non-Cacheable.

ArchMsg.Warning.warning_software_step_set_while_enabled

TAGS **UNPREDICTABLE** DISPLAY MDSCR_EL1.SS set to 1 while software step debug exceptions are enabled.

ArchMsg.Warning.warning_thumb_instruction_wraps_around_memory

TAGS UNPREDICTABLE DISPLAY Thumb instruction wraps around memory.

ArchMsg.Warning.warning_ttbr_sbz_bits_are_not_zero

TAGS UNPREDICTABLE DISPLAY %{TTBR}[%{MSB:d}:0] should be zero, but are not. Fields:

MSB unsigned int

x-1.

TTBR enum

which TTBR register.

ArchMsg.Warning.warning_unaligned_address_dbgdttrrx_write

TAGS UNPREDICTABLE DISPLAY Unaligned address %{ADDR} in X0/R0 when DBGDTTRX written in memory access mode. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unaligned_address_dbgdttrtx_read

TAGS UNPREDICTABLE DISPLAY Unaligned address %{ADDR} in X0/R0 when DBGDTRTX read in memory access mode. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unaligned_to_device

TAGS UNPREDICTABLE DISPLAY Unaligned Access to Device Memory %{ADDR}. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unaligned_to_strongly_ordered

TAGS UNPREDICTABLE DISPLAY Unaligned Access to Strongly Ordered Memory %{ADDR}. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unknown_sau_rnr

TAGS UNPREDICTABLE DISPLAY SAU_RNR was set to an unsupported value.

ArchMsg.Warning.warning_unpredictable_AIRCR_PRIS_and_BFHFNMINs

TAGS UNPREDICTABLE DISPLAY The effect of setting both AIRCR.BFHFNMINs and AIRCR.PRIS to 1 is UNPREDICTABLE.

ArchMsg.Warning.warning_unpredictable_AIRCR_VECTCLRACTIVE_when_not_in_debug

TAGS UNPREDICTABLE DISPLAY The effect of writing a 1 to AIRCR.VECTCLRACTIVE if the processor is not halted in Debug state is UNPREDICTABLE.

ArchMsg.Warning.warning_unpredictable_AIRCR_VECTRESET_and_SYSRESETREQ

TAGS **UNPREDICTABLE** DISPLAY When the processor is halted in Debug state, if a write to the register writes a 1 to both VECTRESET and SYSRESETREQ, the behavior is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_AIRCR_VECTRESET_when_not_in_debug

TAGS **UNPREDICTABLE** DISPLAY The effect of writing a 1 to AIRCR.VECTRESET if the processor is not halted in Debug state is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_AIRCR_incorrect_VKEY

TAGS **UNPREDICTABLE** DISPLAY The value 0x05FA must be written to AIRCR.VKEY, otherwise the register write is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_EXC_RETURN_Reserved_Bit

EXC_RETURN[23:7] are reserved with the special condition that all bits should be written as one. Values other than all 1s are **UNPREDICTABLE**. TAGS **UNPREDICTABLE**. Fields:

EXC_RETURN unsigned int

Special PC value.

SBO_MASK unsigned int

Which bits Should Be One to not UNPRED.

ArchMsg.Warning.warning_unpredictable_PMEVTYPEPn_ELO_VS_reserved_value

TAGS **UNPREDICTABLE** DISPLAY Unpredictable reserved value of PMEVTYPEN<n>_ELO.VS.

ArchMsg.Warning.warning_unpredictable_change_to_DEMCR_MON_STEP_at_insufficient_priority

TAGS **UNPREDICTABLE** DISPLAY The effect of changing DEMCR.MON_STEP at an execution priority that is lower than the priority of the DebugMonitor exception is **CONSTRAINED UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_change_to_priority_of_active_exception

TAGS **UNPREDICTABLE** DISPLAY Changing the priority of an active exception is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_clear_lspact

TAGS **UNPREDICTABLE** DISPLAY **UNPREDICTABLE** state on context switch. Fields:

DRVEXC_TAKEN unsigned int

Derived exception taken, not pended.

REACHED_LAST_STORE unsigned int

Only the last faulted.

ArchMsg.Warning.warning_unpredictable_dbg_exit_unaligned_dlr

TAGS **UNPREDICTABLE** DISPLAY Debug exit to AArch32 with DLR[0] set to 1. PC[0] can be set to 0 or to DLR[0].

ArchMsg.Warning.warning_unpredictable_exception_catch

TAGS **UNPREDICTABLE** DISPLAY Generation of an exception catch event is unpredictable.

ArchMsg.Warning.warning_unpredictable_exception_return_inconsistent_state

TAGS **UNPREDICTABLE** DISPLAY state on exception return is unpredictable.

ArchMsg.Warning.warning_unpredictable_exception_return_instruction

TAGS **UNPREDICTABLE** DISPLAY exception return instruction is unpredictable.

ArchMsg.Warning.warning_unpredictable_in_debug_state

TAGS **UNPREDICTABLE** DISPLAY Instruction is **UNPREDICTABLE** when executed in debug state.

ArchMsg.Warning.warning_unpredictable_mair_encoding

TAGS **UNPREDICTABLE** DISPLAY Memory attribute %{ATTR} has invalid MAIR encoding; this may lead to unpredictable results. Fields:

ATTR unsigned int

Memory attribute.

ArchMsg.Warning.warning_unpredictable_pmu_counter_access

TAGS **UNPREDICTABLE** DISPLAY Access to PMU counters from non-secure EL0 or EL1 is **UNPREDICTABLE** with MDCR_EL2.HPMN set to 0.

ArchMsg.Warning.warning_unpredictable_prioritization_breakpoint_match_vector_catch

TAGS **UNPREDICTABLE** DISPLAY Two events with the same priority occurred at the same instruction: Address Matching Vector Catch debug event, and Breakpoint debug event. It is constrained unpredictable which is taken.

ArchMsg.Warning.warning_unpredictable_stack_selection

TAGS **UNPREDICTABLE** DISPLAY **UNPREDICTABLE** state on context switch.

ArchMsg.Warning.warning_unpredictable_tsize_out_of_range

TAGS **UNPREDICTABLE** DISPLAY %{TCR}.T%{TTBR:d}SZ (%{TSIZE:d} bits) is out of range. Must be between 25 and %{TSIZE_MAX:d}. Fields:

TCR enum

Which TCR.

TSIZE signed int

Value of TSize.

TSIZE_MAX signed int

The maximum allowed value.

TTBR unsigned int

Which TTBR region.

ArchMsg.Warning.warning_unpredictable_unaligned_pc_as_base_register

TAGS **UNPREDICTABLE** DISPLAY The PC value must be word-aligned, otherwise the behavior of the instruction is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_unaligned_pop_stack

TAGS **UNPREDICTABLE** DISPLAY **CONSTRAINED UNPREDICTABLE** unaligned stack frame %{FRAME_ADDR:x} in exception-return. Fields:

FRAME_ADDR unsigned int

unaligned stack frame address.

ArchMsg.Warning.warning_unpredictable_unaligned_to_non_writeback

TAGS **UNPREDICTABLE** DISPLAY Unaligned Access to Non-Writeback-Cacheable Memory.

ArchMsg.Warning.warning_unpredictable_vmsa_memattrib

TAGS **UNPREDICTABLE** DISPLAY Unpredictable vmsa memory attribute with texcb value programmed to %{TEXCB} when tex remap is %{TEX_REMAP}. Fields:

TEXCB unsigned int

texcb value.

TEX_REMAP bool

use tex remap.

ArchMsg.Warning.warning_unpredictable_vtcr_t0sz_sl0

TAGS **UNPREDICTABLE** DISPLAY The combination of VTCR.TOSZ=%{TOSZ:d} and VTCR.SL0=%{SL0:d} is unpredictable. Fields:

SL0 signed int

VTCR.SL0, starting level for stage 2 translation table walks.

TOSZ signed int

VTCR.TOSZ, required input address range.

ArchMsg.Warning.warning_unpredictable_write_DHCSR

TAGS **UNPREDICTABLE** DISPLAY **UNPREDICTABLE** set of bit-changes in DHCSR. Fields:

NEW unsigned int

data attempted to be written.

OLD unsigned int

previous value.

ArchMsg.Warning.warning_unpredictable_write_SHCSR

TAGS **UNPREDICTABLE** DISPLAY **UNPREDICTABLE** setting a pending bit to 1 for an exception with priority >= execution priority.

ArchMsg.Warning.warning_unsupported_access_to_memory_mapped_register

TAGS **UNPREDICTABLE** DISPLAY Access to memory mapped register is unsupported.

ArchMsg.Warning.warning_user_jmcr_access

TAGS **UNPREDICTABLE** DISPLAY User %{WRITE:(write|read)} access to JMCR is unpredictable. Fields:

WRITE unsigned int

access is write.

ArchMsg.Warning.warning_wcr_mask_and_bas

TAGS **UNPREDICTABLE** DISPLAY DBGWCR_EL1.MASK is non-zero and BAS is not 0xff.

ArchMsg.Warning.warning_wcr_mask_reserved

TAGS **UNPREDICTABLE** DISPLAY DBGWCR_ELn.MASK is set to a reserved value.

ArchMsg.Warning.warning_wcr_non_configuous_bas

TAGS **UNPREDICTABLE** DISPLAY DBGWCR_EL1.BAS has non-contiguous set of ones.

ArchMsg.Warning.warning_wvr_masked_bit_not_zero

TAGS **UNPREDICTABLE** DISPLAY DBGWVRn_EL1.VA contains a bit masked by DBGWCRn_EL1.MASK that is not zero.

ArchMsg.warning_unpred_system_register_access

DISPLAY Unpredictable %{IS_WRITE:(read|write)} access to system register offset %{OFFSET}. Fields:

IS_WRITE unsigned int

Write Not Read.

OFFSET unsigned int

Register Offset.

BRANCH_MISPREDICT

Simulating branch mispredict. Fields:

PC unsigned int

Origin address (or 0 if unavailable).

BRA_DIR

Direct branches, to immediate address. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISSET enum

The instruction set of the branch instruction.

IS_COND bool

Indicates if this is a conditional branch.

IS_HINTED bool

Whether the conditional instruction is hinted.

IS_ISB bool

Whether the instruction is ISB.

IS_LINK bool

Indicates if this is a branch with link.

PC unsigned int

The address of the branch instruction.

TARGET_ISET enum

The instruction set after the branch.

TARGET_PC unsigned int

The address the instruction branches to.

BRA_INDIR

Indirect branches, perhaps to a register. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISET enum

The instruction set of the branch instruction.

IS_COND bool

Indicates if this is a conditional branch.

IS_HINTED bool

Whether the conditional instruction is hinted.

IS_ISB bool

Whether the instruction is ISB.

IS_LINK bool

Indicates if this is a branch with link.

PC unsigned int

The address of the branch instruction.

TARGET_ISET enum

The instruction set after the branch.

TARGET_PC unsigned int

The address the instruction branches to.

BRA_INDIR_CCFAIL

Indirect branches fail, perhaps to a register.

BRA_RET_CCFAIL

Branch return fail, perhaps to a register.

CCFAIL

Conditional instruction condition check fail. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CCFAIL_UNC

Unconditional instruction condition check fail. Fields:

COND **enum**

The condition of the conditional instruction.

CORE_NUM **unsigned int**

Core number in a multi processor.

PC **unsigned int**

The address of the conditional instruction.

CCPASS

Conditional instruction condition check pass. Fields:

COND **enum**

The condition of the conditional instruction.

CORE_NUM **unsigned int**

Core number in a multi processor.

PC **unsigned int**

The address of the conditional instruction.

CCPASS_UNC

Unconditional instruction condition check pass. Fields:

COND **enum**

The condition of the conditional instruction.

CORE_NUM **unsigned int**

Core number in a multi processor.

PC **unsigned int**

The address of the conditional instruction.

CODE_CACHE_MAINT

Code cache maintenance. Fields:

END_INCL **unsigned int**

Inclusive end address for VA or PA requests.

NS **bool**

Non-secure world for PA.

START **unsigned int**

Start address for VA or PA requests.

TYPE **enum**

Request type.

COMPILE_BLOCK_END

Last instruction of basic block translated. Fields:

VADDR unsigned int

Address of next instruction after this basic block.

COMPILE_BLOCK_START

First instruction of basic block translated. Fields:

VADDR unsigned int

Address of first instruction in basic block.

COMPILE_INST

ARM instruction compiled. Fields:

DISASS string

Disassembly of the instruction.

ISSET enum

The instruction set of this instruction.

ITSTATE unsigned int

The ITSTATE current for the instruction.

OPCODE unsigned int

The opcode of the instruction.

PC unsigned int

The address of the instruction.

SIZE unsigned int

The size of the instruction in bytes.

CONTEXTIDR

Context ID Register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

NS bool

Secure or nonsecure banked register is accessed.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The new value written.

CONTEXT_SYNC

Called for every context synchronization event. Fields:

ARCH_CSE bool

Architectural Context Synchronization Event.

PC unsigned int

The address of the synchronization instruction.

SYSREG_SYNC bool

System register synchronization event.

CORE_ENDIAN

Core BE8 Big-Endian state changed. Fields:

BE8 bool

Core BE8 Big-Endian state.

CORE_INFO

Static processor attributes. Only triggered by a call to DumpState(). Fields:

ARCH_PROFILE enum

The architecture profile of the core.

CLUSTER_ID unsigned int

The cluster ID of this processor.

CORE_NUM unsigned int

The number of this core in an MP processor.

DETAILED_CORE_FEATURES enum

Detailed core features as a bitfield. Enum values are bitmasks to extract the info from the field.

FPU_VERSION enum

The VFP version implemented by the core.

MEM_ARCH enum

The memory architecture of the core.

MIDR_EL1 unsigned int

MIDR_EL1 value.

NUM_CORES unsigned int

The number of cores in this MP processor.

QUANTUM_SIZE unsigned int

The default quantum size of the core (in ticks).

SECURITY_FEATURES bool

Does the core have security features?.

CORE_LOADS

Processor load accesses. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ACQREL enum

Is this an acquire/release.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DATA unsigned int

The data read or written.

ELEMENT_SIZE unsigned int

Width of each element.

ENDIANNESS enum

Byte ordering of this access.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

LOCK enum

Normal, exclusive or locked access.

MECID unsigned int

Memory Encryption Context ID, if applicable.

MECID2 unsigned int

Memory Encryption Context ID of the second page, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PAGE_SIZE unsigned int

Page size as $\log_2(\text{size in bytes})$.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK, 2=Exclusive Failed, 3=Unsupported.

SIZE unsigned int

Width of the access in bytes. Only required if DATA is not traced.

TRANS bool

Is this a translated access.

VADDR unsigned int

The virtual address of the access.

CORE_REGS

Changes of the core registers R0 to R14. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number, 0 to 14.

OLD_VALUE unsigned int

The old value overwritten.

PHYS_ID enum

The physical register accessed.

VALUE unsigned int

The new value written to the register.

CORE_STALLED_TRANSITION

Detect when a core is prevented from progressing PChannel states. Fields:

CURRENT_STATE enum

Current power mode of core.

REASON string

The reason a transition was stalled, and what must be done to unblock it.

TARGET_STATE enum

Desired power mode of core after transition.

CORE_STORES

Processor store accesses. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ACQREL enum

Is this an acquire/release.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DATA unsigned int

The data read or written.

ELEMENT_SIZE unsigned int

Width of each element.

ENDIANNESS enum

Byte ordering of this access.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

LOCK enum

Normal, exclusive or locked access.

MECID unsigned int

Memory Encryption Context ID, if applicable.

MECID2 unsigned int

Memory Encryption Context ID of the second page, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PAGE_SIZE unsigned int

Page size as $\log_2(\text{size in bytes})$.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK, 2=Exclusive Failed, 3=Unsupported.

SIZE unsigned int

Width of the access in bytes. Only required if DATA is not traced.

TRANS bool

Is this a translated access.

VADDR unsigned int

The virtual address of the access.

CRYPTO_SPEC

Every crypto instruction speculatively executed.

DATA_CACHE_SET_ALLOC_TAGS

Set allocation tags from DC GVA instruction. Fields:

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

NSDESC unsigned int

The security state of the access.

PADDR unsigned int

The physical address of the access.

PAS enum

The physical address space of the page.

RESPONSE enum

0=Aborted, 1=OK.

SIZE unsigned int

The size of memory in bytes.

TAG unsigned int

The value of the tag being set.

TAGGING_TYPE enum

Type of Memory tagging (Physical or Virtual).

TAG_VADDR unsigned int

The virtual address of the Virtual Tag if Virtual tagging is used; otherwise 0.

VADDR unsigned int

Virtual address used for setting allocation tags.

DATA_CACHE_ZERO

Zero and invalidate cache line from DC ZVA instruction. Fields:

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

NSDESC unsigned int

The security state of the access.

PADDR unsigned int

The physical address of the access.

PADDR_TAG unsigned int

If tag checked, the physical address tag for the access.

PAS enum

The physical address space of the page.

RESPONSE enum

0=Aborted, 1=OK.

SIZE unsigned int

The size of zero'd memory in bytes.

TAG_CHECKED bool

Is this access tag checked.

VADDR unsigned int

Virtual address of the zero'd cache line.

DEBUG_EVENT

Hardware debug support event. Fields:

EVENT enum

Description of event.

VALUE unsigned int

data value.

DMI_ALLOCATE

DMI allocated. Fields:

ACCESS_DIRECTION enum

READ / WRITE.

PADDR unsigned int

The physical address of the DMI.

PTR unsigned int

DMI range pointer.

SIDE enum

DSIDE / ISIDE.

SIZE unsigned int

The size of DMI.

VADDR unsigned int

The virtual address of the DMI..

DMI_HIT

DMI hit, access used DMI (only fire for small DMI regions i.e. less than a full page). Fields:

ACCESS_DIRECTION enum

READ / WRITE.

PADDR unsigned int

The physical address of the DMI.

SIDE enum

DSIDE / ISIDE.

SIZE unsigned int

The size of the DMI.

VADDR unsigned int

The virtual address of the DMI..

DMI_REVOKE

DMI revoked. Fields:

ACCESS_DIRECTION enum

READ / WRITE.

PADDR unsigned int

The physical address of the DMI.

SIDE enum

DSIDE / ISIDE.

SIZE unsigned int

The size of DMI.

VADDR unsigned int

The virtual address of the DMI..

DWT_MATCH

DWT comparator matches. Fields:

NUM unsigned int

DWT comparator number.

TYPE enum

DWT comparator configuration.

END_COMPILE

Compilation end. Fields:

END_OF_PAGE_COUNT unsigned int

Number of basic blocks exited due to page boundary since START_COMPILE.

FETCHFAIL_COUNT unsigned int

Number of basic blocks exited due to ifetch failure START_COMPILE.

INST_COUNT unsigned int

Number of instructions compiled since START_COMPILE.

NONSEQ_COUNT unsigned int

Number of basic blocks exited due to non-sequential instructions since START_COMPILE.

PAGE_STRADDLE_COUNT unsigned int

Number of basic blocks exited due to unaligned instructions crossing page since START_COMPILE.

EXCEPTION

Exceptions that are taken. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ISSET enum

The instruction set of the processor when the exception occurred.

LR unsigned int

The value assigned to the link register.

NS enum

The core's non-secure bit.

PC unsigned int

The location where the exception occurred.

TARGET_ISET enum

The instruction set of the exception handler code.

TARGET_PC unsigned int

The address the exception branches to.

VECTOR enum

The exception vector.

EXCEPTION_END

Every exception completed. Fields:

NS enum

The core's non-secure bit.

EXCEPTION_ENTRY

Event marking the entry of an exception. All pushes of registers to the stack will follow this event. Fields:

PC unsigned int

The location where the exception occurred.

VECTOR enum

The exception that occurred.

EXCEPTION_RAISE

Every exception raised.

EXCEPTION_RETURN

Marks the end of an exception, but on an M core does not indicate a program flow change (branch). In most situations it will be followed by an indirect branch. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISET enum

The instruction set of the branch instruction.

PC unsigned int

The address of the branch instruction.

EXCEPTION_RETURN_PREBRANCH

Exception return event that must occur before PC and processor state have updated.

EXCEPTION_START

Every exception started. Fields:

NS enum

The core's non-secure bit.

EXCEPTION_VECTOR_FETCH

I-side vector fetch for M-class exception. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

DATA unsigned int

Result of fetch if OK.

RESPONSE enum

0=Aborted, 1=OK.

VADDR unsigned int

Where the vector is fetched from (including VTOR).

FP_STATE

Floating point state. Fields:

FPSR unsigned int

FPSR register value.

MASK unsigned int

The mask indicating updated cumulative bits.

FREQ_CHANGED

Core frequency changed. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The instruction count of this core.

NEW_FREQ signed int

The new frequency of this core (expressed in Hz).

OLD_FREQ signed int

The old frequency of this core (expressed in Hz).

HLT

HLT instruction occurred. Fields:

IMM16 unsigned int

The 16-bit immediate value encoded in HLT instruction.

INFO_EXCEPTION_REASON

Provide information of real cause of exception. Fields:

FaultCause enum

Which bit of which V7M/V8M-mainline FSR/HFSR would have been set.

PC unsigned int

The location where the exception occurred.

PHASE enum

What we are doing with the exception.

REASONS enum

Why we are doing it.

SecurityState enum

Privacy State of exception.

VECTOR enum

The exception that occurred.

INFO_STACKING

Provide information of real cause of exception. Fields:

PHASE enum

Are we starting or ending a sequence of stack writes.

REASONS enum

Why we are doing it.

INST

Every instruction executed. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DEBUG_STATE bool

The instruction is executed in debug state.

DISASS string

Disassembly of instruction.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISET enum

The current instruction set.

ITSTATE unsigned int

The current ITSTATE.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

MODE enum

The mode the core is in.

NS unsigned int

The core's non-secure bit.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

OPCODE unsigned int

The opcode of the instruction.

PADDR unsigned int

The physical address of the instruction.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the instruction.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the page.

PC unsigned int

The address of the instruction.

SECURITY_STATE enum

The core's security state.

SIZE unsigned int

The size of the instruction in bytes.

INST_END

Every instruction completed.

INST_START

Every instruction started. Fields:

DEBUG_STATE bool

The instruction is executed in debug state.

ISSET enum

The current instruction set.

MODE enum

The mode the core is in.

NS enum

The current Secure State.

PC unsigned int

The address of the conditional instruction.

SECURITY_STATE enum

The current Security State.

INST_STRADDLE

Instruction straddles boundary. Fields:

DEBUG_STATE bool

The instruction is executed in debug state.

ISET enum

The current instruction set.

MODE enum

The mode the core is in.

NS enum

The current Secure State.

PC unsigned int

The address of the conditional instruction.

SECURITY_STATE enum

The current Security State.

ISB

ISB instruction occurred. Fields:

PC unsigned int

The address of the ISB instruction.

LOAD_MULTIPLE_REGS_LIST

List of destination registers (not source) of a multiple registers load instruction. Fields:

REG string

Register to add into the list.

LOCAL_MONITOR

Local monitor activity. Fields:

PADDR unsigned int

Local Monitor Address.

State enum

State of the monitor (Open/Exclusive).

LOCKUP_CYCLE

This event is triggered if the core enters or stays in the lock-up state. Fields:

CAUSE enum

What causes the lockup?.

LOCKUP_ENTRY

This event is triggered if the core enters the lock-up state. Fields:

PC unsigned int

The PC when the lockup state is entered.

LOCKUP_EXIT

This event is triggered if the core leaves the lock-up state.

MODE_CHANGE

Mode change. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

MODE enum

The new mode.

OLD_MODE enum

The old mode.

MPU_TRANS

Address translation information. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

EXEC_PERM enum

Execution Permission.

IDAU_REG_NUM enum

IDAU region-number or code.

INNERCACHE_RA bool

Is the inner cache allocate on read.

INNERCACHE_TYPE enum

Inner Caching scheme (NC/MB/WA).

INNERCACHE_WA bool

Is the inner cache allocate on write.

MEMTYPE enum

Memory type.

NSDESC enum

Security-attributes of the transaction.

OUTERCACHE_RA bool

Is the outer cache allocate on read.

OUTERCACHE_TYPE enum

Outer Caching scheme (NC/MB/WA).

OUTERCACHE_WA bool

Is the outer cache allocate on write.

PADDR unsigned int

Address of the access.

PAGESIZE unsigned int

Page size as log2(size).

PAS enum

PAS accessed is secure or nonsecure memory.

READ_PERM enum

Read Permission.

REG_NUM unsigned int

MPU register based region-number 0-15 (else 0xFF=none, 0xBB/0xDD=background/default).

SAU_REG_NUM enum

SAU region-number or code.

SAU_RGN_TYPE enum

Privacy Region Type.

SH enum

Shareability.

SIDE enum

Inst / Data.

WRITE_PERM enum

Write Permission.

PERIODIC

Called for every quantum. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The instruction count of this CPU.

PC unsigned int

The address of the next instruction to be executed on this CPU.

POTINDEX_UPDATE

Value of POT index for current pc changed. Fields:

NEW_POTINDEX unsigned int

New POTIndex.

OLD_POTINDEX unsigned int

Old POTIndex.

PPB_READ_ACCESS

Trace reads from the core to memory mapped registers in the PPB (private peripheral bus) address range. Fields:

ACCESS_SIZE unsigned int

Logarithm of the access size: 0=byte, 1=halfword, 2=word, 3=doubleword, ...

BURST_LEN unsigned int

The number of data transfers (beats) in this burst.

DATA unsigned int

The data transferred.

INNER_ATTRIBUTES unsigned int

The inner attributes: bit0=write allocate, bit1=read allocate, bit2=cachable, bit3=bufferable, bit4=shareable.

INST bool

Is an instruction fetch.

LOCK enum

Normal / Locked / Exclusive.

MANAGER_ID unsigned int

The AXI manager ID.

NS bool

Is a nonsecure access.

OUTER_ATTRIBUTES unsigned int

The outer attributes: bit0=write allocate, bit1=read allocate, bit2=cachable, bit3=bufferable, bit4=shareable.

PADDR unsigned int

Physical address of access.

PRIV bool

Is privileged access.

RESPONSE enum

Whether the transaction was successful, or an error occurred.

USER_FLAGS unsigned int

Core specific additional signals.

PPB_WRITE_ACCESS

Trace writes from the core to memory mapped registers in the PPB (private peripheral bus) address range. Fields:

ACCESS_SIZE unsigned int

Logarithm of the access size: 0=byte, 1=halfword, 2=word, 3=doubleword, ...

BURST_LEN unsigned int

The number of data transfers (beats) in this burst.

DATA unsigned int

The data transferred.

INNER_ATTRIBUTES unsigned int

The inner attributes: bit0=write allocate, bit1=read allocate, bit2=cachable, bit3=bufferable, bit4=shareable.

INST bool

Is an instruction fetch.

LOCK enum

Normal / Locked / Exclusive.

MANAGER_ID unsigned int

The AXI manager ID.

NS bool

Is a nonsecure access.

OUTER_ATTRIBUTES unsigned int

The outer attributes: bit0=write allocate, bit1=read allocate, bit2=cachable, bit3=bufferable, bit4=shareable.

PADDR unsigned int

Physical address of access.

PRIV bool

Is privileged access.

RESPONSE enum

Whether the transaction was successful, or an error occurred.

USER_FLAGS unsigned int

Core specific additional signals.

PRELOAD_DATA

Data preload from PLD instruction. Fields:

VADDR unsigned int

Virtual address of the data that should be preloaded.

PRELOAD_INST

Instruction preload from PLI instruction. Fields:

VADDR unsigned int

Virtual address of the instruction that should be preloaded.

PRE_CORE_LOAD

Trace just before a core load. Fields:

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

PRE_CORE_STORE

Trace just before a core store. Fields:

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

PSTATE_PACTIVE_OPMODE_TRANSITION

Trace pactive and Opmode mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Operating mode after the transition.

OLD_STATE enum

Operating mode before the transition.

PSTATE_PACTIVE_OPMODE_TRANSITION_DENIED

Trace the denied pactive and Opmode mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Operating mode after the transition.

OLD_STATE enum

Operating mode before the transition.

REASON string

Reason for the transition being denied.

PSTATE_PACTIVE_POWER_TRANSITION

Trace pactive and power mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Power mode after the transition.

OLD_STATE enum

Power mode before the transition.

PSTATE_PACTIVE_POWER_TRANSITION_DENIED

Trace the denied pactive and power mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Power mode after the transition.

OLD_STATE enum

Power mode before the transition.

REASON string

Reason for the transition being denied.

PSTATE_PACTIVE_THREAD_OPMODE_TRANSITION

Trace pactive and Opmode mode transitions for threads on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Operating mode after the transition.

OLD_STATE enum

Operating mode before the transition.

QUANTUM_START

Called at the start of every quantum.

RUN_STATE

Run state transition. Fields:

INST_COUNT unsigned int

Ticks count at point of transition.

NEW enum

New run state.

OLD enum

Old run state.

SEMIHOSTING_CALL

Call of a semihost function occurred. Fields:

PC unsigned int

The program counter after the semihosting call.

SEMIHOSTING_PRECALL

About to call semihost. Fields:

EL2 bool

Use EL2 translation regime.

NS bool

Is Non-Secure.

REG_WIDTH unsigned int

The current register width in bytes.

SIGNAL

External signal state change. Fields:

SIGNAL enum

Signal that changed.

STATE bool

Signal asserted state.

START_COMPILE

Compilation started. Fields:

VADDR unsigned int

Instruction where compilation begins.

STG_MULTIPLE_MICRO_OPS_ADDRESS_RANGE

Address range for MTE stores with multiple uops. Fields:

ADDR_END unsigned int

Final address.

ADDR_START unsigned int

Starting address.

STORE_MULTIPLE_REGS_ADDRESS_RANGE

Address range for stores with multiple registers. Fields:

ADDR_END unsigned int

Final address.

ADDR_START unsigned int

Starting address.

SUBOPTIMAL_LDST_RETIRED

suboptimal load/store. Fields:

ACCESS_SIZE unsigned int

Log2 of the access width in bytes used for alignment checking (1: half-word, 2: word ...).

OPTIMAL_ALIGN_SELECTOR unsigned int

Selector to decide the optimal alignment size (1: access width, 2: 4-bytes, 3: 8-bytes, ..., 12: 4-kbytes).

VADDR unsigned int

The virtual address of the access.

SYNC

Called for every synchronization, that is every quantum break. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The instruction count of this CPU.

LOCAL_QUANTUM unsigned int

The local quantum of this CPU.

LOCAL_TIME unsigned int

The local time of this CPU.

SYNC_REASON enum

Reason for sync.

SYSCALL

System call instruction executed. Fields:

IMM unsigned int

Immediate value of the system call instruction.

TYPE enum

System call type.

VADDR unsigned int

Instruction that caused the system call.

SYSREG_READ

Trace reads from system registers. Fields:

REG_NAME enum

Register number.

VALUE unsigned int

The value read.

SYSREG_UPDATE128

Triggers when the system updates a register. Fields:

REG enum

Register number.

UNKNOWN unsigned int

Bits of the register which became unknown.

VALUE unsigned int

Value written to the register.

SYSREG_UPDATE32

Triggers when the system updates a register. Fields:

REG enum

Register number.

UNKNOWN unsigned int

Bits of the register which became unknown.

VALUE unsigned int

Value written to the register.

SYSREG_UPDATE64

Triggers when the system updates a register. Fields:

REG enum

Register number.

UNKNOWN unsigned int

Bits of the register which became unknown.

VALUE unsigned int

Value written to the register.

SYSREG_WRITE

Trace write to system registers. Fields:

REG_ACCESS enum

Register access status.

REG_NAME enum

Register number.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

Value written to the register.

UNALIGNED_LDST_RETIRED

Processor unaligned load/store. Fields:

ACCESS_SIZE unsigned int

Log2 of the access width in bytes used for alignment checking (1=>half-word, 2->word ...).

NUMBER_OF_BEATS unsigned int

Number of accesses (beats) in this burst (Total size in bytes is ACCESS_SIZE * NUMBER_OF_BEATS).

VADDR unsigned int

The virtual address of the access.

WAYPOINT

Signals end of basic block execution. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ISSET enum

Origin instruction set.

IS_COND bool

Indicates if this is a conditional waypoint.

PC unsigned int

Origin address (or 0 if unavailable).

TAKEN bool

Indicates if this waypoint was taken.

TARGET unsigned int

Destination address.

TARGET_ISET enum

Destination instruction set.

WFE_END

WFE ended. Fields:

INST_COUNT unsigned int

Ticks count when leaving WFE.

WFE_EVENT_REGISTER

WFE event register status: set/clear, reason. Fields:

INST_COUNT unsigned int

Ticks count.

REASON enum

Reason for set/clear. Only REASON==0 (Cleared by WFE) clears the bit, all other reasons set it.

WFE_IGNORED

WFE ignored. Fields:

EVENT bool

This WFE was ignored because the event register was set.

INST_COUNT unsigned int

Ticks count when ignoring WFE.

TRAPPED bool

This WFE was trapped.

WFE_START

WFE entered. Fields:

INST_COUNT unsigned int

Ticks count when entering WFE.

WFI_END

WFI ended. Fields:

INST_COUNT **unsigned int**

Ticks count when leaving WFI.

WFI_IGNORED

WFI ignored. Fields:

DISABLED **bool**

This WFI was ignored because WFI is disabled.

INST_COUNT **unsigned int**

Ticks count when ignoring WFI.

REASON **enum**

specifies reason why WFI trace was ignored.

TRAPPED **bool**

This WFI was trapped.

WFI_START

WFI entered. Fields:

INST_COUNT **unsigned int**

Ticks count when entering WFI.

WFI_WAKEUP

WFI wakeup. Fields:

INST_COUNT **unsigned int**

Ticks count when WFI wakeup occurred.

REASON **enum**

Reason for wakeup.

XPSR

Changes to the xPSR register. Fields:

CORE_NUM **unsigned int**

Core number in a multi processor.

OLD_VALUE **unsigned int**

The old xPSR value.

UNKNOWN **unsigned int**

Bits within the register that have unknown value.

VALUE **unsigned int**

The new xPSR value.

2.41 ARM_Cortex-M3

This section describes the trace sources.

ASYNC_MEMORY_FAULT

Context ID Register write. Fields:

FAULT unsigned int

Fault status present in ESR.ISS encoding for Serror in ARCH64 and present in DFSR encoding for external abort format in ARCH32.

PADDR unsigned int

Physical Address (or 0 if unavailable).

ATOMIC_END_ACCESS

Bus trace access for atomics. Fields:

ACCESS_FAIL bool

Memory access failed.

ADDR unsigned int

The virtual address of the access.

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

COMPARE_VALUE unsigned int

Compare value for CAS.

LOAD_VALUE unsigned int

Loaded values.

NSDESC unsigned int

The security state of the access.

OPERAND_VALUE unsigned int

Operation's operand.

OPERATION enum

Operation type.

PADDR unsigned int

The physical address of the access.

PRIV bool

Is this a privileged access?.

ATOMIC_START_ACCESS

Bus trace access for atomics. Fields:

ADDR unsigned int

The virtual address of the access.

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

COMPARE_VALUE unsigned int

Compare value for CAS.

NSDESC unsigned int

The security state of the access.

OPERAND_VALUE unsigned int

Operation's operand.

OPERATION enum

Operation type.

PADDR unsigned int

The physical address of the access.

PRIV bool

Is this a privileged access?.

ArchMsg.Error.exit_code

DISPLAY exit code: %{EXITCODE} component kind: %{KIND} reason: %{REASON}. Fields:

EXITCODE signed int

Exit code to be returned.

KIND string

Component kind that invoked the exit code trace.

REASON string

Reason why the exit code trace was invoked (optional).

ArchMsg.Warning.dap_csw_bad_size

A write to CM3DAP CSW has an invalid size field. DISPLAY Write %{DATA} to CM3DAP CSW has invalid size field. Fields:

DATA unsigned int

bits[2:0] are size.

ArchMsg.Warning.recursive_branch

An instruction is performing a branch that targets the same instruction. If this was intended then a WFI might be more effective DISPLAY recursive branch detected at PC=%{PC}. PRIMARY KEY PC. Fields:

PC unsigned int

Address of instruction causing the branch.

ArchMsg.Warning.recursive_exception

An instruction has generated an exception that targets the same instruction. DISPLAY recursive %{TYPE} exception detected at PC=%{PC} (CPSR=%{CPSR}). PRIMARY KEY PC. Fields:

CPSR unsigned int

Processor state.

PC unsigned int

Address of instruction causing the exception.

TYPE enum

Type of exception.

ArchMsg.Warning.secure_vector_fetch_from_nonsecure

DISPLAY Secure vector table is being fetched from Non-secure memory.

ArchMsg.Warning.sg_in_it_block

Arm recommends that software does not place SG (security gate instruction) inside an IT instruction block.

ArchMsg.warning_unpred_system_register_access

DISPLAY Unpredictable %{IS_WRITE:(read|write)} access to system register offset %{OFFSET}. Fields:

IS_WRITE unsigned int

Write Not Read.

OFFSET unsigned int

Register Offset.

BRANCH_MISPREDICT

Simulating branch mispredict. Fields:

PC unsigned int

Origin address (or 0 if unavailable).

BRA_DIR

Direct branches, to immediate address. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISET enum

The instruction set of the branch instruction.

IS_COND bool

Indicates if this is a conditional branch.

IS_HINTED bool

Whether the conditional instruction is hinted.

IS_ISB bool

Whether the instruction is ISB.

IS_LINK bool

Indicates if this is a branch with link.

PC unsigned int

The address of the branch instruction.

TARGET_ISET enum

The instruction set after the branch.

TARGET_PC unsigned int

The address the instruction branches to.

BRA_INDIR

Indirect branches, perhaps to a register. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISET enum

The instruction set of the branch instruction.

IS_COND bool

Indicates if this is a conditional branch.

IS_HINTED bool

Whether the conditional instruction is hinted.

IS_ISB bool

Whether the instruction is ISB.

IS_LINK bool

Indicates if this is a branch with link.

PC unsigned int

The address of the branch instruction.

TARGET_ISET enum

The instruction set after the branch.

TARGET_PC unsigned int

The address the instruction branches to.

BRA_INDIR_CCFAIL

Indirect branches fail, perhaps to a register.

BRA_RET_CCFAIL

Branch return fail, perhaps to a register.

CCFAIL

Conditional instruction condition check fail. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CCFAIL_UNC

Unconditional instruction condition check fail. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CCPASS

Conditional instruction condition check pass. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CCPASS_UNC

Unconditional instruction condition check pass. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CODE_CACHE_MAINT

Code cache maintenance. Fields:

END_INCL unsigned int

Inclusive end address for VA or PA requests.

NS bool

Non-secure world for PA.

START unsigned int

Start address for VA or PA requests.

TYPE enum

Request type.

COMPILE_BLOCK_END

Last instruction of basic block translated. Fields:

VADDR unsigned int

Address of next instruction after this basic block.

COMPILE_BLOCK_START

First instruction of basic block translated. Fields:

VADDR unsigned int

Address of first instruction in basic block.

COMPILE_INST

ARM instruction compiled. Fields:

DISASS string

Disassembly of the instruction.

ISSET enum

The instruction set of this instruction.

ITSTATE unsigned int

The ITSTATE current for the instruction.

OPCODE unsigned int

The opcode of the instruction.

PC unsigned int

The address of the instruction.

SIZE unsigned int

The size of the instruction in bytes.

CONTEXTIDR

Context ID Register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

NS bool

Secure or nonsecure banked register is accessed.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The new value written.

CONTEXT_SYNC

Called for every context synchronization event. Fields:

ARCH_CSE bool

Architectural Context Synchronization Event.

PC unsigned int

The address of the synchronization instruction.

SYSREG_SYNC bool

System register synchronization event.

CORE_ENDIAN

Core BE8 Big-Endian state changed. Fields:

BE8 bool

Core BE8 Big-Endian state.

CORE_INFO

Static processor attributes. Only triggered by a call to DumpState(). Fields:

ARCH_PROFILE enum

The architecture profile of the core.

CLUSTER_ID unsigned int

The cluster ID of this processor.

CORE_NUM unsigned int

The number of this core in an MP processor.

DETAILED_CORE_FEATURES enum

Detailed core features as a bitfield. Enum values are bitmasks to extract the info from the field.

FPU_VERSION enum

The VFP version implemented by the core.

MEM_ARCH enum

The memory architecture of the core.

MIDR_EL1 unsigned int

MIDR_EL1 value.

NUM_CORES unsigned int

The number of cores in this MP processor.

QUANTUM_SIZE unsigned int

The default quantum size of the core (in ticks).

SECURITY_FEATURES bool

Does the core have security features?.

CORE_LOADS

Processor load accesses. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ACQREL enum

Is this an acquire/release.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DATA unsigned int

The data read or written.

ELEMENT_SIZE unsigned int

Width of each element.

ENDIANNESS enum

Byte ordering of this access.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

LOCK enum

Normal, exclusive or locked access.

MECID unsigned int

Memory Encryption Context ID, if applicable.

MECID2 unsigned int

Memory Encryption Context ID of the second page, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PAGE_SIZE unsigned int

Page size as $\log_2(\text{size in bytes})$.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK, 2=Exclusive Failed, 3=Unsupported.

SIZE unsigned int

Width of the access in bytes. Only required if DATA is not traced.

TRANS bool

Is this a translated access.

VADDR unsigned int

The virtual address of the access.

CORE_REGS

Changes of the core registers R0 to R14. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number, 0 to 14.

OLD_VALUE unsigned int

The old value overwritten.

PHYS_ID enum

The physical register accessed.

VALUE unsigned int

The new value written to the register.

CORE_STALLED_TRANSITION

Detect when a core is prevented from progressing PChannel states. Fields:

CURRENT_STATE enum

Current power mode of core.

REASON string

The reason a transition was stalled, and what must be done to unblock it.

TARGET_STATE enum

Desired power mode of core after transition.

CORE_STORES

Processor store accesses. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ACQREL enum

Is this an acquire/release.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DATA unsigned int

The data read or written.

ELEMENT_SIZE unsigned int

Width of each element.

ENDIANNESS enum

Byte ordering of this access.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

LOCK enum

Normal, exclusive or locked access.

MECID unsigned int

Memory Encryption Context ID, if applicable.

MECID2 unsigned int

Memory Encryption Context ID of the second page, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PAGE_SIZE unsigned int

Page size as $\log_2(\text{size in bytes})$.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK, 2=Exclusive Failed, 3=Unsupported.

SIZE unsigned int

Width of the access in bytes. Only required if DATA is not traced.

TRANS bool

Is this a translated access.

VADDR unsigned int

The virtual address of the access.

CRYPTO_SPEC

Every crypto instruction speculatively executed.

DATA_CACHE_SET_ALLOC_TAGS

Set allocation tags from DC GVA instruction. Fields:

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

NSDESC unsigned int

The security state of the access.

PADDR unsigned int

The physical address of the access.

PAS enum

The physical address space of the page.

RESPONSE enum

0=Aborted, 1=OK.

SIZE unsigned int

The size of memory in bytes.

TAG unsigned int

The value of the tag being set.

TAGGING_TYPE enum

Type of Memory tagging (Physical or Virtual).

TAG_VADDR unsigned int

The virtual address of the Virtual Tag if Virtual tagging is used; otherwise 0.

VADDR unsigned int

Virtual address used for setting allocation tags.

DATA_CACHE_ZERO

Zero and invalidate cache line from DC ZVA instruction. Fields:

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

NSDESC unsigned int

The security state of the access.

PADDR unsigned int

The physical address of the access.

PADDR_TAG unsigned int

If tag checked, the physical address tag for the access.

PAS enum

The physical address space of the page.

RESPONSE enum

0=Aborted, 1=OK.

SIZE unsigned int

The size of zero'd memory in bytes.

TAG_CHECKED bool

Is this access tag checked.

VADDR unsigned int

Virtual address of the zero'd cache line.

DEBUG_EVENT

Hardware debug support event. Fields:

EVENT enum

Description of event.

VALUE unsigned int

data value.

DMI_ALLOCATE

DMI allocated. Fields:

ACCESS_DIRECTION enum

READ / WRITE.

PADDR unsigned int

The physical address of the DMI.

PTR unsigned int

DMI range pointer.

SIDE enum

DSIDE / ISIDE.

SIZE unsigned int

The size of DMI.

VADDR unsigned int

The virtual address of the DMI..

DMI_HIT

DMI hit, access used DMI (only fire for small DMI regions i.e. less than a full page). Fields:

ACCESS_DIRECTION enum

READ / WRITE.

PADDR unsigned int

The physical address of the DMI.

SIDE enum

DSIDE / ISIDE.

SIZE unsigned int

The size of the DMI.

VADDR unsigned int

The virtual address of the DMI..

DMI_REVOKE

DMI revoked. Fields:

ACCESS_DIRECTION enum

READ / WRITE.

PADDR unsigned int

The physical address of the DMI.

SIDE enum

DSIDE / ISIDE.

SIZE unsigned int

The size of DMI.

VADDR unsigned int

The virtual address of the DMI..

DWT_MATCH

DWT comparator matches. Fields:

NUM unsigned int

DWT comparator number.

TYPE enum

DWT comparator configuration.

END_COMPILE

Compilation end. Fields:

END_OF_PAGE_COUNT unsigned int

Number of basic blocks exited due to page boundary since START_COMPILE.

FETCHFAIL_COUNT unsigned int

Number of basic blocks exited due to ifetch failure START_COMPILE.

INST_COUNT unsigned int

Number of instructions compiled since START_COMPILE.

NONSEQ_COUNT unsigned int

Number of basic blocks exited due to non-sequential instructions since START_COMPILE.

PAGE_STRADDLE_COUNT unsigned int

Number of basic blocks exited due to unaligned instructions crossing page since START_COMPILE.

EXCEPTION

Exceptions that are taken. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ISET enum

The instruction set of the processor when the exception occurred.

LR unsigned int

The value assigned to the link register.

PC unsigned int

The location where the exception occurred.

TARGET_ISET enum

The instruction set of the exception handler code.

TARGET_PC unsigned int

The address the exception branches to.

VECTOR enum

The exception vector.

EXCEPTION_END

Every exception completed.

EXCEPTION_ENTRY

Event marking the entry of an exception. All pushes of registers to the stack will follow this event. Fields:

PC unsigned int

The location where the exception occurred.

VECTOR enum

The exception that occurred.

EXCEPTION_RAISE

Every exception raised.

EXCEPTION_RETURN

Marks the end of an exception, but on an M core does not indicate a program flow change (branch). In most situations it will be followed by an indirect branch. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISSET enum

The instruction set of the branch instruction.

PC unsigned int

The address of the branch instruction.

EXCEPTION_RETURN_PREBRANCH

Exception return event that must occur before PC and processor state have updated.

EXCEPTION_START

Every exception started.

EXCEPTION_VECTOR_FETCH

I-side vector fetch for M-class exception. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

DATA unsigned int

Result of fetch if OK.

RESPONSE enum

0=Aborted, 1=OK.

VADDR unsigned int

Where the vector is fetched from (including VTOR).

FP_STATE

Floating point state. Fields:

FPSR unsigned int

FPSR register value.

MASK unsigned int

The mask indicating updated cumulative bits.

FREQ_CHANGED

Core frequency changed. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The instruction count of this core.

NEW_FREQ signed int

The new frequency of this core (expressed in Hz).

OLD_FREQ signed int

The old frequency of this core (expressed in Hz).

HLT

HLT instruction occurred. Fields:

IMM16 unsigned int

The 16-bit immediate value encoded in HLT instruction.

INFO_EXCEPTION_REASON

Provide information of real cause of exception. Fields:

FaultCause enum

Which bit of which V7M/V8M-mainline FSR/HFSR would have been set.

PC unsigned int

The location where the exception occurred.

PHASE enum

What we are doing with the exception.

REASONS enum

Why we are doing it.

VECTOR enum

The exception that occurred.

INFO_STACKING

Provide information of real cause of exception. Fields:

PHASE enum

Are we starting or ending a sequence of stack writes.

REASONS enum

Why we are doing it.

INST

Every instruction executed. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DEBUG_STATE bool

The instruction is executed in debug state.

DISASS string

Disassembly of instruction.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISET enum

The current instruction set.

ITSTATE unsigned int

The current ITSTATE.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

MODE enum

The mode the core is in.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

OPCODE unsigned int

The opcode of the instruction.

PADDR unsigned int

The physical address of the instruction.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the instruction.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the page.

PC unsigned int

The address of the instruction.

SECURITY_STATE enum

The core's security state.

SIZE unsigned int

The size of the instruction in bytes.

INST_END

Every instruction completed.

INST_START

Every instruction started. Fields:

DEBUG_STATE bool

The instruction is executed in debug state.

ISSET enum

The current instruction set.

MODE enum

The mode the core is in.

NS enum

The current Secure State.

PC unsigned int

The address of the conditional instruction.

SECURITY_STATE enum

The current Security State.

INST_STRADDLE

Instruction straddles boundary. Fields:

DEBUG_STATE bool

The instruction is executed in debug state.

ISET enum

The current instruction set.

MODE enum

The mode the core is in.

NS enum

The current Secure State.

PC unsigned int

The address of the conditional instruction.

SECURITY_STATE enum

The current Security State.

ISB

ISB instruction occurred. Fields:

PC unsigned int

The address of the ISB instruction.

ITM

Instrumentation Trace Macrocell. Fields:

ITM_PACKET_TYPE enum

ITM and DWT packets type.

PACKET_HEADER unsigned int

ITM Packet Header.

PACKET_PAYLOAD unsigned int

ITM Packet Payload.

LOAD_MULTIPLE_REGS_LIST

List of destination registers (not source) of a multiple registers load instruction. Fields:

REG string

Register to add into the list.

LOCAL_MONITOR

Local monitor activity. Fields:

PADDR unsigned int

Local Monitor Address.

State enum

State of the monitor (Open/Exclusive).

LOCKUP_CYCLE

This event is triggered if the core enters or stays in the lock-up state. Fields:

CAUSE enum

What causes the lockup?.

LOCKUP_ENTRY

This event is triggered if the core enters the lock-up state. Fields:

PC unsigned int

The PC when the lockup state is entered.

LOCKUP_EXIT

This event is triggered if the core leaves the lock-up state.

MODE_CHANGE

Mode change. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

MODE enum

The new mode.

OLD_MODE enum

The old mode.

MPU_TRANS

Address translation information. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

EXEC_PERM enum

Execution Permission.

INNERCACHE_RA bool

Is the inner cache allocate on read.

INNERCACHE_TYPE enum

Inner Caching scheme (NC/MB/WA).

INNERCACHE_WA bool

Is the inner cache allocate on write.

MEMTYPE enum

Memory type.

OUTERCACHE_RA bool

Is the outer cache allocate on read.

OUTERCACHE_TYPE enum

Outer Caching scheme (NC/MB/WA).

OUTERCACHE_WA bool

Is the outer cache allocate on write.

PADDR unsigned int

Address of the access.

PAGESIZE unsigned intPage size as $\log_2(\text{size})$.**READ_PERM enum**

Read Permission.

REG_NUM unsigned int

MPU register based region-number 0-15 (else 0xFF=none, 0xBB/0xDD=background/default).

SH enum

Shareability.

SIDE enum

Inst / Data.

WRITE_PERM enum

Write Permission.

PERIODIC

Called for every quantum. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The instruction count of this CPU.

PC unsigned int

The address of the next instruction to be executed on this CPU.

POTINDEX_UPDATE

Value of POT index for current pc changed. Fields:

NEW_POTINDEX unsigned int

New POTIndex.

OLD_POTINDEX unsigned int

Old POTIndex.

PPB_READ_ACCESS

Trace reads from the core to memory mapped registers in the PPB (private peripheral bus) address range. Fields:

ACCESS_SIZE unsigned int

Logarithm of the access size: 0=byte, 1=halfword, 2=word, 3=doubleword, ...

BURST_LEN unsigned int

The number of data transfers (beats) in this burst.

DATA unsigned int

The data transferred.

INNER_ATTRIBUTES unsigned int

The inner attributes: bit0=write allocate, bit1=read allocate, bit2=cachable, bit3=bufferable, bit4=shareable.

INST bool

Is an instruction fetch.

LOCK enum

Normal / Locked / Exclusive.

MANAGER_ID unsigned int

The AXI manager ID.

NS bool

Is a nonsecure access.

OUTER_ATTRIBUTES unsigned int

The outer attributes: bit0=write allocate, bit1=read allocate, bit2=cachable, bit3=bufferable, bit4=shareable.

PADDR unsigned int

Physical address of access.

PRIV bool

Is privileged access.

RESPONSE enum

Whether the transaction was successful, or an error occurred.

USER_FLAGS unsigned int

Core specific additional signals.

PPB_WRITE_ACCESS

Trace writes from the core to memory mapped registers in the PPB (private peripheral bus) address range. Fields:

ACCESS_SIZE unsigned int

Logarithm of the access size: 0=byte, 1=halfword, 2=word, 3=doubleword, ...

BURST_LEN unsigned int

The number of data transfers (beats) in this burst.

DATA unsigned int

The data transferred.

INNER_ATTRIBUTES unsigned int

The inner attributes: bit0=write allocate, bit1=read allocate, bit2=cachable, bit3=bufferable, bit4=shareable.

INST bool

Is an instruction fetch.

LOCK enum

Normal / Locked / Exclusive.

MANAGER_ID unsigned int

The AXI manager ID.

NS bool

Is a nonsecure access.

OUTER_ATTRIBUTES unsigned int

The outer attributes: bit0=write allocate, bit1=read allocate, bit2=cachable, bit3=bufferable, bit4=shareable.

PADDR unsigned int

Physical address of access.

PRIV bool

Is privileged access.

RESPONSE enum

Whether the transaction was successful, or an error occurred.

USER_FLAGS unsigned int

Core specific additional signals.

PRELOAD_DATA

Data preload from PLD instruction. Fields:

VADDR unsigned int

Virtual address of the data that should be preloaded.

PRELOAD_INST

Instruction preload from PLI instruction. Fields:

VADDR unsigned int

Virtual address of the instruction that should be preloaded.

PRE_CORE_LOAD

Trace just before a core load. Fields:

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

PRE_CORE_STORE

Trace just before a core store. Fields:

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

PSTATE_PACTIVE_OPMODE_TRANSITION

Trace pactive and Opmode mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Operating mode after the transition.

OLD_STATE enum

Operating mode before the transition.

PSTATE_PACTIVE_OPMODE_TRANSITION_DENIED

Trace the denied pactive and Opmode mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Operating mode after the transition.

OLD_STATE enum

Operating mode before the transition.

REASON string

Reason for the transition being denied.

PSTATE_PACTIVE_POWER_TRANSITION

Trace pactive and power mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Power mode after the transition.

OLD_STATE enum

Power mode before the transition.

PSTATE_PACTIVE_POWER_TRANSITION_DENIED

Trace the denied pactive and power mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Power mode after the transition.

OLD_STATE enum

Power mode before the transition.

REASON string

Reason for the transition being denied.

PSTATE_PACTIVE_THREAD_OPMODE_TRANSITION

Trace pactive and Opmode mode transitions for threads on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Operating mode after the transition.

OLD_STATE enum

Operating mode before the transition.

QUANTUM_START

Called at the start of every quantum.

RUN_STATE

Run state transition. Fields:

INST_COUNT unsigned int

Ticks count at point of transition.

NEW enum

New run state.

OLD enum

Old run state.

SEMIHOSTING_CALL

Call of a semihost function occurred. Fields:

PC unsigned int

The program counter after the semihosting call.

SEMIHOSTING_PRECALL

About to call semihost. Fields:

EL2 bool

Use EL2 translation regime.

NS bool

Is Non-Secure.

REG_WIDTH unsigned int

The current register width in bytes.

SIGNAL

External signal state change. Fields:

SIGNAL enum

Signal that changed.

STATE bool

Signal asserted state.

START_COMPILE

Compilation started. Fields:

VADDR unsigned int

Instruction where compilation begins.

STG_MULTIPLE_MICRO_OPS_ADDRESS_RANGE

Address range for MTE stores with multiple uops. Fields:

ADDR_END unsigned int

Final address.

ADDR_START unsigned int

Starting address.

STORE_MULTIPLE_REGS_ADDRESS_RANGE

Address range for stores with multiple registers. Fields:

ADDR_END unsigned int

Final address.

ADDR_START unsigned int

Starting address.

SUBOPTIMAL_LDST_RETIRED

suboptimal load/store. Fields:

ACCESS_SIZE unsigned int

Log2 of the access width in bytes used for alignment checking (1: half-word, 2: word ...).

OPTIMAL_ALIGN_SELECTOR unsigned int

Selector to decide the optimal alignment size (1: access width, 2: 4-bytes, 3: 8-bytes, ..., 12: 4-kbytes).

VADDR unsigned int

The virtual address of the access.

SYNC

Called for every synchronization, that is every quantum break. Fields:

Copyright © 2023–2025 Arm Limited (or its affiliates). All rights reserved.

Non-Confidential

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The instruction count of this CPU.

LOCAL_QUANTUM unsigned int

The local quantum of this CPU.

LOCAL_TIME unsigned int

The local time of this CPU.

SYNC_REASON enum

Reason for sync.

SYSCALL

System call instruction executed. Fields:

IMM unsigned int

Immediate value of the system call instruction.

TYPE enum

System call type.

VADDR unsigned int

Instruction that caused the system call.

SYSREG_READ

Trace reads from system registers. Fields:

REG_NAME enum

Register number.

VALUE unsigned int

The value read.

SYSREG_UPDATE128

Triggers when the system updates a register. Fields:

REG enum

Register number.

UNKNOWN unsigned int

Bits of the register which became unknown.

VALUE unsigned int

Value written to the register.

SYSREG_UPDATE32

Triggers when the system updates a register. Fields:

REG enum

Register number.

UNKNOWN unsigned int

Bits of the register which became unknown.

VALUE unsigned int

Value written to the register.

SYSREG_UPDATE64

Triggers when the system updates a register. Fields:

REG enum

Register number.

UNKNOWN unsigned int

Bits of the register which became unknown.

VALUE unsigned int

Value written to the register.

SYSREG_WRITE

Trace write to system registers. Fields:

REG_ACCESS enum

Register access status.

REG_NAME enum

Register number.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

Value written to the register.

UNALIGNED_LDST_RETIRED

Processor unaligned load/store. Fields:

ACCESS_SIZE unsigned int

Log2 of the access width in bytes used for alignment checking (1=>half-word, 2->word ...).

NUMBER_OF_BEATS unsigned int

Number of accesses (beats) in this burst (Total size in bytes is ACCESS_SIZE * NUMBER_OF_BEATS).

VADDR unsigned int

The virtual address of the access.

WAYPOINT

Signals end of basic block execution. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ISET enum

Origin instruction set.

IS_COND bool

Indicates if this is a conditional waypoint.

PC unsigned int

Origin address (or 0 if unavailable).

TAKEN bool

Indicates if this waypoint was taken.

TARGET unsigned int

Destination address.

TARGET_ISET enum

Destination instruction set.

WFE_END

WFE ended. Fields:

INST_COUNT unsigned int

Ticks count when leaving WFE.

WFE_EVENT_REGISTER

WFE event register status: set/clear, reason. Fields:

INST_COUNT unsigned int

Ticks count.

REASON enum

Reason for set/clear. Only REASON==0 (Cleared by WFE) clears the bit, all other reasons set it.

WFE_IGNORED

WFE ignored. Fields:

EVENT bool

This WFE was ignored because the event register was set.

INST_COUNT unsigned int

Ticks count when ignoring WFE.

TRAPPED bool

This WFE was trapped.

WFE_START

WFE entered. Fields:

INST_COUNT unsigned int

Ticks count when entering WFE.

WFI_END

WFI ended. Fields:

INST_COUNT **unsigned int**

Ticks count when leaving WFI.

WFI_IGNORED

WFI ignored. Fields:

DISABLED **bool**

This WFI was ignored because WFI is disabled.

INST_COUNT **unsigned int**

Ticks count when ignoring WFI.

REASON **enum**

specifies reason why WFI trace was ignored.

TRAPPED **bool**

This WFI was trapped.

WFI_START

WFI entered. Fields:

INST_COUNT **unsigned int**

Ticks count when entering WFI.

WFI_WAKEUP

WFI wakeup. Fields:

INST_COUNT **unsigned int**

Ticks count when WFI wakeup occurred.

REASON **enum**

Reason for wakeup.

XPSR

Changes to the xPSR register. Fields:

CORE_NUM **unsigned int**

Core number in a multi processor.

OLD_VALUE **unsigned int**

The old xPSR value.

UNKNOWN **unsigned int**

Bits within the register that have unknown value.

VALUE **unsigned int**

The new xPSR value.

2.42 ARM_Cortex-M33

This section describes the trace sources.

ASYNC_MEMORY_FAULT

Context ID Register write. Fields:

FAULT unsigned int

Fault status present in ESR.ISS encoding for Serror in ARCH64 and present in DFSR encoding for external abort format in ARCH32.

PADDR unsigned int

Physical Address (or 0 if unavailable).

ATOMIC_END_ACCESS

Bus trace access for atomics. Fields:

ACCESS_FAIL bool

Memory access failed.

ADDR unsigned int

The virtual address of the access.

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

COMPARE_VALUE unsigned int

Compare value for CAS.

LOAD_VALUE unsigned int

Loaded values.

NSDESC unsigned int

The security state of the access.

OPERAND_VALUE unsigned int

Operation's operand.

OPERATION enum

Operation type.

PADDR unsigned int

The physical address of the access.

PRIV bool

Is this a privileged access?.

ATOMIC_START_ACCESS

Bus trace access for atomics. Fields:

ADDR unsigned int

The virtual address of the access.

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

COMPARE_VALUE unsigned int

Compare value for CAS.

NSDESC unsigned int

The security state of the access.

OPERAND_VALUE unsigned int

Operation's operand.

OPERATION enum

Operation type.

PADDR unsigned int

The physical address of the access.

PRIV bool

Is this a privileged access?.

ArchMsg.Error.exit_code

DISPLAY exit code: %{EXITCODE} component kind: %{KIND} reason: %{REASON}. Fields:

EXITCODE signed int

Exit code to be returned.

KIND string

Component kind that invoked the exit code trace.

REASON string

Reason why the exit code trace was invoked (optional).

ArchMsg.Warning.Unpredictably Indexed PM Event Register

TAGS UNPREDICTABLE DISPLAY Event %{IsDirect:(Selector value|Register)} %{SEL} is >= %{N}. Fields:

IsDirect unsigned int

Direct Access rather than indirect.

N unsigned int

Number of Accessible Registers.

SEL unsigned int

Selector.

ArchMsg.Warning.dap_csw_bad_size

A write to CM3DAP CSW has an invalid size field. DISPLAY Write %{DATA} to CM3DAP CSW has invalid size field. Fields:

DATA unsigned int

bits[2:0] are size.

ArchMsg.Warning.dcimvac_matches_watchpoint

TAGS **UNPREDICTABLE** DISPLAY Watchpoints matching an AArch32 DCIMVA are implementation defined.

ArchMsg.Warning.decode_fieldmismatch

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with different fields %{FIELD1} and %{FIELD2}. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_frachitsnegative

TAGS **UNPREDICTABLE** DISPLAY VCVT Instruction is unpredictable with a negative number of fraction bits.

ArchMsg.Warning.decode_initblock

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable in an IT block.

ArchMsg.Warning.decode_initblocknotlast

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable in an IT block when not the last.

ArchMsg.Warning.decode_invalid_condition

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable when the condition is not AL.

ArchMsg.Warning.decode_invalidfieldcombination

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with the values in %{FIELD1} and %{FIELD2}. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_invalidvalue

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with the value in %{FIELD}. Fields:

FIELD enum

Field name.

ArchMsg.Warning.decode_mem_hint_unallocated

TAGS **UNPREDICTABLE** DISPLAY There is no memory hint allocated to this bit pattern.

ArchMsg.Warning.decode_registermatch

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with %{FIELD1} and %{FIELD2} as the same register. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_registermismatch

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with %{FIELD1} and %{FIELD2} as different registers. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_registeroutofrange

TAGS **UNPREDICTABLE** DISPLAY Floating-point instruction is unpredictable indexing beyond the end of the register bank.

ArchMsg.Warning.decode_sbzsbo

TAGS **UNPREDICTABLE** DISPLAY Reserved bits in the instruction are not canonical.

ArchMsg.Warning.decode_transactiontoobig

TAGS **UNPREDICTABLE** DISPLAY Attempt to perform a memory transaction of more than 128 bytes.

ArchMsg.Warning.decode_unpred_other

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable.

ArchMsg.Warning.decode_unpreduseofpc

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable when %{FIELD} is PC. Fields:

FIELD enum

Field name.

ArchMsg.Warning.decode_unpreduseofr13

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable when %{FIELD} is R13. Fields:

FIELD enum

Field name.

ArchMsg.Warning.decode_unpreduseofr13inreglist

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with R13 in register-list.

ArchMsg.Warning.decode_useofd16orhigher

TAGS **UNPREDICTABLE** DISPLAY Attempt to use register D16 or higher where it is not allowed.

ArchMsg.Warning.decode_writebackandbaseinlist

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with write back when the base register is in the transfer list.

ArchMsg.Warning.decode_zeroregistersinlist

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable when no registers are to be transferred.

ArchMsg.Warning.recursive_branch

An instruction is performing a branch that targets the same instruction. If this was intended then a WFI might be more effective DISPLAY recursive branch detected at PC=%{PC}. PRIMARY KEY PC. Fields:

PC unsigned int

Address of instruction causing the branch.

ArchMsg.Warning.recursive_exception

An instruction has generated an exception that targets the same instruction. DISPLAY recursive %{TYPE} exception detected at PC=%{PC} (CPSR=%{CPSR}). PRIMARY KEY PC. Fields:

CPSR unsigned int

Processor state.

PC unsigned int

Address of instruction causing the exception.

TYPE enum

Type of exception.

ArchMsg.Warning.reentrant_vector_catch

TAGS **UNPREDICTABLE** DISPLAY Vector catch debug event on Prefetch or Data abort vector %{ADDR}. Fields:

ADDR unsigned int

vector address.

ArchMsg.Warning.reserved_it_state

TAGS **UNPREDICTABLE** DISPLAY Execution with reserved IT state %{ITSTATE}. Fields:

ITSTATE unsigned int

reserved IT state.

ArchMsg.Warning.secure_vector_fetch_from_nonsecure

DISPLAY Secure vector table is being fetched from Non-secure memory.

ArchMsg.Warning.sg_in_it_block

Arm recommends that software does not place SG (security gate instruction) inside an IT instruction block.

ArchMsg.Warning.unpred_FAR_on_LRCPC3_non_atomic_mem_access

TAGS **UNPREDICTABLE** Unpredictable FAR/FSTATUS on LRCPC3 pair instructions resulting in two separate memory accesses
 DISPLAY Unpredictable FAR on LRCPC3 non atomic memory access at %{ADDR1} with FSTATUS %{FSTATUS1} or %{ADDR2} with FSTATUS %{FSTATUS2}. Fields:

ADDR1 unsigned int

Possible FAR address 1.

ADDR2 unsigned int

Possible FAR address 2.

FSTATUS1 unsigned int

Possible fault status value 1.

FSTATUS2 unsigned int

Possible fault status value 2.

ArchMsg.Warning.unpred_async_tag_check_fail_on_sve_speculative_load_do_update_ffr

Unpredictable set to FALSE of the FFR element associated to SVE speculative load with an async tag check fail TAGS **UNPREDICTABLE**.

ArchMsg.Warning.unpredictable_a32_breakpoint

TAGS **UNPREDICTABLE** DISPLAY Unpredictable breakpoint on a A32 instruction at %{ADDR1} Offset %{OFFSET} BAS %{BAS} for a %{IS_ADDRESS_MISMATCH_BREAKPOINT:(Mismatch|Match)} Breakpoint number %{N:d} Breakpoint address %{ADDR2}. Fields:

ADDR1 unsigned int

Address.

ADDR2 unsigned int

Breakpoint address.

BAS unsigned int

BAS field.

IS_ADDRESS_MISMATCH_BREAKPOINT bool

Is a Mismatch breakpoint, not a Match breakpoint.

N unsigned int

Breakpoint number.

OFFSET unsigned int

Always has a value of zero.

ArchMsg.Warning.unpredictable_banked_register_access

TAGS **UNPREDICTABLE** DISPLAY Access to banked register (R=%{R:d} SYSm=%{SYSM:2x}) is unpredictable. Fields:

R unsigned int

R bit.

SYSM unsigned int
SYSm.

ArchMsg.Warning.unpredictable_ccfail_undef

TAGS **UNPREDICTABLE** DISPLAY An instruction %{OPCODE} at VA=%{VADDR} failed its condition codes check but would otherwise have caused a non-data-dependent trap or undefined exception. Behaviour is implementation-defined choice of exception or nop. Fields:

OPCODE unsigned int
opcode of the instruction.

VADDR unsigned int
Virtual address of the instruction being executed.

ArchMsg.Warning.unpredictable_ici_bits

TAGS **UNPREDICTABLE** DISPLAY ICI bits do not apply or are architecturally unpredictable for instruction.

ArchMsg.Warning.unpredictable_t32_breakpoint

TAGS **UNPREDICTABLE** DISPLAY Unpredictable breakpoint on a T32 instruction at %{ADDR1} Offset %{OFFSET} BAS %{BAS} for a %{IS_ADDRESS_MISMATCH_BREAKPOINT:(Mismatch|Match)} Breakpoint number %{N:d} Breakpoint address %{ADDR2}. Fields:

ADDR1 unsigned int
Address.

ADDR2 unsigned int
Breakpoint address.

BAS unsigned int
BAS field.

IS_ADDRESS_MISMATCH_BREAKPOINT bool
Is a Mismatch breakpoint, not a Match breakpoint.

N unsigned int
Breakpoint number.

OFFSET unsigned int
Indicates which halfword of a 32-bit Thumb instruction the breakpoint is set on. 0 for the first halfword or if the instruction is 16 bits wide, or 1 for the second halfword.

ArchMsg.Warning.unpredictable_t32_vector_catch

TAGS **UNPREDICTABLE** DISPLAY Unpredictable Vector catch exception on a T32 instruction at %{ADDR} Offset %{OFFSET}. Fields:

ADDR unsigned int
Address.

OFFSET unsigned int

Indicates which halfword of a 32-bit Thumb instruction the vector catch exception is set on. 0 for the first halfword or if the instruction is 16 bits wide, or 1 for the second halfword.

ArchMsg.Warning.unpredictable_watchpoint_far

TAGS **UNPREDICTABLE** DISPLAY The value for FAR/EDWAR for hitting this watchpoint may be anywhere in the range %{LOWER_BOUND} to %{UPPER_BOUND}. Fields:

LOWER_BOUND unsigned int

the lowest address accessed by the instruction that triggered the watchpoint.

UPPER_BOUND unsigned int

the highest watchpointed address accessed by that instruction.

ArchMsg.Warning.warning_AdvSIMDExpandImmUnexpectedZero

TAGS **UNPREDICTABLE** DISPLAY AdvSIMDExpandImm may treat this immediate value as **UNPREDICTABLE**.

ArchMsg.Warning.warning_ConditionalSMC

TAGS **UNPREDICTABLE** DISPLAY SMC instruction has **UNPREDICTABLE** effects when conditional and when combined with traps.

ArchMsg.Warning.warning_access_crosses_page_boundary

TAGS **UNPREDICTABLE** DISPLAY Access crosses page boundary.

ArchMsg.Warning.warning_access_crosses_page_boundary_has_fault

TAGS **UNPREDICTABLE** DISPLAY Access that crosses page boundary has fault(s).

ArchMsg.Warning.warning_access_crosses_page_boundary_has_fault_on_both_pages

TAGS **UNPREDICTABLE** DISPLAY Access that crosses page boundary has a fault on each page. Fields:

ADDR1 unsigned int

FAR address on first page.

ADDR2 unsigned int

FAR address on second page.

FSTATUS1 unsigned int

Fault status on first page.

FSTATUS2 unsigned int

Fault status on second page.

ArchMsg.Warning.warning_access_crosses_page_boundary_spanning_different_memory

TAGS **UNPREDICTABLE** DISPLAY Access at address %{ADDR} crosses page from MEM_TYPE %{MEMTYPE_PAGE1} to %{MEMTYPE_PAGE2}. Fields:

ADDR unsigned int

address of access crossing page.

MEMTYPE_PAGE1 signed int

First Page's memory type (0-Normal, 1-Non-Normal).

MEMTYPE_PAGE2 signed int

Second Page's memory type (0-Normal, 1-Non-Normal).

ArchMsg.Warning.warning_access_wraps_around_memory

TAGS UNPREDICTABLE DISPLAY Access wraps around memory in %{WIDTH:d}bit mode.
Fields:

WIDTH unsigned int

Address width.

ArchMsg.Warning.warning_bcr_linking_status

TAGS UNPREDICTABLE DISPLAY DBGBCR.{BT2,BT} combination gives unpredictable behaviour.

ArchMsg.Warning.warning_bcr_mask_reserved

TAGS UNPREDICTABLE DISPLAY DBGBCR_ELn.MASK is set to a reserved value.

ArchMsg.Warning.warning_bvr_masked_bit_not_zero

TAGS UNPREDICTABLE DISPLAY DBGBVRn_EL1.VA contains a bit masked by DBGBCRn_EL1.MASK that is not zero.

ArchMsg.Warning.warning_ccsidr_unimplemented_level

TAGS UNPREDICTABLE DISPLAY CCSIDR read while CSSELR %{CSSELR:x} points to an unimplemented cache level. Fields:

CSSELR unsigned int

current effective value of CSSELR.

ArchMsg.Warning.warning_change_to_ns_when_tge_set

TAGS UNPREDICTABLE DISPLAY Attempting to change NS to 1 when HCR.TGE = 1.

ArchMsg.Warning.warning_contiguous_bit_check_abort

TAGS UNPREDICTABLE DISPLAY An abort occurred whilst attempting to check TLB entry is contiguous at %{ENTRY_ADDR}. Fields:

ENTRY_ADDR unsigned int

address of conflicting TLB entry.

ArchMsg.Warning.warning_contiguous_bit_error

TAGS UNPREDICTABLE DISPLAY TLB entry at %{ENTRY_ADDR} was not contiguous with entry at %{CONTIG_ADDR} Contents are %{ENTRY_DATA} but expected %{CONTIG_DATA}.
Fields:

CONTIG_ADDR unsigned int

address of the TLB first read, with which this entry is expected to be contiguous.

CONTIG_DATA unsigned int

expected contents based on the entry at CONTIG_ADDR.

ENTRY_ADDR unsigned int

address of conflicting TLB entry.

ENTRY_DATA unsigned int

contents of conflicting TLB entry.

ArchMsg.Warning.warning_contiguous_bit_error_D128

TAGS **UNPREDICTABLE** DISPLAY TLB entry at %{ENTRY_ADDR} was not contiguous with entry at %{CONTIG_ADDR} Contents are %{ENTRY_DATA_HI} %{ENTRY_DATA_LO} but expected %{CONTIG_DATA}. Fields:

CONTIG_ADDR unsigned int

address of the TLB first read, with which this entry is expected to be contiguous.

CONTIG_DATA unsigned int

expected contents based on the entry at CONTIG_ADDR.

ENTRY_ADDR unsigned int

address of conflicting TLB entry.

ENTRY_DATA_HI unsigned int

Higher bit contents of conflicting TLB entry.

ENTRY_DATA_LO unsigned int

Lower bit contents of conflicting TLB entry.

ArchMsg.Warning.warning_cp10_cp11_mismatch

TAGS **UNPREDICTABLE** DISPLAY Unpredictable configuration of CP10 and CP11 controls at EL %{EL:d}. Fields:

EL unsigned int

EL owning the control.

ArchMsg.Warning.warning_cp10_cp11_reserved_value

TAGS **UNPREDICTABLE** DISPLAY Unpredictable configuration of CPACR CP10 (%{CP10}) or CP11 (%{CP11}). Fields:

CP10 unsigned int

CP10 access.

CP11 unsigned int

CP11 access.

ArchMsg.Warning.warning_csselr_level_out_of_range

TAGS **UNPREDICTABLE** DISPLAY CSSELR.Level out of range, written value %{LEVEL} with only %{IMPLEMENTED:d} levels implemented. Fields:

IMPLEMENTED unsigned int

number of cache levels implemented.

LEVEL unsigned int

written value of CSSELR.Level.

TYPE enum

The type of cache selected by CSSELR.InD and TnD.

ArchMsg.Warning.warning_debug_flow_control_bits_not_obeyed

TAGS **UNPREDICTABLE** DISPLAY Flow control bits for %{REGISTER} were not obeyed, causing %{ERROR}. Fields:

ERROR enum

type of error resulting.

REGISTER enum

register under consideration.

ArchMsg.Warning.warning_debug_register_access_during_reset

TAGS **UNPREDICTABLE** DISPLAY Unpredictable %{IS_WRITE:(read|write)} access to debug register offset %{OFFSET} during reset. Fields:

IS_WRITE unsigned int

Write Not Read.

OFFSET unsigned int

Register Offset.

ArchMsg.Warning.warning_decode_cps_inconsistent_fields

TAGS **UNPREDICTABLE** DISPLAY The mode setting fields in the CPS instruction are inconsistent.

ArchMsg.Warning.warning_decode_invalid_state

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable in current exception-level/security state.

ArchMsg.Warning.warning_default_cacheable_mmu_on

TAGS **UNPREDICTABLE** DISPLAY Default cacheable is enabled (HCR.DC = 1) while MMU is enabled.

ArchMsg.Warning.warning_default_cacheable_vmmu_off

TAGS **UNPREDICTABLE** DISPLAY Default cacheable is enabled (HCR.DC = 1) while HCR.VM = 0.

ArchMsg.Warning.warning_deprecated_wvr_bit_2_set

TAGS **UNPREDICTABLE** DISPLAY DBGWVR%{N:d}_EL1=%{WVR:h} has bit 2 set, deprecated in ARMv8. Fields:

N unsigned int

index.

WVR unsigned int

DBGWVR.

ArchMsg.Warning.warning_eret_while_software_step_active_pending

Missing ISB between setting MDSCR_EL1.SS and ERET TAGS **UNPREDICTABLE**.

ArchMsg.Warning.warning_exclusive_to_non_normal

TAGS **UNPREDICTABLE** DISPLAY Exclusive Access to Non-Normal Memory %{ADDR}. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_exclusive_to_non_writeback

TAGS **UNPREDICTABLE** DISPLAY Exclusive Access to Non-Writeback-Cacheable Memory %{ADDR}. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_execute_from_device_memory

TAGS **UNPREDICTABLE** DISPLAY An attempt was made to execute from Device memory.

ArchMsg.Warning.warning_illegal_cpsr_mode

TAGS **UNPREDICTABLE** DISPLAY Writing an illegal or unimplement mode (%{NEW_MODE:h}) to CPSR was unpredictable. Fields:

NEW_MODE unsigned int

New value of CPSR.M.

ArchMsg.Warning.warning_illegal_srs_mode

TAGS **UNPREDICTABLE** DISPLAY Illegal or **UNPREDICTABLE** mode (%{MODE:h}) used for SRS instruction. Fields:

MODE unsigned int

mode for Banked SP.

ArchMsg.Warning.warning_implementation_defined_read_debug_register_in_SCS

Read Access to Debug registers from the processor is **IMPLEMENTATION DEFINED** TAGS **IMP_DEF**. Fields:

PPB_OFFSET unsigned int

debug register offset.

ArchMsg.Warning.warning_implementation_defined_sequential_security_transitions_supported

The behavior of sequential instruction fetches that cross from non-secure to secure memory and contain an SG instruction is **CONSTRAINED UNPREDICTABLE** TAGS **UNPREDICTABLE**.

ArchMsg.Warning.warning_implementation_defined_stack_limit_check_supported

It is **IMPLEMENTATION DEFINED** whether stack pointer limit checking is performed for this instruction TAGS **IMP_DEF**. Fields:

ADDRESS unsigned int

Address of instruction.

OPCODE unsigned int

instruction opcode.

ArchMsg.Warning.warning_implementation_defined_taggednonsharedmemorytype

TAGS IMP_DEF DISPLAY Memory attribute has Tagged, non shared memory type; it is **IMPLEMENTATION DEFINED** whether the memory location is treated as Tagged or non-Tagged.

ArchMsg.Warning.warning_implementation_defined_write_debug_register_in_SCS

Write Access to Debug registers from the processor is **IMPLEMENTATION DEFINED** TAGS IMP_DEF. Fields:

DATA unsigned int

data attempted to be written.

PPB_OFFSET unsigned int

debug register offset.

ArchMsg.Warning.warning_invalid_tcr_granule

TAGS **UNPREDICTABLE** DISPLAY TCR.TG%{TG_ID:(0|1)} bits have been set to a granule size %{REQUEST:d}K not implemented - using %{SUBSTITUTE:d}K. Fields:

REQUEST signed int

page size requested (or 0 for reserved).

SUBSTITUTE signed int

best guess available page size.

TG_ID bool

bits TG0 or TG1.

ArchMsg.Warning.warning_load_multiple_user_registers_from_user_mode

TAGS **UNPREDICTABLE** DISPLAY An LDM(user registers) instruction executed from user mode.

ArchMsg.Warning.warning_mte_in_mem_attr_not_supported

TAGS **UNPREDICTABLE** DISPLAY Memory attribute %{ATTR} has Tagged memory type but Armv8.5-MTE is not implemented. Fields:

ATTR unsigned int

Memory attribute.

ArchMsg.Warning.warning_reserved_breakpoint_state_match

TAGS **UNPREDICTABLE** DISPLAY %{IS_BREAKPOINT:(Watchpoint|Breakpoint)} programmed with a reserved combination of HMC, SSC and %{IS_BREAKPOINT:(PAC|PMC)}. Fields:

IS_BREAKPOINT unsigned int

Is a breakpoint, not a watchpoint.

ArchMsg.Warning.warning_shareability

TAGS **UNPREDICTABLE** DISPLAY Unpredictable: combination of the 1st and 2nd stages of translation is Normal Inner Non-Cacheable, Outer Non-Cacheable.

ArchMsg.Warning.warning_software_step_set_while_enabled

TAGS **UNPREDICTABLE** DISPLAY MDSCR_EL1.SS set to 1 while software step debug exceptions are enabled.

ArchMsg.Warning.warning_thumb_instruction_wraps_around_memory

TAGS UNPREDICTABLE DISPLAY Thumb instruction wraps around memory.

ArchMsg.Warning.warning_ttbr_sbz_bits_are_not_zero

TAGS UNPREDICTABLE DISPLAY %{TTBR}[%{MSB:d}:0] should be zero, but are not. Fields:

MSB unsigned int

x-1.

TTBR enum

which TTBR register.

ArchMsg.Warning.warning_unaligned_address_dbgdttrrx_write

TAGS UNPREDICTABLE DISPLAY Unaligned address %{ADDR} in X0/R0 when DBGDTTRRX written in memory access mode. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unaligned_address_dbgdttrtx_read

TAGS UNPREDICTABLE DISPLAY Unaligned address %{ADDR} in X0/R0 when DBGDTTRTX read in memory access mode. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unaligned_to_device

TAGS UNPREDICTABLE DISPLAY Unaligned Access to Device Memory %{ADDR}. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unaligned_to_strongly_ordered

TAGS UNPREDICTABLE DISPLAY Unaligned Access to Strongly Ordered Memory %{ADDR}. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unknown_sau_rnr

TAGS UNPREDICTABLE DISPLAY SAU_RNR was set to an unsupported value.

ArchMsg.Warning.warning_unpredictable_AIRCR_PRIS_and_BFHFNMINs

TAGS UNPREDICTABLE DISPLAY The effect of setting both AIRCR.BFHFNMINs and AIRCR.PRIS to 1 is UNPREDICTABLE.

ArchMsg.Warning.warning_unpredictable_AIRCR_VECTCLRACTIVE_when_not_in_debug

TAGS UNPREDICTABLE DISPLAY The effect of writing a 1 to AIRCR.VECTCLRACTIVE if the processor is not halted in Debug state is UNPREDICTABLE.

ArchMsg.Warning.warning_unpredictable_AIRCR_VECTRESET_and_SYSRESETREQ

TAGS **UNPREDICTABLE** DISPLAY When the processor is halted in Debug state, if a write to the register writes a 1 to both VECTRESET and SYSRESETREQ, the behavior is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_AIRCR_VECTRESET_when_not_in_debug

TAGS **UNPREDICTABLE** DISPLAY The effect of writing a 1 to AIRCR.VECTRESET if the processor is not halted in Debug state is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_AIRCR_incorrect_VKEY

TAGS **UNPREDICTABLE** DISPLAY The value 0x05FA must be written to AIRCR.VKEY, otherwise the register write is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_EXC_RETURN_Reserved_Bit

EXC_RETURN[23:7] are reserved with the special condition that all bits should be written as one. Values other than all 1s are **UNPREDICTABLE**. TAGS **UNPREDICTABLE**. Fields:

EXC_RETURN unsigned int

Special PC value.

SBO_MASK unsigned int

Which bits Should Be One to not UNPRED.

ArchMsg.Warning.warning_unpredictable_PMEVTYPEPn_ELO_VS_reserved_value

TAGS **UNPREDICTABLE** DISPLAY Unpredictable reserved value of PMEVTYPEn<n>_ELO.VS.

ArchMsg.Warning.warning_unpredictable_change_to_DEMCR_MON_STEP_at_insufficient_priority

TAGS **UNPREDICTABLE** DISPLAY The effect of changing DEMCR.MON_STEP at an execution priority that is lower than the priority of the DebugMonitor exception is **CONSTRAINED UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_change_to_priority_of_active_exception

TAGS **UNPREDICTABLE** DISPLAY Changing the priority of an active exception is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_clear_lspact

TAGS **UNPREDICTABLE** DISPLAY **UNPREDICTABLE** state on context switch. Fields:

DRVEXC_TAKEN unsigned int

Derived exception taken, not pended.

REACHED_LAST_STORE unsigned int

Only the last faulted.

ArchMsg.Warning.warning_unpredictable_dbg_exit_unaligned_dlr

TAGS **UNPREDICTABLE** DISPLAY Debug exit to AArch32 with DLR[0] set to 1. PC[0] can be set to 0 or to DLR[0].

ArchMsg.Warning.warning_unpredictable_exception_catch

TAGS **UNPREDICTABLE** DISPLAY Generation of an exception catch event is unpredictable.

ArchMsg.Warning.warning_unpredictable_exception_return_inconsistent_state

TAGS **UNPREDICTABLE** DISPLAY state on exception return is unpredictable.

ArchMsg.Warning.warning_unpredictable_exception_return_instruction

TAGS **UNPREDICTABLE** DISPLAY exception return instruction is unpredictable.

ArchMsg.Warning.warning_unpredictable_in_debug_state

TAGS **UNPREDICTABLE** DISPLAY Instruction is **UNPREDICTABLE** when executed in debug state.

ArchMsg.Warning.warning_unpredictable_mair_encoding

TAGS **UNPREDICTABLE** DISPLAY Memory attribute %{ATTR} has invalid MAIR encoding; this may lead to unpredictable results. Fields:

ATTR unsigned int

Memory attribute.

ArchMsg.Warning.warning_unpredictable_pmu_counter_access

TAGS **UNPREDICTABLE** DISPLAY Access to PMU counters from non-secure EL0 or EL1 is **UNPREDICTABLE** with MDCR_EL2.HPMN set to 0.

ArchMsg.Warning.warning_unpredictable_prioritization_breakpoint_match_vector_catch

TAGS **UNPREDICTABLE** DISPLAY Two events with the same priority occurred at the same instruction: Address Matching Vector Catch debug event, and Breakpoint debug event. It is constrained unpredictable which is taken.

ArchMsg.Warning.warning_unpredictable_stack_selection

TAGS **UNPREDICTABLE** DISPLAY **UNPREDICTABLE** state on context switch.

ArchMsg.Warning.warning_unpredictable_tsize_out_of_range

TAGS **UNPREDICTABLE** DISPLAY %{TCR}.T%{TTBR:d}SZ (%{TSIZE:d} bits) is out of range. Must be between 25 and %{TSIZE_MAX:d}. Fields:

TCR enum

Which TCR.

TSIZE signed int

Value of TSize.

TSIZE_MAX signed int

The maximum allowed value.

TTBR unsigned int

Which TTBR region.

ArchMsg.Warning.warning_unpredictable_unaligned_pc_as_base_register

TAGS **UNPREDICTABLE** DISPLAY The PC value must be word-aligned, otherwise the behavior of the instruction is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_unaligned_pop_stack

TAGS **UNPREDICTABLE** DISPLAY **CONSTRAINED UNPREDICTABLE** unaligned stack frame %{FRAME_ADDR:x} in exception-return. Fields:

FRAME_ADDR unsigned int

unaligned stack frame address.

ArchMsg.Warning.warning_unpredictable_unaligned_to_non_writeback

TAGS **UNPREDICTABLE** DISPLAY Unaligned Access to Non-Writeback-Cacheable Memory.

ArchMsg.Warning.warning_unpredictable_vmsa_memattrib

TAGS **UNPREDICTABLE** DISPLAY Unpredictable vmsa memory attribute with texcb value programmed to %{TEXCB} when tex remap is %{TEX_REMAP}. Fields:

TEXCB unsigned int

texcb value.

TEX_REMAP bool

use tex remap.

ArchMsg.Warning.warning_unpredictable_vtcr_t0sz_sl0

TAGS **UNPREDICTABLE** DISPLAY The combination of VTCR.TOSZ=%{TOSZ:d} and VTCR.SL0=%{SL0:d} is unpredictable. Fields:

SL0 signed int

VTCR.SL0, starting level for stage 2 translation table walks.

TOSZ signed int

VTCR.TOSZ, required input address range.

ArchMsg.Warning.warning_unpredictable_write_DHCSR

TAGS **UNPREDICTABLE** DISPLAY **UNPREDICTABLE** set of bit-changes in DHCSR. Fields:

NEW unsigned int

data attempted to be written.

OLD unsigned int

previous value.

ArchMsg.Warning.warning_unpredictable_write_SHCSR

TAGS **UNPREDICTABLE** DISPLAY **UNPREDICTABLE** setting a pending bit to 1 for an exception with priority >= execution priority.

ArchMsg.Warning.warning_unsupported_access_to_memory_mapped_register

TAGS **UNPREDICTABLE** DISPLAY Access to memory mapped register is unsupported.

ArchMsg.Warning.warning_user_jmcr_access

TAGS **UNPREDICTABLE** DISPLAY User %{WRITE:(write|read)} access to JMCR is unpredictable. Fields:

WRITE unsigned int

access is write.

ArchMsg.Warning.warning_wcr_mask_and_bas

TAGS **UNPREDICTABLE** DISPLAY DBGWCR_EL1.MASK is non-zero and BAS is not 0xff.

ArchMsg.Warning.warning_wcr_mask_reserved

TAGS **UNPREDICTABLE** DISPLAY DBGWCR_ELn.MASK is set to a reserved value.

ArchMsg.Warning.warning_wcr_non_configuous_bas

TAGS **UNPREDICTABLE** DISPLAY DBGWCR_EL1.BAS has non-contiguous set of ones.

ArchMsg.Warning.warning_wvr_masked_bit_not_zero

TAGS **UNPREDICTABLE** DISPLAY DBGWVRn_EL1.VA contains a bit masked by DBGWCRn_EL1.MASK that is not zero.

ArchMsg.warning_unpred_system_register_access

DISPLAY Unpredictable %{IS_WRITE:(read|write)} access to system register offset %{OFFSET}. Fields:

IS_WRITE unsigned int

Write Not Read.

OFFSET unsigned int

Register Offset.

BRANCH_MISPREDICT

Simulating branch mispredict. Fields:

PC unsigned int

Origin address (or 0 if unavailable).

BRA_DIR

Direct branches, to immediate address. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISSET enum

The instruction set of the branch instruction.

IS_COND bool

Indicates if this is a conditional branch.

IS_HINTED bool

Whether the conditional instruction is hinted.

IS_ISB bool

Whether the instruction is ISB.

IS_LINK bool

Indicates if this is a branch with link.

PC unsigned int

The address of the branch instruction.

TARGET_ISET enum

The instruction set after the branch.

TARGET_PC unsigned int

The address the instruction branches to.

BRA_INDIR

Indirect branches, perhaps to a register. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISET enum

The instruction set of the branch instruction.

IS_COND bool

Indicates if this is a conditional branch.

IS_HINTED bool

Whether the conditional instruction is hinted.

IS_ISB bool

Whether the instruction is ISB.

IS_LINK bool

Indicates if this is a branch with link.

PC unsigned int

The address of the branch instruction.

TARGET_ISET enum

The instruction set after the branch.

TARGET_PC unsigned int

The address the instruction branches to.

BRA_INDIR_CCFAIL

Indirect branches fail, perhaps to a register.

BRA_RET_CCFAIL

Branch return fail, perhaps to a register.

CCFAIL

Conditional instruction condition check fail. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CCFAIL_UNC

Unconditional instruction condition check fail. Fields:

COND *enum*

The condition of the conditional instruction.

CORE_NUM *unsigned int*

Core number in a multi processor.

PC *unsigned int*

The address of the conditional instruction.

CCPASS

Conditional instruction condition check pass. Fields:

COND *enum*

The condition of the conditional instruction.

CORE_NUM *unsigned int*

Core number in a multi processor.

PC *unsigned int*

The address of the conditional instruction.

CCPASS_UNC

Unconditional instruction condition check pass. Fields:

COND *enum*

The condition of the conditional instruction.

CORE_NUM *unsigned int*

Core number in a multi processor.

PC *unsigned int*

The address of the conditional instruction.

CODE_CACHE_MAINT

Code cache maintenance. Fields:

END_INCL *unsigned int*

Inclusive end address for VA or PA requests.

NS *bool*

Non-secure world for PA.

START *unsigned int*

Start address for VA or PA requests.

TYPE *enum*

Request type.

COMPILE_BLOCK_END

Last instruction of basic block translated. Fields:

VADDR unsigned int

Address of next instruction after this basic block.

COMPILE_BLOCK_START

First instruction of basic block translated. Fields:

VADDR unsigned int

Address of first instruction in basic block.

COMPILE_INST

ARM instruction compiled. Fields:

DISASS string

Disassembly of the instruction.

ISSET enum

The instruction set of this instruction.

ITSTATE unsigned int

The ITSTATE current for the instruction.

OPCODE unsigned int

The opcode of the instruction.

PC unsigned int

The address of the instruction.

SIZE unsigned int

The size of the instruction in bytes.

CONTEXTIDR

Context ID Register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

NS bool

Secure or nonsecure banked register is accessed.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The new value written.

CONTEXT_SYNC

Called for every context synchronization event. Fields:

ARCH_CSE bool

Architectural Context Synchronization Event.

PC unsigned int

The address of the synchronization instruction.

SYSREG_SYNC bool

System register synchronization event.

CORE_ENDIAN

Core BE8 Big-Endian state changed. Fields:

BE8 bool

Core BE8 Big-Endian state.

CORE_INFO

Static processor attributes. Only triggered by a call to DumpState(). Fields:

ARCH_PROFILE enum

The architecture profile of the core.

CLUSTER_ID unsigned int

The cluster ID of this processor.

CORE_NUM unsigned int

The number of this core in an MP processor.

DETAILED_CORE_FEATURES enum

Detailed core features as a bitfield. Enum values are bitmasks to extract the info from the field.

FPU_VERSION enum

The VFP version implemented by the core.

MEM_ARCH enum

The memory architecture of the core.

MIDR_EL1 unsigned int

MIDR_EL1 value.

NUM_CORES unsigned int

The number of cores in this MP processor.

QUANTUM_SIZE unsigned int

The default quantum size of the core (in ticks).

SECURITY_FEATURES bool

Does the core have security features?.

CORE_LOADS

Processor load accesses. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ACQREL enum

Is this an acquire/release.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DATA unsigned int

The data read or written.

ELEMENT_SIZE unsigned int

Width of each element.

ENDIANNESS enum

Byte ordering of this access.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

LOCK enum

Normal, exclusive or locked access.

MECID unsigned int

Memory Encryption Context ID, if applicable.

MECID2 unsigned int

Memory Encryption Context ID of the second page, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PAGE_SIZE unsigned int

Page size as $\log_2(\text{size in bytes})$.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK, 2=Exclusive Failed, 3=Unsupported.

SIZE unsigned int

Width of the access in bytes. Only required if DATA is not traced.

TRANS bool

Is this a translated access.

VADDR unsigned int

The virtual address of the access.

CORE_REGS

Changes of the core registers R0 to R14. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number, 0 to 14.

OLD_VALUE unsigned int

The old value overwritten.

PHYS_ID enum

The physical register accessed.

VALUE unsigned int

The new value written to the register.

CORE_STALLED_TRANSITION

Detect when a core is prevented from progressing PChannel states. Fields:

CURRENT_STATE enum

Current power mode of core.

REASON string

The reason a transition was stalled, and what must be done to unblock it.

TARGET_STATE enum

Desired power mode of core after transition.

CORE_STORES

Processor store accesses. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ACQREL enum

Is this an acquire/release.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DATA unsigned int

The data read or written.

ELEMENT_SIZE unsigned int

Width of each element.

ENDIANNESS enum

Byte ordering of this access.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

LOCK enum

Normal, exclusive or locked access.

MECID unsigned int

Memory Encryption Context ID, if applicable.

MECID2 unsigned int

Memory Encryption Context ID of the second page, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PAGE_SIZE unsigned int

Page size as $\log_2(\text{size in bytes})$.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK, 2=Exclusive Failed, 3=Unsupported.

SIZE unsigned int

Width of the access in bytes. Only required if DATA is not traced.

TRANS bool

Is this a translated access.

VADDR unsigned int

The virtual address of the access.

CRYPTO_SPEC

Every crypto instruction speculatively executed.

DATA_CACHE_SET_ALLOC_TAGS

Set allocation tags from DC GVA instruction. Fields:

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

NSDESC unsigned int

The security state of the access.

PADDR unsigned int

The physical address of the access.

PAS enum

The physical address space of the page.

RESPONSE enum

0=Aborted, 1=OK.

SIZE unsigned int

The size of memory in bytes.

TAG unsigned int

The value of the tag being set.

TAGGING_TYPE enum

Type of Memory tagging (Physical or Virtual).

TAG_VADDR unsigned int

The virtual address of the Virtual Tag if Virtual tagging is used; otherwise 0.

VADDR unsigned int

Virtual address used for setting allocation tags.

DATA_CACHE_ZERO

Zero and invalidate cache line from DC ZVA instruction. Fields:

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

NSDESC unsigned int

The security state of the access.

PADDR unsigned int

The physical address of the access.

PADDR_TAG unsigned int

If tag checked, the physical address tag for the access.

PAS enum

The physical address space of the page.

RESPONSE enum

0=Aborted, 1=OK.

SIZE unsigned int

The size of zero'd memory in bytes.

TAG_CHECKED bool

Is this access tag checked.

VADDR unsigned int

Virtual address of the zero'd cache line.

DEBUG_EVENT

Hardware debug support event. Fields:

EVENT enum

Description of event.

VALUE unsigned int

data value.

DMI_ALLOCATE

DMI allocated. Fields:

ACCESS_DIRECTION enum

READ / WRITE.

PADDR unsigned int

The physical address of the DMI.

PTR unsigned int

DMI range pointer.

SIDE enum

DSIDE / ISIDE.

SIZE unsigned int

The size of DMI.

VADDR unsigned int

The virtual address of the DMI..

DMI_HIT

DMI hit, access used DMI (only fire for small DMI regions i.e. less than a full page). Fields:

ACCESS_DIRECTION enum

READ / WRITE.

PADDR unsigned int

The physical address of the DMI.

SIDE enum

DSIDE / ISIDE.

SIZE unsigned int

The size of the DMI.

VADDR unsigned int

The virtual address of the DMI..

DMI_REVOKE

DMI revoked. Fields:

ACCESS_DIRECTION enum

READ / WRITE.

PADDR unsigned int

The physical address of the DMI.

SIDE enum

DSIDE / ISIDE.

SIZE unsigned int

The size of DMI.

VADDR unsigned int

The virtual address of the DMI..

DWT_MATCH

DWT comparator matches. Fields:

NUM unsigned int

DWT comparator number.

TYPE enum

DWT comparator configuration.

END_COMPILE

Compilation end. Fields:

END_OF_PAGE_COUNT unsigned int

Number of basic blocks exited due to page boundary since START_COMPILE.

FETCHFAIL_COUNT unsigned int

Number of basic blocks exited due to ifetch failure START_COMPILE.

INST_COUNT unsigned int

Number of instructions compiled since START_COMPILE.

NONSEQ_COUNT unsigned int

Number of basic blocks exited due to non-sequential instructions since START_COMPILE.

PAGE_STRADDLE_COUNT unsigned int

Number of basic blocks exited due to unaligned instructions crossing page since START_COMPILE.

EXCEPTION

Exceptions that are taken. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ISSET enum

The instruction set of the processor when the exception occurred.

LR unsigned int

The value assigned to the link register.

NS enum

The core's non-secure bit.

PC unsigned int

The location where the exception occurred.

TARGET_ISET enum

The instruction set of the exception handler code.

TARGET_PC unsigned int

The address the exception branches to.

VECTOR enum

The exception vector.

EXCEPTION_END

Every exception completed. Fields:

NS enum

The core's non-secure bit.

EXCEPTION_ENTRY

Event marking the entry of an exception. All pushes of registers to the stack will follow this event. Fields:

PC unsigned int

The location where the exception occurred.

VECTOR enum

The exception that occurred.

EXCEPTION_RAISE

Every exception raised.

EXCEPTION_RETURN

Marks the end of an exception, but on an M core does not indicate a program flow change (branch). In most situations it will be followed by an indirect branch. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISET enum

The instruction set of the branch instruction.

PC unsigned int

The address of the branch instruction.

EXCEPTION_RETURN_PREBRANCH

Exception return event that must occur before PC and processor state have updated.

EXCEPTION_START

Every exception started. Fields:

NS enum

The core's non-secure bit.

EXCEPTION_VECTOR_FETCH

I-side vector fetch for M-class exception. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

DATA unsigned int

Result of fetch if OK.

RESPONSE enum

0=Aborted, 1=OK.

VADDR unsigned int

Where the vector is fetched from (including VTOR).

FP_STATE

Floating point state. Fields:

FPSR unsigned int

FPSR register value.

MASK unsigned int

The mask indicating updated cumulative bits.

FREQ_CHANGED

Core frequency changed. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The instruction count of this core.

NEW_FREQ signed int

The new frequency of this core (expressed in Hz).

OLD_FREQ signed int

The old frequency of this core (expressed in Hz).

HLT

HLT instruction occurred. Fields:

IMM16 unsigned int

The 16-bit immediate value encoded in HLT instruction.

INFO_EXCEPTION_REASON

Provide information of real cause of exception. Fields:

FaultCause enum

Which bit of which V7M/V8M-mainline FSR/HFSR would have been set.

PC unsigned int

The location where the exception occurred.

PHASE enum

What we are doing with the exception.

REASONS enum

Why we are doing it.

SecurityState enum

Privacy State of exception.

VECTOR enum

The exception that occurred.

INFO_FP_CONTEXT

Inform when ExecuteFPCheck writes CONTROL.FPCA and FPSCR etc. Fields:

WHAT enum

What we are doing.

INFO_STACKING

Provide information of real cause of exception. Fields:

PHASE enum

Are we starting or ending a sequence of stack writes.

REASONS enum

Why we are doing it.

INST

Every instruction executed. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DEBUG_STATE bool

The instruction is executed in debug state.

DISASS string

Disassembly of instruction.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISET enum

The current instruction set.

ITSTATE unsigned int

The current ITSTATE.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

MODE enum

The mode the core is in.

NS unsigned int

The core's non-secure bit.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

OPCODE unsigned int

The opcode of the instruction.

PADDR unsigned int

The physical address of the instruction.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the instruction.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the page.

PC unsigned int

The address of the instruction.

SECURITY_STATE enum

The core's security state.

SIZE unsigned int

The size of the instruction in bytes.

INST_END

Every instruction completed.

INST_START

Every instruction started. Fields:

DEBUG_STATE bool

The instruction is executed in debug state.

ISET enum

The current instruction set.

MODE enum

The mode the core is in.

NS enum

The current Secure State.

PC unsigned int

The address of the conditional instruction.

SECURITY_STATE enum

The current Security State.

INST_STRADDLE

Instruction straddles boundary. Fields:

DEBUG_STATE bool

The instruction is executed in debug state.

ISSET enum

The current instruction set.

MODE enum

The mode the core is in.

NS enum

The current Secure State.

PC unsigned int

The address of the conditional instruction.

SECURITY_STATE enum

The current Security State.

ISB

ISB instruction occurred. Fields:

PC unsigned int

The address of the ISB instruction.

ITM

Instrumentation Trace Macrocell. Fields:

ITM_PACKET_TYPE enum

ITM and DWT packets type.

PACKET_HEADER unsigned int

ITM Packet Header.

PACKET_PAYLOAD unsigned int

ITM Packet Payload.

LOAD_MULTIPLE_REGS_LIST

List of destination registers (not source) of a multiple registers load instruction. Fields:

REG string

Register to add into the list.

LOCAL_MONITOR

Local monitor activity. Fields:

PADDR unsigned int

Local Monitor Address.

State enum

State of the monitor (Open/Exclusive).

LOCKUP_CYCLE

This event is triggered if the core enters or stays in the lock-up state. Fields:

CAUSE enum

What causes the lockup?.

LOCKUP_ENTRY

This event is triggered if the core enters the lock-up state. Fields:

PC unsigned int

The PC when the lockup state is entered.

LOCKUP_EXIT

This event is triggered if the core leaves the lock-up state.

MODE_CHANGE

Mode change. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

MODE enum

The new mode.

OLD_MODE enum

The old mode.

MPU_TRANS

Address translation information. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

EXEC_PERM enum

Execution Permission.

IDAU_REG_NUM enum

IDAU region-number or code.

INNERCACHE_RA bool

Is the inner cache allocate on read.

INNERCACHE_TYPE enum

Inner Caching scheme (NC/MB/WA).

INNERCACHE_WA bool

Is the inner cache allocate on write.

MEMTYPE enum

Memory type.

NSDESC enum

Security-attributes of the transaction.

OUTERCACHE_RA bool

Is the outer cache allocate on read.

OUTERCACHE_TYPE enum

Outer Caching scheme (NC/MB/WA).

OUTERCACHE_WA bool

Is the outer cache allocate on write.

PADDR unsigned int

Address of the access.

PAGESIZE unsigned intPage size as $\log_2(\text{size})$.**PAS enum**

PAS accessed is secure or nonsecure memory.

READ_PERM enum

Read Permission.

REG_NUM unsigned int

MPU register based region-number 0-15 (else 0xFF=none, 0xBB/0xDD=background/default).

SAU_REG_NUM enum

SAU region-number or code.

SAU_RGN_TYPE enum

Privacy Region Type.

SH enum

Shareability.

SIDE enum

Inst / Data.

WRITE_PERM enum

Write Permission.

PERIODIC

Called for every quantum. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The instruction count of this CPU.

PC unsigned int

The address of the next instruction to be executed on this CPU.

POTINDEX_UPDATE

Value of POT index for current pc changed. Fields:

NEW_POTINDEX unsigned int

New POTIndex.

OLD_POTINDEX unsigned int

Old POTIndex.

PPB_READ_ACCESS

Trace reads from the core to memory mapped registers in the PPB (private peripheral bus) address range. Fields:

ACCESS_SIZE unsigned int

Logarithm of the access size: 0=byte, 1=halfword, 2=word, 3=doubleword, ...

BURST_LEN unsigned int

The number of data transfers (beats) in this burst.

DATA unsigned int

The data transferred.

INNER_ATTRIBUTES unsigned int

The inner attributes: bit0=write allocate, bit1=read allocate, bit2=cachable, bit3=bufferable, bit4=shareable.

INST bool

Is an instruction fetch.

LOCK enum

Normal / Locked / Exclusive.

MANAGER_ID unsigned int

The AXI manager ID.

NS bool

Is a nonsecure access.

OUTER_ATTRIBUTES unsigned int

The outer attributes: bit0=write allocate, bit1=read allocate, bit2=cachable, bit3=bufferable, bit4=shareable.

PADDR unsigned int

Physical address of access.

PRIV bool

Is privileged access.

RESPONSE enum

Whether the transaction was successful, or an error occurred.

USER_FLAGS unsigned int

Core specific additional signals.

PPB_WRITE_ACCESS

Trace writes from the core to memory mapped registers in the PPB (private peripheral bus) address range. Fields:

ACCESS_SIZE unsigned int

Logarithm of the access size: 0=byte, 1=halfword, 2=word, 3=doubleword, ...

BURST_LEN unsigned int

The number of data transfers (beats) in this burst.

DATA unsigned int

The data transferred.

INNER_ATTRIBUTES unsigned int

The inner attributes: bit0=write allocate, bit1=read allocate, bit2=cachable, bit3=bufferable, bit4=shareable.

INST bool

Is an instruction fetch.

LOCK enum

Normal / Locked / Exclusive.

MANAGER_ID unsigned int

The AXI manager ID.

NS bool

Is a nonsecure access.

OUTER_ATTRIBUTES unsigned int

The outer attributes: bit0=write allocate, bit1=read allocate, bit2=cachable, bit3=bufferable, bit4=shareable.

PADDR unsigned int

Physical address of access.

PRIV bool

Is privileged access.

RESPONSE enum

Whether the transaction was successful, or an error occurred.

USER_FLAGS unsigned int

Core specific additional signals.

PRELOAD_DATA

Data preload from PLD instruction. Fields:

VADDR unsigned int

Virtual address of the data that should be preloaded.

PRELOAD_INST

Instruction preload from PLI instruction. Fields:

VADDR unsigned int

Virtual address of the instruction that should be preloaded.

PRE_CORE_LOAD

Trace just before a core load. Fields:

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

PRE_CORE_STORE

Trace just before a core store. Fields:

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

PSTATE_PACTIVE_OPMODE_TRANSITION

Trace pactive and Opmode mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Operating mode after the transition.

OLD_STATE enum

Operating mode before the transition.

PSTATE_PACTIVE_OPMODE_TRANSITION_DENIED

Trace the denied pactive and Opmode mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Operating mode after the transition.

OLD_STATE enum

Operating mode before the transition.

REASON string

Reason for the transition being denied.

PSTATE_PACTIVE_POWER_TRANSITION

Trace pactive and power mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Power mode after the transition.

OLD_STATE enum

Power mode before the transition.

PSTATE_PACTIVE_POWER_TRANSITION_DENIED

Trace the denied pactive and power mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Power mode after the transition.

OLD_STATE enum

Power mode before the transition.

REASON string

Reason for the transition being denied.

PSTATE_PACTIVE_THREAD_OPMODE_TRANSITION

Trace pactive and Opmode mode transitions for threads on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Operating mode after the transition.

OLD_STATE enum

Operating mode before the transition.

QUANTUM_START

Called at the start of every quantum.

RUN_STATE

Run state transition. Fields:

INST_COUNT unsigned int

Ticks count at point of transition.

NEW enum

New run state.

OLD enum

Old run state.

SEMIHOSTING_CALL

Call of a semihost function occurred. Fields:

PC unsigned int

The program counter after the semihosting call.

SEMIHOSTING_PRECALL

About to call semihost. Fields:

EL2 bool

Use EL2 translation regime.

NS bool

Is Non-Secure.

REG_WIDTH unsigned int

The current register width in bytes.

SIGNAL

External signal state change. Fields:

SIGNAL enum

Signal that changed.

STATE bool

Signal asserted state.

START_COMPILE

Compilation started. Fields:

VADDR unsigned int

Instruction where compilation begins.

STG_MULTIPLE_MICRO_OPS_ADDRESS_RANGE

Address range for MTE stores with multiple uops. Fields:

ADDR_END unsigned int

Final address.

ADDR_START unsigned int

Starting address.

STORE_MULTIPLE_REGS_ADDRESS_RANGE

Address range for stores with multiple registers. Fields:

ADDR_END unsigned int

Final address.

ADDR_START unsigned int

Starting address.

SUBOPTIMAL_LDST_RETIRED

suboptimal load/store. Fields:

ACCESS_SIZE unsigned int

Log2 of the access width in bytes used for alignment checking (1: half-word, 2: word ...).

OPTIMAL_ALIGN_SELECTOR unsigned int

Selector to decide the optimal alignment size (1: access width, 2: 4-bytes, 3: 8-bytes, ..., 12: 4-kbytes).

VADDR unsigned int

The virtual address of the access.

SYNC

Called for every synchronization, that is every quantum break. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The instruction count of this CPU.

LOCAL_QUANTUM unsigned int

The local quantum of this CPU.

LOCAL_TIME unsigned int

The local time of this CPU.

SYNC_REASON enum

Reason for sync.

SYSCALL

System call instruction executed. Fields:

IMM unsigned int

Immediate value of the system call instruction.

TYPE enum

System call type.

VADDR unsigned int

Instruction that caused the system call.

SYSREG_READ

Trace reads from system registers. Fields:

REG_NAME enum

Register number.

VALUE unsigned int

The value read.

SYSREG_UPDATE128

Triggers when the system updates a register. Fields:

REG enum

Register number.

UNKNOWN unsigned int

Bits of the register which became unknown.

VALUE unsigned int

Value written to the register.

SYSREG_UPDATE32

Triggers when the system updates a register. Fields:

REG enum

Register number.

UNKNOWN unsigned int

Bits of the register which became unknown.

VALUE unsigned int

Value written to the register.

SYSREG_UPDATE64

Triggers when the system updates a register. Fields:

REG enum

Register number.

UNKNOWN unsigned int

Bits of the register which became unknown.

VALUE unsigned int

Value written to the register.

SYSREG_WRITE

Trace write to system registers. Fields:

REG_ACCESS enum

Register access status.

REG_NAME enum

Register number.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

Value written to the register.

UNALIGNED_LDST_RETIRED

Processor unaligned load/store. Fields:

ACCESS_SIZE unsigned int

Log2 of the access width in bytes used for alignment checking (1=>half-word, 2->word ...).

NUMBER_OF_BEATS unsigned int

Number of accesses (beats) in this burst (Total size in bytes is ACCESS_SIZE * NUMBER_OF_BEATS).

VADDR unsigned int

The virtual address of the access.

VFP_D_REGS

VFP/NEON D 64 bit register write. Fields:

ALIASING enum

Type of register aliasing used.

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a write to the corresponding bit in the register. For example, 0x1 would indicate a write to VALUE[0].

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

VFP_D_REGS_READ

VFP/NEON D 64 bit register read. Fields:

ID unsigned int

The register number.

VALUE unsigned int

The current value of the register.

VFP_Q_REGS

VFP/NEON Q 128 bit register write. Fields:

ALIASING enum

Type of register aliasing used.

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a write to the corresponding byte in the register. For example, 0x1 would indicate a write to VALUE[7:0].

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

VFP_Q_REGS_READ

VFP/NEON Q 128 bit register read. Fields:

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a read to the corresponding byte in the register. For example, 0x1 would indicate a read to VALUE[7:0].

VALUE unsigned int

The current value of the register.

VFP_SYS_REGS

Writes to the VFP/NEON units system registers. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ID enum

Which VFP system register is written.

MASK unsigned int

The mask indicating updated cumulative bits.

OLD_VALUE unsigned int

The register's old value overwritten.

VALUE unsigned int

The new value written to the VFP system register.

VFP_S_REGS

VFP/NEON S 32 bit register write. Fields:

ALIASING enum

Type of register aliasing used.

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a write to the corresponding bit in the register. For example, 0x1 would indicate a write to VALUE[0].

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

VFP_S_REGS_READ

VFP/NEON S 32 bit register read. Fields:

ID unsigned int

The register number.

VALUE unsigned int

The current value of the register.

WAYPOINT

Signals end of basic block execution. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ISET enum

Origin instruction set.

IS_COND bool

Indicates if this is a conditional waypoint.

PC unsigned int

Origin address (or 0 if unavailable).

TAKEN bool

Indicates if this waypoint was taken.

TARGET unsigned int

Destination address.

TARGET_ISET enum

Destination instruction set.

WFE_END

WFE ended. Fields:

INST_COUNT unsigned int

Ticks count when leaving WFE.

WFE_EVENT_REGISTER

WFE event register status: set/clear, reason. Fields:

INST_COUNT unsigned int

Ticks count.

REASON enum

Reason for set/clear. Only REASON==0 (Cleared by WFE) clears the bit, all other reasons set it.

WFE_IGNORED

WFE ignored. Fields:

EVENT bool

This WFE was ignored because the event register was set.

INST_COUNT unsigned int

Ticks count when ignoring WFE.

TRAPPED bool

This WFE was trapped.

WFE_START

WFE entered. Fields:

INST_COUNT unsigned int

Ticks count when entering WFE.

WFI_END

WFI ended. Fields:

INST_COUNT unsigned int

Ticks count when leaving WFI.

WFI_IGNORED

WFI ignored. Fields:

DISABLED bool

This WFI was ignored because WFI is disabled.

INST_COUNT unsigned int

Ticks count when ignoring WFI.

REASON enum

specifies reason why WFI trace was ignored.

TRAPPED bool

This WFI was trapped.

WFI_START

WFI entered. Fields:

INST_COUNT unsigned int

Ticks count when entering WFI.

WFI_WAKEUP

WFI wakeup. Fields:

INST_COUNT unsigned int

Ticks count when WFI wakeup occurred.

REASON enum

Reason for wakeup.

XPSR

Changes to the xPSR register. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

OLD_VALUE unsigned int

The old xPSR value.

UNKNOWN unsigned int

Bits within the register that have unknown value.

VALUE unsigned int

The new xPSR value.

2.43 ARM_Cortex-M35P

This section describes the trace sources.

ASYNC_MEMORY_FAULT

Context ID Register write. Fields:

FAULT unsigned int

Fault status present in ESR.ISS encoding for Serror in ARCH64 and present in DFSR encoding for external abort format in ARCH32.

PADDR unsigned int

Physical Address (or 0 if unavailable).

ATOMIC_END_ACCESS

Bus trace access for atomics. Fields:

ACCESS_FAIL bool

Memory access failed.

ADDR unsigned int

The virtual address of the access.

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate

when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

COMPARE_VALUE unsigned int

Compare value for CAS.

LOAD_VALUE unsigned int

Loaded values.

NSDESC unsigned int

The security state of the access.

OPERAND_VALUE unsigned int

Operation's operand.

OPERATION enum

Operation type.

PADDR unsigned int

The physical address of the access.

PRIV bool

Is this a privileged access?.

ATOMIC_START_ACCESS

Bus trace access for atomics. Fields:

ADDR unsigned int

The virtual address of the access.

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

COMPARE_VALUE unsigned int

Compare value for CAS.

NSDESC unsigned int

The security state of the access.

OPERAND_VALUE unsigned int

Operation's operand.

OPERATION enum

Operation type.

PADDR unsigned int

The physical address of the access.

PRIV bool

Is this a privileged access?.

ArchMsg.Error.exit_code

DISPLAY exit code: %{EXITCODE} component kind: %{KIND} reason: %{REASON}. Fields:

EXITCODE signed int

Exit code to be returned.

KIND string

Component kind that invoked the exit code trace.

REASON string

Reason why the exit code trace was invoked (optional).

ArchMsg.Warning.Unpredictably Indexed PM Event Register

TAGS **UNPREDICTABLE** DISPLAY Event %{IsDirect:(Selector value|Register)} %{SEL} is >= %{N}. Fields:

IsDirect unsigned int

Direct Access rather than indirect.

N unsigned int

Number of Accessible Registers.

SEL unsigned int

Selector.

ArchMsg.Warning.dap_csw_bad_size

A write to CM3DAP CSW has an invalid size field. DISPLAY Write %{DATA} to CM3DAP CSW has invalid size field. Fields:

DATA unsigned int

bits[2:0] are size.

ArchMsg.Warning.dcimvac_matches_watchpoint

TAGS **UNPREDICTABLE** DISPLAY Watchpoints matching an AArch32 DCIMVA are implementation defined.

ArchMsg.Warning.decode_fieldmismatch

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with different fields %{FIELD1} and %{FIELD2}. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_fracbitsnegative

TAGS **UNPREDICTABLE** DISPLAY VCVT Instruction is unpredictable with a negative number of fraction bits.

ArchMsg.Warning.decode_initblock

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable in an IT block.

ArchMsg.Warning.decode_initblocknotlast

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable in an IT block when not the last.

ArchMsg.Warning.decode_invalid_condition

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable when the condition is not AL.

ArchMsg.Warning.decode_invalidfieldcombination

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with the values in %{FIELD1} and %{FIELD2}. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_invalidvalue

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with the value in %{FIELD}. Fields:

FIELD enum

Field name.

ArchMsg.Warning.decode_mem_hint_unallocated

TAGS **UNPREDICTABLE** DISPLAY There is no memory hint allocated to this bit pattern.

ArchMsg.Warning.decode_registermatch

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with %{FIELD1} and %{FIELD2} as the same register. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_registermismatch

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with %{FIELD1} and %{FIELD2} as different registers. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_registeroutofrange

TAGS **UNPREDICTABLE** DISPLAY Floating-point instruction is unpredictable indexing beyond the end of the register bank.

ArchMsg.Warning.decode_sbzsbo

TAGS **UNPREDICTABLE** DISPLAY Reserved bits in the instruction are not canonical.

ArchMsg.Warning.decode_transactiontoobig

TAGS UNPREDICTABLE DISPLAY Attempt to perform a memory transaction of more than 128 bytes.

ArchMsg.Warning.decode_unpred_other

TAGS UNPREDICTABLE DISPLAY Instruction is unpredictable.

ArchMsg.Warning.decode_unpreduseofpc

TAGS UNPREDICTABLE DISPLAY Instruction is unpredictable when %{FIELD} is PC. Fields:

FIELD enum

Field name.

ArchMsg.Warning.decode_unpreduseofr13

TAGS UNPREDICTABLE DISPLAY Instruction is unpredictable when %{FIELD} is R13. Fields:

FIELD enum

Field name.

ArchMsg.Warning.decode_unpreduseofr13inreglist

TAGS UNPREDICTABLE DISPLAY Instruction is unpredictable with R13 in register-list.

ArchMsg.Warning.decode_useofd16orhigher

TAGS UNPREDICTABLE DISPLAY Attempt to use register D16 or higher where it is not allowed.

ArchMsg.Warning.decode_writebackandbaseinlist

TAGS UNPREDICTABLE DISPLAY Instruction is unpredictable with write back when the base register is in the transfer list.

ArchMsg.Warning.decode_zeroregistersinlist

TAGS UNPREDICTABLE DISPLAY Instruction is unpredictable when no registers are to be transferred.

ArchMsg.Warning.recursive_branch

An instruction is performing a branch that targets the same instruction. If this was intended then a WFI might be more effective DISPLAY recursive branch detected at PC=%{PC}. PRIMARY KEY PC. Fields:

PC unsigned int

Address of instruction causing the branch.

ArchMsg.Warning.recursive_exception

An instruction has generated an exception that targets the same instruction. DISPLAY recursive %{TYPE} exception detected at PC=%{PC} (CPSR=%{CPSR}). PRIMARY KEY PC. Fields:

CPSR unsigned int

Processor state.

PC unsigned int

Address of instruction causing the exception.

TYPE enum

Type of exception.

ArchMsg.Warning.reentrant_vector_catch

TAGS **UNPREDICTABLE** DISPLAY Vector catch debug event on Prefetch or Data abort vector %{ADDR}. Fields:

ADDR unsigned int

vector address.

ArchMsg.Warning.reserved_it_state

TAGS **UNPREDICTABLE** DISPLAY Execution with reserved IT state %{ITSTATE}. Fields:

ITSTATE unsigned int

reserved IT state.

ArchMsg.Warning.secure_vector_fetch_from_nonsecure

DISPLAY Secure vector table is being fetched from Non-secure memory.

ArchMsg.Warning.sg_in_it_block

Arm recommends that software does not place SG (security gate instruction) inside an IT instruction block.

ArchMsg.Warning.unpred_FAR_on_LRCPC3_non_atomic_mem_access

TAGS **UNPREDICTABLE** Unpredictable FAR/FSTATUS on LRCPC3 pair instructions resulting in two separate memory accessesDISPLAY Unpredictable FAR on LRCPC3 non atomic memory access at %{ADDR1} with FSTATUS %{FSTATUS1} or %{ADDR2} with FSTATUS %{FSTATUS2}. Fields:

ADDR1 unsigned int

Possible FAR address 1.

ADDR2 unsigned int

Possible FAR address 2.

FSTATUS1 unsigned int

Possible fault status value 1.

FSTATUS2 unsigned int

Possible fault status value 2.

ArchMsg.Warning.unpred_async_tag_check_fail_on_sve_speculative_load_do_update_ffr

Unpredictable set to FALSE of the FFR element associated to SVE speculative load with an async tag check fail TAGS **UNPREDICTABLE**.

ArchMsg.Warning.unpredictable_a32_breakpoint

TAGS **UNPREDICTABLE** DISPLAY Unpredictable breakpoint on a A32 instruction at %{ADDR1} Offset %{OFFSET} BAS %{BAS} for a %{IS_ADDRESS_MISMATCH_BREAKPOINT:(Mismatch| Match)} Breakpoint number %{N:d} Breakpoint address %{ADDR2}. Fields:

ADDR1 unsigned int

Address.

ADDR2 unsigned int

Breakpoint address.

BAS unsigned int

BAS field.

IS_ADDRESS_MISMATCH_BREAKPOINT bool

Is a Mismatch breakpoint, not a Match breakpoint.

N unsigned int

Breakpoint number.

OFFSET unsigned int

Always has a value of zero.

ArchMsg.Warning.unpredictable_banked_register_access

TAGS **UNPREDICTABLE** DISPLAY Access to banked register (R=%{R:d} SYSm=%{SYSM:2x}) is unpredictable. Fields:

R unsigned int

R bit.

SYSM unsigned int

SYSm.

ArchMsg.Warning.unpredictable_ccfail_undef

TAGS **UNPREDICTABLE** DISPLAY An instruction %{OPCODE} at VA=%{VADDR} failed its condition codes check but would otherwise have caused a non-data-dependent trap or undefined exception. Behaviour is implementation-defined choice of exception or nop. Fields:

OPCODE unsigned int

opcode of the instruction.

VADDR unsigned int

Virtual address of the instruction being executed.

ArchMsg.Warning.unpredictable_ici_bits

TAGS **UNPREDICTABLE** DISPLAY ICI bits do not apply or are architecturally unpredictable for instruction.

ArchMsg.Warning.unpredictable_t32_breakpoint

TAGS **UNPREDICTABLE** DISPLAY Unpredictable breakpoint on a T32 instruction at %{ADDR1} Offset %{OFFSET} BAS %{BAS} for a %{IS_ADDRESS_MISMATCH_BREAKPOINT:(Mismatch|Match)} Breakpoint number %{N:d} Breakpoint address %{ADDR2}. Fields:

ADDR1 unsigned int

Address.

ADDR2 unsigned int

Breakpoint address.

BAS unsigned int

BAS field.

IS_ADDRESS_MISMATCH_BREAKPOINT bool

Is a Mismatch breakpoint, not a Match breakpoint.

N unsigned int

Breakpoint number.

OFFSET unsigned int

Indicates which halfword of a 32-bit Thumb instruction the breakpoint is set on. 0 for the first halfword or if the instruction is 16 bits wide, or 1 for the second halfword.

ArchMsg.Warning.unpredictable_t32_vector_catch

TAGS **UNPREDICTABLE** DISPLAY Unpredictable Vector catch exception on a T32 instruction at %{ADDR} Offset %{OFFSET}. Fields:

ADDR unsigned int

Address.

OFFSET unsigned int

Indicates which halfword of a 32-bit Thumb instruction the vector catch exception is set on. 0 for the first halfword or if the instruction is 16 bits wide, or 1 for the second halfword.

ArchMsg.Warning.unpredictable_watchpoint_far

TAGS **UNPREDICTABLE** DISPLAY The value for FAR/EDWAR for hitting this watchpoint may be anywhere in the range %{LOWER_BOUND} to %{UPPER_BOUND}. Fields:

LOWER_BOUND unsigned int

the lowest address accessed by the instruction that triggered the watchpoint.

UPPER_BOUND unsigned int

the highest watchpointed address accessed by that instruction.

ArchMsg.Warning.warning_AdvSIMDEExpandImmUnexpectedZero

TAGS **UNPREDICTABLE** DISPLAY AdvSIMDEExpandImm may treat this immediate value as **UNPREDICTABLE**.

ArchMsg.Warning.warning_ConditionalSMC

TAGS **UNPREDICTABLE** DISPLAY SMC instruction has **UNPREDICTABLE** effects when conditional and when combined with traps.

ArchMsg.Warning.warning_access_crosses_page_boundary

TAGS **UNPREDICTABLE** DISPLAY Access crosses page boundary.

ArchMsg.Warning.warning_access_crosses_page_boundary_has_fault

TAGS **UNPREDICTABLE** DISPLAY Access that crosses page boundary has fault(s).

ArchMsg.Warning.warning_access_crosses_page_boundary_has_fault_on_both_pages

TAGS **UNPREDICTABLE** DISPLAY Access that crosses page boundary has a fault on each page. Fields:

ADDR1 unsigned int

FAR address on first page.

ADDR2 unsigned int

FAR address on second page.

FSTATUS1 unsigned int

Fault status on first page.

FSTATUS2 unsigned int

Fault status on second page.

ArchMsg.Warning.warning_access_crosses_page_boundary_spanning_different_memory

TAGS **UNPREDICTABLE** DISPLAY Access at address %{ADDR} crosses page from MEM_TYPE %{MEMTYPE_PAGE1} to %{MEMTYPE_PAGE2}. Fields:

ADDR unsigned int

address of access crossing page.

MEMTYPE_PAGE1 signed int

First Page's memory type (0-Normal, 1-Non-Normal).

MEMTYPE_PAGE2 signed int

Second Page's memory type (0-Normal, 1-Non-Normal).

ArchMsg.Warning.warning_access_wraps_around_memory

TAGS **UNPREDICTABLE** DISPLAY Access wraps around memory in %{WIDTH:d}bit mode. Fields:

WIDTH unsigned int

Address width.

ArchMsg.Warning.warning_bcr_linking_status

TAGS **UNPREDICTABLE** DISPLAY DBGBCR.{BT2,BT} combination gives unpredictable behaviour.

ArchMsg.Warning.warning_bcr_mask_reserved

TAGS **UNPREDICTABLE** DISPLAY DBGBCR_ELn.MASK is set to a reserved value.

ArchMsg.Warning.warning_bvr_masked_bit_not_zero

TAGS **UNPREDICTABLE** DISPLAY DBGBVRn_EL1.VA contains a bit masked by DBGBCRn_EL1.MASK that is not zero.

ArchMsg.Warning.warning_ccsidr_unimplemented_level

TAGS **UNPREDICTABLE** DISPLAY CCSIDR read while CSSELR %{CSSELR:x} points to an unimplemented cache level. Fields:

CSSELR unsigned int

current effective value of CSSELR.

ArchMsg.Warning.warning_change_to_ns_when_tge_set

TAGS **UNPREDICTABLE** DISPLAY Attempting to change NS to 1 when HCR.TGE = 1.

ArchMsg.Warning.warning_contiguous_bit_check_abort

TAGS **UNPREDICTABLE** DISPLAY An abort occurred whilst attempting to check TLB entry is contiguous at %{ENTRY_ADDR}. Fields:

ENTRY_ADDR unsigned int

address of conflicting TLB entry.

ArchMsg.Warning.warning_contiguous_bit_error

TAGS **UNPREDICTABLE** DISPLAY TLB entry at %{ENTRY_ADDR} was not contiguous with entry at %{CONTIG_ADDR} Contents are %{ENTRY_DATA} but expected %{CONTIG_DATA}. Fields:

CONTIG_ADDR unsigned int

address of the TLB first read, with which this entry is expected to be contiguous.

CONTIG_DATA unsigned int

expected contents based on the entry at CONTIG_ADDR.

ENTRY_ADDR unsigned int

address of conflicting TLB entry.

ENTRY_DATA unsigned int

contents of conflicting TLB entry.

ArchMsg.Warning.warning_contiguous_bit_error_D128

TAGS **UNPREDICTABLE** DISPLAY TLB entry at %{ENTRY_ADDR} was not contiguous with entry at %{CONTIG_ADDR} Contents are %{ENTRY_DATA_HI} %{ENTRY_DATA_LO} but expected %{CONTIG_DATA}. Fields:

CONTIG_ADDR unsigned int

address of the TLB first read, with which this entry is expected to be contiguous.

CONTIG_DATA unsigned int

expected contents based on the entry at CONTIG_ADDR.

ENTRY_ADDR unsigned int

address of conflicting TLB entry.

ENTRY_DATA_HI unsigned int

Higher bit contents of conflicting TLB entry.

ENTRY_DATA_LO unsigned int

Lower bit contents of conflicting TLB entry.

ArchMsg.Warning.warning_cp10_cp11_mismatch

TAGS **UNPREDICTABLE** DISPLAY Unpredictable configuration of CP10 and CP11 controls at EL %{EL:d}. Fields:

EL unsigned int

EL owning the control.

ArchMsg.Warning.warning_cp10_cp11_reserved_value

TAGS **UNPREDICTABLE** DISPLAY Unpredictable configuration of CPACR CP10 ({CP10}) or CP11 ({CP11}). Fields:

CP10 unsigned int

CP10 access.

CP11 unsigned int

CP11 access.

ArchMsg.Warning.warning_csselr_level_out_of_range

TAGS **UNPREDICTABLE** DISPLAY CSSELR.Level out of range, written value {LEVEL} with only {IMPLEMENTED:d} levels implemented. Fields:

IMPLEMENTED unsigned int

number of cache levels implemented.

LEVEL unsigned int

written value of CSSELR.Level.

TYPE enum

The type of cache selected by CSSELR.InD and TnD.

ArchMsg.Warning.warning_debug_flow_control_bits_not_obeyed

TAGS **UNPREDICTABLE** DISPLAY Flow control bits for {REGISTER} were not obeyed, causing {ERROR}. Fields:

ERROR enum

type of error resulting.

REGISTER enum

register under consideration.

ArchMsg.Warning.warning_debug_register_access_during_reset

TAGS **UNPREDICTABLE** DISPLAY Unpredictable {IS_WRITE:(read|write)} access to debug register offset {OFFSET} during reset. Fields:

IS_WRITE unsigned int

Write Not Read.

OFFSET unsigned int

Register Offset.

ArchMsg.Warning.warning_decode_cps_inconsistent_fields

TAGS **UNPREDICTABLE** DISPLAY The mode setting fields in the CPS instruction are inconsistent.

ArchMsg.Warning.warning_decode_invalid_state

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable in current exception-level/security state.

ArchMsg.Warning.warning_default_cacheable_mmu_on

TAGS **UNPREDICTABLE** DISPLAY Default cacheable is enabled (HCR.DC = 1) while MMU is enabled.

ArchMsg.Warning.warning_default_cacheable_vmmu_off

TAGS **UNPREDICTABLE** DISPLAY Default cacheable is enabled (HCR.DC = 1) while HCR.VM = 0.

ArchMsg.Warning.warning_deprecated_wvr_bit_2_set

TAGS **UNPREDICTABLE** DISPLAY DBGWVR%{N:d}_EL1=%{WVR:h} has bit 2 set, deprecated in ARMv8. Fields:

N unsigned int

index.

WVR unsigned int

DBGWVR.

ArchMsg.Warning.warning_eret_while_software_step_active_pending

Missing ISB between setting MDSCR_EL1.SS and ERET TAGS **UNPREDICTABLE**.

ArchMsg.Warning.warning_exclusive_to_non_normal

TAGS **UNPREDICTABLE** DISPLAY Exclusive Access to Non-Normal Memory %{ADDR}. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_exclusive_to_non_writeback

TAGS **UNPREDICTABLE** DISPLAY Exclusive Access to Non-Writeback-Cacheable Memory %{ADDR}. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_execute_from_device_memory

TAGS **UNPREDICTABLE** DISPLAY An attempt was made to execute from Device memory.

ArchMsg.Warning.warning_illegal_cpsr_mode

TAGS **UNPREDICTABLE** DISPLAY Writing an illegal or unimplement mode (%{NEW_MODE:h}) to CPSR was unpredictable. Fields:

NEW_MODE unsigned int

New value of CPSR.M.

ArchMsg.Warning.warning_illegal_srs_mode

TAGS **UNPREDICTABLE** DISPLAY Illegal or **UNPREDICTABLE** mode (%{MODE:h}) used for SRS instruction. Fields:

MODE unsigned int

mode for Banked SP.

ArchMsg.Warning.warning_implementation_defined_read_debug_register_in_SCS

Read Access to Debug registers from the processor is IMPLEMENTATION DEFINED TAGS IMP_DEF. Fields:

PPB_OFFSET unsigned int
debug register offset.

ArchMsg.Warning.warning_implementation_defined_sequential_security_transitions_supported

The behavior of sequential instruction fetches that cross from non-secure to secure memory and contain an SG instruction is CONSTRAINED UNPREDICTABLE TAGS UNPREDICTABLE.

ArchMsg.Warning.warning_implementation_defined_stack_limit_check_supported

It is IMPLEMENTATION DEFINED whether stack pointer limit checking is performed for this instruction TAGS IMP_DEF. Fields:

ADDRESS unsigned int
Address of instruction.

OPCODE unsigned int
instruction opcode.

ArchMsg.Warning.warning_implementation_defined_taggednonsharedmemorytype

TAGS IMP_DEF DISPLAY Memory attribute has Tagged, non shared memory type; it is IMPLEMENTATION DEFINED whether the memory location is treated as Tagged or non-Tagged.

ArchMsg.Warning.warning_implementation_defined_write_debug_register_in_SCS

Write Access to Debug registers from the processor is IMPLEMENTATION DEFINED TAGS IMP_DEF. Fields:

DATA unsigned int
data attempted to be written.

PPB_OFFSET unsigned int
debug register offset.

ArchMsg.Warning.warning_invalid_tcr_granule

TAGS UNPREDICTABLE DISPLAY TCR.TG%{TG_ID:(0|1)} bits have been set to a granule size %{{REQUEST:d}}K not implemented - using %{{SUBSTITUTE:d}}K. Fields:

REQUEST signed int
page size requested (or 0 for reserved).

SUBSTITUTE signed int
best guess available page size.

TG_ID bool
bits TG0 or TG1.

ArchMsg.Warning.warning_load_multiple_user_registers_from_user_mode

TAGS UNPREDICTABLE DISPLAY An LDM(user registers) instruction executed from user mode.

ArchMsg.Warning.warning_mte_in_mem_attr_not_supported

TAGS UNPREDICTABLE DISPLAY Memory attribute %{ATTR} has Tagged memory type but Armv8.5-MTE is not implemented. Fields:

ATTR unsigned int

Memory attribute.

ArchMsg.Warning.warning_reserved_breakpoint_state_match

TAGS UNPREDICTABLE DISPLAY %{IS_BREAKPOINT:(Watchpoint|Breakpoint)} programmed with a reserved combination of HMC, SSC and %{IS_BREAKPOINT:(PAC|PMC)}. Fields:

IS_BREAKPOINT unsigned int

Is a breakpoint, not a watchpoint.

ArchMsg.Warning.warning_shareability

TAGS UNPREDICTABLE DISPLAY Unpredictable: combination of the 1st and 2nd stages of translation is Normal Inner Non-Cacheable, Outer Non-Cacheable.

ArchMsg.Warning.warning_software_step_set_while_enabled

TAGS UNPREDICTABLE DISPLAY MDSCR_EL1.SS set to 1 while software step debug exceptions are enabled.

ArchMsg.Warning.warning_thumb_instruction_wraps_around_memory

TAGS UNPREDICTABLE DISPLAY Thumb instruction wraps around memory.

ArchMsg.Warning.warning_ttbr_sbz_bits_are_not_zero

TAGS UNPREDICTABLE DISPLAY %{TTBR}[%{MSB:d}:0] should be zero, but are not. Fields:

MSB unsigned int

x-1.

TTBR enum

which TTBR register.

ArchMsg.Warning.warning_unaligned_address_dbgdtrrx_write

TAGS UNPREDICTABLE DISPLAY Unaligned address %{ADDR} in X0/R0 when DBGDTRRX written in memory access mode. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unaligned_address_dbgdtrtx_read

TAGS UNPREDICTABLE DISPLAY Unaligned address %{ADDR} in X0/R0 when DBGDTRTX read in memory access mode. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unaligned_to_device

TAGS UNPREDICTABLE DISPLAY Unaligned Access to Device Memory %{ADDR}. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unaligned_to_strongly_ordered

TAGS **UNPREDICTABLE** DISPLAY Unaligned Access to Strongly Ordered Memory %{ADDR}.

Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unknown_sau_rnr

TAGS **UNPREDICTABLE** DISPLAY SAU_RNR was set to an unsupported value.

ArchMsg.Warning.warning_unpredictable_AIRCR_PRIS_and_BFHFNMINs

TAGS **UNPREDICTABLE** DISPLAY The effect of setting both AIRCR.BFHFNMINs and AIRCR.PRIS to 1 is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_AIRCR_VECTCLRACTIVE_when_not_in_debug

TAGS **UNPREDICTABLE** DISPLAY The effect of writing a 1 to AIRCR.VECTCLRACTIVE if the processor is not halted in Debug state is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_AIRCR_VECTRESET_and_SYSRESETREQ

TAGS **UNPREDICTABLE** DISPLAY When the processor is halted in Debug state, if a write to the register writes a 1 to both VECTRESET and SYSRESETREQ, the behavior is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_AIRCR_VECTRESET_when_not_in_debug

TAGS **UNPREDICTABLE** DISPLAY The effect of writing a 1 to AIRCR.VECTRESET if the processor is not halted in Debug state is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_AIRCR_incorrect_VKEY

TAGS **UNPREDICTABLE** DISPLAY The value 0x05FA must be written to AIRCR.VKEY, otherwise the register write is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_EXC_RETURN_Reserved_Bit

EXC_RETURN[23:7] are reserved with the special condition that all bits should be written as one. Values other than all 1s are **UNPREDICTABLE**. TAGS **UNPREDICTABLE**. Fields:

EXC_RETURN unsigned int

Special PC value.

SBO_MASK unsigned int

Which bits Should Be One to not UNPRED.

ArchMsg.Warning.warning_unpredictable_PMEVTYPEPn_ELO_VS_reserved_value

TAGS **UNPREDICTABLE** DISPLAY Unpredictable reserved value of PMEVTYPEn<n>_ELO.VS.

ArchMsg.Warning.warning_unpredictable_change_to_DEMCR_MON_STEP_at_insufficient_priority

TAGS **UNPREDICTABLE** DISPLAY The effect of changing DEMCR.MON_STEP at an execution priority that is lower than the priority of the DebugMonitor exception is **CONSTRAINED UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_change_to_priority_of_active_exception

TAGS UNPREDICTABLE DISPLAY Changing the priority of an active exception is UNPREDICTABLE.

ArchMsg.Warning.warning_unpredictable_clear_lspact

TAGS UNPREDICTABLE DISPLAY UNPREDICTABLE state on context switch. Fields:

DRVEXC_TAKEN unsigned int

Derived exception taken, not pended.

REACHED_LAST_STORE unsigned int

Only the last faulted.

ArchMsg.Warning.warning_unpredictable_dbg_exit_unaligned_dlr

TAGS UNPREDICTABLE DISPLAY Debug exit to AArch32 with DLR[0] set to 1. PC[0] can be set to 0 or to DLR[0].

ArchMsg.Warning.warning_unpredictable_exception_catch

TAGS UNPREDICTABLE DISPLAY Generation of an exception catch event is unpredictable.

ArchMsg.Warning.warning_unpredictable_exception_return_inconsistent_state

TAGS UNPREDICTABLE DISPLAY state on exception return is unpredictable.

ArchMsg.Warning.warning_unpredictable_exception_return_instruction

TAGS UNPREDICTABLE DISPLAY exception return instruction is unpredictable.

ArchMsg.Warning.warning_unpredictable_in_debug_state

TAGS UNPREDICTABLE DISPLAY Instruction is UNPREDICTABLE when executed in debug state.

ArchMsg.Warning.warning_unpredictable_mair_encoding

TAGS UNPREDICTABLE DISPLAY Memory attribute %{ATTR} has invalid MAIR encoding; this may lead to unpredictable results. Fields:

ATTR unsigned int

Memory attribute.

ArchMsg.Warning.warning_unpredictable_pmu_counter_access

TAGS UNPREDICTABLE DISPLAY Access to PMU counters from non-secure EL0 or EL1 is UNPREDICTABLE with MDCR_EL2.HPMN set to 0.

ArchMsg.Warning.warning_unpredictable_prioritization_breakpoint_match_vector_catch

TAGS UNPREDICTABLE DISPLAY Two events with the same priority occurred at the same instruction: Address Matching Vector Catch debug event, and Breakpoint debug event. It is constrained unpredictable which is taken.

ArchMsg.Warning.warning_unpredictable_stack_selection

TAGS UNPREDICTABLE DISPLAY UNPREDICTABLE state on context switch.

ArchMsg.Warning.warning_unpredictable_tsize_out_of_range

TAGS UNPREDICTABLE DISPLAY %{TCR}.T%{TTBR:d}SZ (%{TSIZE:d} bits) is out of range. Must be between 25 and %{TSIZE_MAX:d}. Fields:

TCR enum

Which TCR.

TSize signed int

Value of TSize.

TSize_MAX signed int

The maximum allowed value.

TTBR unsigned int

Which TTBR region.

ArchMsg.Warning.warning_unpredictable_unaligned_pc_as_base_register

TAGS **UNPREDICTABLE** DISPLAY The PC value must be word-aligned, otherwise the behavior of the instruction is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_unaligned_pop_stack

TAGS **UNPREDICTABLE** DISPLAY **CONSTRAINED UNPREDICTABLE** unaligned stack frame `%{FRAME_ADDR:x}` in exception-return. Fields:

FRAME_ADDR unsigned int

unaligned stack frame address.

ArchMsg.Warning.warning_unpredictable_unaligned_to_non_writeback

TAGS **UNPREDICTABLE** DISPLAY Unaligned Access to Non-Writeback-Cacheable Memory.

ArchMsg.Warning.warning_unpredictable_vmsa_memattrib

TAGS **UNPREDICTABLE** DISPLAY Unpredictable vmsa memory attribute with texcb value programmed to `%{TEXCB}` when tex remap is `%{TEX_REMAP}`. Fields:

TEXCB unsigned int

texcb value.

TEX_REMAP bool

use tex remap.

ArchMsg.Warning.warning_unpredictable_vtcr_t0sz_sl0

TAGS **UNPREDICTABLE** DISPLAY The combination of VTCR.TOSZ=`%{TOSZ:d}` and VTCR.SL0=`%{SL0:d}` is unpredictable. Fields:

SL0 signed int

VTCR.SL0, starting level for stage 2 translation table walks.

TOSZ signed int

VTCR.TOSZ, required input address range.

ArchMsg.Warning.warning_unpredictable_write_DHCSR

TAGS **UNPREDICTABLE** DISPLAY **UNPREDICTABLE** set of bit-changes in DHCSR. Fields:

NEW unsigned int

data attempted to be written.

OLD unsigned int

previous value.

ArchMsg.Warning.warning_unpredictable_write_SHCSR

TAGS UNPREDICTABLE DISPLAY UNPREDICTABLE setting a pending bit to 1 for an exception with priority \geq execution priority.

ArchMsg.Warning.warning_unsupported_access_to_memory_mapped_register

TAGS UNPREDICTABLE DISPLAY Access to memory mapped register is unsupported.

ArchMsg.Warning.warning_user_jmcr_access

TAGS UNPREDICTABLE DISPLAY User %{WRITE:(write|read)} access to JMCR is unpredictable. Fields:

WRITE unsigned int
access is write.

ArchMsg.Warning.warning_wcr_mask_and_bas

TAGS UNPREDICTABLE DISPLAY DBGWCR_EL1.MASK is non-zero and BAS is not 0xff.

ArchMsg.Warning.warning_wcr_mask_reserved

TAGS UNPREDICTABLE DISPLAY DBGWCR_ELn.MASK is set to a reserved value.

ArchMsg.Warning.warning_wcr_non_configuous_bas

TAGS UNPREDICTABLE DISPLAY DBGWCR_EL1.BAS has non-contiguous set of ones.

ArchMsg.Warning.warning_wvr_masked_bit_not_zero

TAGS UNPREDICTABLE DISPLAY DBGWVRn_EL1.VA contains a bit masked by DBGWCRn_EL1.MASK that is not zero.

ArchMsg.warning_unpred_system_register_access

DISPLAY Unpredictable %{IS_WRITE:(read|write)} access to system register offset %{OFFSET}. Fields:

IS_WRITE unsigned int
Write Not Read.

OFFSET unsigned int
Register Offset.

BRANCH_MISPREDICT

Simulating branch mispredict. Fields:

PC unsigned int
Origin address (or 0 if unavailable).

BRA_DIR

Direct branches, to immediate address. Fields:

CORE_NUM unsigned int
Core number in a multi processor.

INST_COUNT unsigned int
The core's instruction counter, starting at 1 for the first instruction.

ISSET enum

The instruction set of the branch instruction.

IS_COND bool

Indicates if this is a conditional branch.

IS_HINTED bool

Whether the conditional instruction is hinted.

IS_ISB bool

Whether the instruction is ISB.

IS_LINK bool

Indicates if this is a branch with link.

PC unsigned int

The address of the branch instruction.

TARGET_ISSET enum

The instruction set after the branch.

TARGET_PC unsigned int

The address the instruction branches to.

BRA_INDIR

Indirect branches, perhaps to a register. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISSET enum

The instruction set of the branch instruction.

IS_COND bool

Indicates if this is a conditional branch.

IS_HINTED bool

Whether the conditional instruction is hinted.

IS_ISB bool

Whether the instruction is ISB.

IS_LINK bool

Indicates if this is a branch with link.

PC unsigned int

The address of the branch instruction.

TARGET_ISSET enum

The instruction set after the branch.

TARGET_PC unsigned int

The address the instruction branches to.

BRA_INDIR_CCFAIL

Indirect branches fail, perhaps to a register.

BRA_RET_CCFAIL

Branch return fail, perhaps to a register.

CCFAIL

Conditional instruction condition check fail. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CCFAIL_UNC

Unconditional instruction condition check fail. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CCPASS

Conditional instruction condition check pass. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CCPASS_UNC

Unconditional instruction condition check pass. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CODE_CACHE_MAINT

Code cache maintenance. Fields:

END_INCL unsigned int

Inclusive end address for VA or PA requests.

NS bool

Non-secure world for PA.

START unsigned int

Start address for VA or PA requests.

TYPE enum

Request type.

COMPILE_BLOCK_END

Last instruction of basic block translated. Fields:

VADDR unsigned int

Address of next instruction after this basic block.

COMPILE_BLOCK_START

First instruction of basic block translated. Fields:

VADDR unsigned int

Address of first instruction in basic block.

COMPILE_INST

ARM instruction compiled. Fields:

DISASS string

Disassembly of the instruction.

ISSET enum

The instruction set of this instruction.

ITSTATE unsigned int

The ITSTATE current for the instruction.

OPCODE unsigned int

The opcode of the instruction.

PC unsigned int

The address of the instruction.

SIZE unsigned int

The size of the instruction in bytes.

CONTEXTIDR

Context ID Register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

NS bool

Secure or nonsecure banked register is accessed.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The new value written.

CONTEXT_SYNC

Called for every context synchronization event. Fields:

ARCH_CSE bool

Architectural Context Synchronization Event.

PC unsigned int

The address of the synchronization instruction.

SYSREG_SYNC bool

System register synchronization event.

CORE_ENDIAN

Core BE8 Big-Endian state changed. Fields:

BE8 bool

Core BE8 Big-Endian state.

CORE_INFO

Static processor attributes. Only triggered by a call to DumpState(). Fields:

ARCH_PROFILE enum

The architecture profile of the core.

CLUSTER_ID unsigned int

The cluster ID of this processor.

CORE_NUM unsigned int

The number of this core in an MP processor.

DETAILED_CORE_FEATURES enum

Detailed core features as a bitfield. Enum values are bitmasks to extract the info from the field.

FPU_VERSION enum

The VFP version implemented by the core.

MEM_ARCH enum

The memory architecture of the core.

MIDR_EL1 unsigned int

MIDR_EL1 value.

NUM_CORES unsigned int

The number of cores in this MP processor.

QUANTUM_SIZE unsigned int

The default quantum size of the core (in ticks).

SECURITY_FEATURES bool

Does the core have security features?.

CORE_LOADS

Processor load accesses. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ACQREL enum

Is this an acquire/release.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DATA unsigned int

The data read or written.

ELEMENT_SIZE unsigned int

Width of each element.

ENDIANNESS enum

Byte ordering of this access.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

LOCK enum

Normal, exclusive or locked access.

MECID unsigned int

Memory Encryption Context ID, if applicable.

MECID2 unsigned int

Memory Encryption Context ID of the second page, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PAGE_SIZE unsigned int

Page size as $\log_2(\text{size in bytes})$.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK, 2=Exclusive Failed, 3=Unsupported.

SIZE unsigned int

Width of the access in bytes. Only required if DATA is not traced.

TRANS bool

Is this a translated access.

VADDR unsigned int

The virtual address of the access.

CORE_REGS

Changes of the core registers R0 to R14. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number, 0 to 14.

OLD_VALUE unsigned int

The old value overwritten.

PHYS_ID enum

The physical register accessed.

VALUE unsigned int

The new value written to the register.

CORE_STALLED_TRANSITION

Detect when a core is prevented from progressing PChannel states. Fields:

CURRENT_STATE enum

Current power mode of core.

REASON string

The reason a transition was stalled, and what must be done to unblock it.

TARGET_STATE enum

Desired power mode of core after transition.

CORE_STORES

Processor store accesses. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ACQREL enum

Is this an acquire/release.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DATA unsigned int

The data read or written.

ELEMENT_SIZE unsigned int

Width of each element.

ENDIANNESS enum

Byte ordering of this access.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

LOCK enum

Normal, exclusive or locked access.

MECID unsigned int

Memory Encryption Context ID, if applicable.

MECID2 unsigned int

Memory Encryption Context ID of the second page, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PAGE_SIZE unsigned int

Page size as $\log_2(\text{size in bytes})$.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK, 2=Exclusive Failed, 3=Unsupported.

SIZE unsigned int

Width of the access in bytes. Only required if DATA is not traced.

TRANS bool

Is this a translated access.

VADDR unsigned int

The virtual address of the access.

CRYPTO_SPEC

Every crypto instruction speculatively executed.

DATA_CACHE_SET_ALLOC_TAGS

Set allocation tags from DC GVA instruction. Fields:

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

NSDESC unsigned int

The security state of the access.

PADDR unsigned int

The physical address of the access.

PAS enum

The physical address space of the page.

RESPONSE enum

0=Aborted, 1=OK.

SIZE unsigned int

The size of memory in bytes.

TAG unsigned int

The value of the tag being set.

TAGGING_TYPE enum

Type of Memory tagging (Physical or Virtual).

TAG_VADDR unsigned int

The virtual address of the Virtual Tag if Virtual tagging is used; otherwise 0.

VADDR unsigned int

Virtual address used for setting allocation tags.

DATA_CACHE_ZERO

Zero and invalidate cache line from DC ZVA instruction. Fields:

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

NSDESC unsigned int

The security state of the access.

PADDR unsigned int

The physical address of the access.

PADDR_TAG unsigned int

If tag checked, the physical address tag for the access.

PAS enum

The physical address space of the page.

RESPONSE enum

0=Aborted, 1=OK.

SIZE unsigned int

The size of zero'd memory in bytes.

TAG_CHECKED bool

Is this access tag checked.

VADDR unsigned int

Virtual address of the zero'd cache line.

DEBUG_EVENT

Hardware debug support event. Fields:

EVENT enum

Description of event.

VALUE unsigned int

data value.

DMI_ALLOCATE

DMI allocated. Fields:

ACCESS_DIRECTION enum

READ / WRITE.

PADDR unsigned int

The physical address of the DMI.

PTR unsigned int

DMI range pointer.

SIDE enum

DSIDE / ISIDE.

SIZE unsigned int

The size of DMI.

VADDR unsigned int

The virtual address of the DMI..

DMI_HIT

DMI hit, access used DMI (only fire for small DMI regions i.e. less than a full page). Fields:

ACCESS_DIRECTION enum

READ / WRITE.

PADDR unsigned int

The physical address of the DMI.

SIDE enum

DSIDE / ISIDE.

SIZE unsigned int

The size of the DMI.

VADDR unsigned int

The virtual address of the DMI..

DMI_REVOKE

DMI revoked. Fields:

ACCESS_DIRECTION enum

READ / WRITE.

PADDR unsigned int

The physical address of the DMI.

SIDE enum

DSIDE / ISIDE.

SIZE unsigned int

The size of DMI.

VADDR unsigned int

The virtual address of the DMI..

DWT_MATCH

DWT comparator matches. Fields:

NUM unsigned int

DWT comparator number.

TYPE enum

DWT comparator configuration.

END_COMPILE

Compilation end. Fields:

END_OF_PAGE_COUNT unsigned int

Number of basic blocks exited due to page boundary since START_COMPILE.

FETCHFAIL_COUNT unsigned int

Number of basic blocks exited due to ifetch failure START_COMPILE.

INST_COUNT unsigned int

Number of instructions compiled since START_COMPILE.

NONSEQ_COUNT unsigned int

Number of basic blocks exited due to non-sequential instructions since START_COMPILE.

PAGE_STRADDLE_COUNT unsigned int

Number of basic blocks exited due to unaligned instructions crossing page since START_COMPILE.

EXCEPTION

Exceptions that are taken. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ISSET enum

The instruction set of the processor when the exception occurred.

LR unsigned int

The value assigned to the link register.

NS enum

The core's non-secure bit.

PC unsigned int

The location where the exception occurred.

TARGET_ISSET enum

The instruction set of the exception handler code.

TARGET_PC unsigned int

The address the exception branches to.

VECTOR enum

The exception vector.

EXCEPTION_END

Every exception completed. Fields:

NS enum

The core's non-secure bit.

EXCEPTION_ENTRY

Event marking the entry of an exception. All pushes of registers to the stack will follow this event. Fields:

PC unsigned int

The location where the exception occurred.

VECTOR enum

The exception that occurred.

EXCEPTION_RAISE

Every exception raised.

EXCEPTION_RETURN

Marks the end of an exception, but on an M core does not indicate a program flow change (branch). In most situations it will be followed by an indirect branch. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISET enum

The instruction set of the branch instruction.

PC unsigned int

The address of the branch instruction.

EXCEPTION_RETURN_PREBRANCH

Exception return event that must occur before PC and processor state have updated.

EXCEPTION_START

Every exception started. Fields:

NS enum

The core's non-secure bit.

EXCEPTION_VECTOR_FETCH

I-side vector fetch for M-class exception. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

DATA unsigned int

Result of fetch if OK.

RESPONSE enum

0=Aborted, 1=OK.

VADDR unsigned int

Where the vector is fetched from (including VTOR).

FP_STATE

Floating point state. Fields:

FPSR unsigned int

FPSR register value.

MASK unsigned int

The mask indicating updated cumulative bits.

FREQ_CHANGED

Core frequency changed. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The instruction count of this core.

NEW_FREQ signed int

The new frequency of this core (expressed in Hz).

OLD_FREQ signed int

The old frequency of this core (expressed in Hz).

HLT

HLT instruction occurred. Fields:

IMM16 unsigned int

The 16-bit immediate value encoded in HLT instruction.

INFO_EXCEPTION_REASON

Provide information of real cause of exception. Fields:

FaultCause enum

Which bit of which V7M/V8M-mainline FSR/HFSR would have been set.

PC unsigned int

The location where the exception occurred.

PHASE enum

What we are doing with the exception.

REASONS enum

Why we are doing it.

SecurityState enum

Privacy State of exception.

VECTOR enum

The exception that occurred.

INFO_FP_CONTEXT

Inform when ExecuteFPCheck writes CONTROL.FPCA and FPSCR etc. Fields:

WHAT enum

What we are doing.

INFO_STACKING

Provide information of real cause of exception. Fields:

PHASE enum

Are we starting or ending a sequence of stack writes.

REASONS enum

Why we are doing it.

INST

Every instruction executed. Fields:

CORE_NUM *unsigned int*

Core number in a multi processor.

CURRENT_TIME *unsigned int*

The core's current time, as simulated time plus local time.

DEBUG_STATE *bool*

The instruction is executed in debug state.

DISASS *string*

Disassembly of instruction.

INST_COUNT *unsigned int*

The core's instruction counter, starting at 1 for the first instruction.

ISSET *enum*

The current instruction set.

ITSTATE *unsigned int*

The current ITSTATE.

LOCAL_TIME *unsigned int*

The core's local time, relative to the current quantum.

MODE *enum*

The mode the core is in.

NS *unsigned int*

The core's non-secure bit.

NSDESC *unsigned int*

The physical address non-secure bit.

NSDESC2 *unsigned int*

The second page physical address non-secure bit.

OPCODE *unsigned int*

The opcode of the instruction.

PADDR *unsigned int*

The physical address of the instruction.

PADDR2 *unsigned int*

If different from PADDR, the physical address of the second page of the instruction.

PAS *enum*

The physical address space of the page.

PAS2 *enum*

The physical address space of the page.

PC *unsigned int*

The address of the instruction.

SECURITY_STATE enum

The core's security state.

SIZE unsigned int

The size of the instruction in bytes.

INST_END

Every instruction completed.

INST_START

Every instruction started. Fields:

DEBUG_STATE bool

The instruction is executed in debug state.

ISET enum

The current instruction set.

MODE enum

The mode the core is in.

NS enum

The current Secure State.

PC unsigned int

The address of the conditional instruction.

SECURITY_STATE enum

The current Security State.

INST_STRADDLE

Instruction straddles boundary. Fields:

DEBUG_STATE bool

The instruction is executed in debug state.

ISET enum

The current instruction set.

MODE enum

The mode the core is in.

NS enum

The current Secure State.

PC unsigned int

The address of the conditional instruction.

SECURITY_STATE enum

The current Security State.

ISB

ISB instruction occurred. Fields:

PC unsigned int

The address of the ISB instruction.

ITM

Instrumentation Trace Macrocell. Fields:

ITM_PACKET_TYPE enum

ITM and DWT packets type.

PACKET_HEADER unsigned int

ITM Packet Header.

PACKET_PAYLOAD unsigned int

ITM Packet Payload.

LOAD_MULTIPLE_REGS_LIST

List of destination registers (not source) of a multiple registers load instruction. Fields:

REG string

Register to add into the list.

LOCAL_MONITOR

Local monitor activity. Fields:

PADDR unsigned int

Local Monitor Address.

State enum

State of the monitor (Open/Exclusive).

LOCKUP_CYCLE

This event is triggered if the core enters or stays in the lock-up state. Fields:

CAUSE enum

What causes the lockup?.

LOCKUP_ENTRY

This event is triggered if the core enters the lock-up state. Fields:

PC unsigned int

The PC when the lockup state is entered.

LOCKUP_EXIT

This event is triggered if the core leaves the lock-up state.

MODE_CHANGE

Mode change. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

MODE enum

The new mode.

OLD_MODE enum

The old mode.

MPU_TRANS

Address translation information. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

EXEC_PERM enum

Execution Permission.

IDAU_REG_NUM enum

IDAU region-number or code.

INNERCACHE_RA bool

Is the inner cache allocate on read.

INNERCACHE_TYPE enum

Inner Caching scheme (NC/MB/WA).

INNERCACHE_WA bool

Is the inner cache allocate on write.

MEMTYPE enum

Memory type.

NSDESC enum

Security-attributes of the transaction.

OUTERCACHE_RA bool

Is the outer cache allocate on read.

OUTERCACHE_TYPE enum

Outer Caching scheme (NC/MB/WA).

OUTERCACHE_WA bool

Is the outer cache allocate on write.

PADDR unsigned int

Address of the access.

PAGESIZE unsigned int

Page size as $\log_2(\text{size})$.

PAS enum

PAS accessed is secure or nonsecure memory.

READ_PERM enum

Read Permission.

REG_NUM unsigned int

MPU register based region-number 0-15 (else 0xFF=none, 0xBB/0xDD=background/default).

SAU_REG_NUM enum

SAU region-number or code.

SAU_RGN_TYPE enum

Privacy Region Type.

SH enum

Shareability.

SIDE enum

Inst / Data.

WRITE_PERM enum

Write Permission.

PERIODIC

Called for every quantum. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The instruction count of this CPU.

PC unsigned int

The address of the next instruction to be executed on this CPU.

POTINDEX_UPDATE

Value of POT index for current pc changed. Fields:

NEW_POTINDEX unsigned int

New POTIndex.

OLD_POTINDEX unsigned int

Old POTIndex.

PPB_READ_ACCESS

Trace reads from the core to memory mapped registers in the PPB (private peripheral bus) address range. Fields:

ACCESS_SIZE unsigned int

Logarithm of the access size: 0=byte, 1=halfword, 2=word, 3=doubleword, ...

BURST_LEN unsigned int

The number of data transfers (beats) in this burst.

DATA unsigned int

The data transferred.

INNER_ATTRIBUTES unsigned int

The inner attributes: bit0=write allocate, bit1=read allocate, bit2=cachable, bit3=bufferable, bit4=shareable.

INST bool

Is an instruction fetch.

LOCK enum

Normal / Locked / Exclusive.

MANAGER_ID unsigned int

The AXI manager ID.

NS bool

Is a nonsecure access.

OUTER_ATTRIBUTES unsigned int

The outer attributes: bit0=write allocate, bit1=read allocate, bit2=cachable, bit3=bufferable, bit4=shareable.

PADDR unsigned int

Physical address of access.

PRIV bool

Is privileged access.

RESPONSE enum

Whether the transaction was successful, or an error occurred.

USER_FLAGS unsigned int

Core specific additional signals.

PPB_WRITE_ACCESS

Trace writes from the core to memory mapped registers in the PPB (private peripheral bus) address range. Fields:

ACCESS_SIZE unsigned int

Logarithm of the access size: 0=byte, 1=halfword, 2=word, 3=doubleword, ...

BURST_LEN unsigned int

The number of data transfers (beats) in this burst.

DATA unsigned int

The data transferred.

INNER_ATTRIBUTES unsigned int

The inner attributes: bit0=write allocate, bit1=read allocate, bit2=cachable, bit3=bufferable, bit4=shareable.

INST bool

Is an instruction fetch.

LOCK enum

Normal / Locked / Exclusive.

MANAGER_ID unsigned int

The AXI manager ID.

NS bool

Is a nonsecure access.

OUTER_ATTRIBUTES unsigned int

The outer attributes: bit0=write allocate, bit1=read allocate, bit2=cachable, bit3=bufferable, bit4=shareable.

PADDR unsigned int

Physical address of access.

PRIV bool

Is privileged access.

RESPONSE enum

Whether the transaction was successful, or an error occurred.

USER_FLAGS unsigned int

Core specific additional signals.

PRELOAD_DATA

Data preload from PLD instruction. Fields:

VADDR unsigned int

Virtual address of the data that should be preloaded.

PRELOAD_INST

Instruction preload from PLI instruction. Fields:

VADDR unsigned int

Virtual address of the instruction that should be preloaded.

PRE_CORE_LOAD

Trace just before a core load. Fields:

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

PRE_CORE_STORE

Trace just before a core store. Fields:

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

PSTATE_PACTIVE_OPMODE_TRANSITION

Trace active and Opmode mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Operating mode after the transition.

OLD_STATE enum

Operating mode before the transition.

PSTATE_PACTIVE_OPMODE_TRANSITION_DENIED

Trace the denied active and Opmode mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Operating mode after the transition.

OLD_STATE enum

Operating mode before the transition.

REASON string

Reason for the transition being denied.

PSTATE_PACTIVE_POWER_TRANSITION

Trace active and power mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Power mode after the transition.

OLD_STATE enum

Power mode before the transition.

PSTATE_PACTIVE_POWER_TRANSITION_DENIED

Trace the denied active and power mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Power mode after the transition.

OLD_STATE enum

Power mode before the transition.

REASON string

Reason for the transition being denied.

PSTATE_PACTIVE_THREAD_OPMODE_TRANSITION

Trace pactive and Opmode mode transitions for threads on PChannel. Fields:

DEVPACTIVE **unsigned int**

Integer representation of DEVPACTIVE[] signals.

NEW_STATE **enum**

Operating mode after the transition.

OLD_STATE **enum**

Operating mode before the transition.

QUANTUM_START

Called at the start of every quantum.

RUN_STATE

Run state transition. Fields:

INST_COUNT **unsigned int**

Ticks count at point of transition.

NEW **enum**

New run state.

OLD **enum**

Old run state.

SEMIHOSTING_CALL

Call of a semihost function occurred. Fields:

PC **unsigned int**

The program counter after the semihosting call.

SEMIHOSTING_PRECALL

About to call semihost. Fields:

EL2 **bool**

Use EL2 translation regime.

NS **bool**

Is Non-Secure.

REG_WIDTH **unsigned int**

The current register width in bytes.

SIGNAL

External signal state change. Fields:

SIGNAL **enum**

Signal that changed.

STATE **bool**

Signal asserted state.

START_COMPILE

Compilation started. Fields:

VADDR unsigned int

Instruction where compilation begins.

STG_MULTIPLE_MICRO_OPS_ADDRESS_RANGE

Address range for MTE stores with multiple uops. Fields:

ADDR_END unsigned int

Final address.

ADDR_START unsigned int

Starting address.

STORE_MULTIPLE_REGS_ADDRESS_RANGE

Address range for stores with multiple registers. Fields:

ADDR_END unsigned int

Final address.

ADDR_START unsigned int

Starting address.

SUBOPTIMAL_LDST_RETIRED

suboptimal load/store. Fields:

ACCESS_SIZE unsigned int

Log2 of the access width in bytes used for alignment checking (1: half-word, 2: word ...).

OPTIMAL_ALIGN_SELECTOR unsigned int

Selector to decide the optimal alignment size (1: access width, 2: 4-bytes, 3: 8-bytes, ..., 12: 4-kbytes).

VADDR unsigned int

The virtual address of the access.

SYNC

Called for every synchronization, that is every quantum break. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The instruction count of this CPU.

LOCAL_QUANTUM unsigned int

The local quantum of this CPU.

LOCAL_TIME unsigned int

The local time of this CPU.

SYNC_REASON enum

Reason for sync.

SYSCALL

System call instruction executed. Fields:

IMM unsigned int

Immediate value of the system call instruction.

TYPE enum

System call type.

VADDR unsigned int

Instruction that caused the system call.

SYSREG_READ

Trace reads from system registers. Fields:

REG_NAME enum

Register number.

VALUE unsigned int

The value read.

SYSREG_UPDATE128

Triggers when the system updates a register. Fields:

REG enum

Register number.

UNKNOWN unsigned int

Bits of the register which became unknown.

VALUE unsigned int

Value written to the register.

SYSREG_UPDATE32

Triggers when the system updates a register. Fields:

REG enum

Register number.

UNKNOWN unsigned int

Bits of the register which became unknown.

VALUE unsigned int

Value written to the register.

SYSREG_UPDATE64

Triggers when the system updates a register. Fields:

REG enum

Register number.

UNKNOWN unsigned int

Bits of the register which became unknown.

VALUE unsigned int

Value written to the register.

SYSREG_WRITE

Trace write to system registers. Fields:

REG_ACCESS enum

Register access status.

REG_NAME enum

Register number.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

Value written to the register.

UNALIGNED_LDST_RETIRED

Processor unaligned load/store. Fields:

ACCESS_SIZE unsigned int

Log2 of the access width in bytes used for alignment checking (1=>half-word, 2->word ...).

NUMBER_OF_BEATS unsigned int

Number of accesses (beats) in this burst (Total size in bytes is ACCESS_SIZE * NUMBER_OF_BEATS).

VADDR unsigned int

The virtual address of the access.

VFP_D_REGS

VFP/NEON D 64 bit register write. Fields:

ALIASING enum

Type of register aliasing used.

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a write to the corresponding bit in the register. For example, 0x1 would indicate a write to VALUE[0].

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

VFP_D_REGS_READ

VFP/NEON D 64 bit register read. Fields:

ID unsigned int

The register number.

VALUE unsigned int

The current value of the register.

VFP_Q_REGS

VFP/NEON Q 128 bit register write. Fields:

ALIASING enum

Type of register aliasing used.

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a write to the corresponding byte in the register. For example, 0x1 would indicate a write to VALUE[7:0].

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

VFP_Q_REGS_READ

VFP/NEON Q 128 bit register read. Fields:

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a read to the corresponding byte in the register. For example, 0x1 would indicate a read to VALUE[7:0].

VALUE unsigned int

The current value of the register.

VFP_SYS_REGS

Writes to the VFP/NEON units system registers. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ID enum

Which VFP system register is written.

MASK unsigned int

The mask indicating updated cumulative bits.

OLD_VALUE unsigned int

The register's old value overwritten.

VALUE unsigned int

The new value written to the VFP system register.

VFP_S_REGS

VFP/NEON S 32 bit register write. Fields:

ALIASING enum

Type of register aliasing used.

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a write to the corresponding bit in the register. For example, 0x1 would indicate a write to VALUE[0].

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

VFP_S_REGS_READ

VFP/NEON S 32 bit register read. Fields:

ID unsigned int

The register number.

VALUE unsigned int

The current value of the register.

WAYPOINT

Signals end of basic block execution. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ISSET enum

Origin instruction set.

IS_COND bool

Indicates if this is a conditional waypoint.

PC unsigned int

Origin address (or 0 if unavailable).

TAKEN bool

Indicates if this waypoint was taken.

TARGET unsigned int

Destination address.

TARGET_ISET enum

Destination instruction set.

WFE_END

WFE ended. Fields:

INST_COUNT unsigned int

Ticks count when leaving WFE.

WFE_EVENT_REGISTER

WFE event register status: set/clear, reason. Fields:

INST_COUNT unsigned int

Ticks count.

REASON enum

Reason for set/clear. Only REASON==0 (Cleared by WFE) clears the bit, all other reasons set it.

WFE_IGNORED

WFE ignored. Fields:

EVENT bool

This WFE was ignored because the event register was set.

INST_COUNT unsigned int

Ticks count when ignoring WFE.

TRAPPED bool

This WFE was trapped.

WFE_START

WFE entered. Fields:

INST_COUNT unsigned int

Ticks count when entering WFE.

WFI_END

WFI ended. Fields:

INST_COUNT unsigned int

Ticks count when leaving WFI.

WFI_IGNORED

WFI ignored. Fields:

DISABLED bool

This WFI was ignored because WFI is disabled.

INST_COUNT unsigned int

Ticks count when ignoring WFI.

REASON enum

specifies reason why WFI trace was ignored.

TRAPPED bool

This WFI was trapped.

WFI_START

WFI entered. Fields:

INST_COUNT unsigned int

Ticks count when entering WFI.

WFI_WAKEUP

WFI wakeup. Fields:

INST_COUNT unsigned int

Ticks count when WFI wakeup occurred.

REASON enum

Reason for wakeup.

xPSR

Changes to the xPSR register. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

OLD_VALUE unsigned int

The old xPSR value.

UNKNOWN unsigned int

Bits within the register that have unknown value.

VALUE unsigned int

The new xPSR value.

2.44 ARM_Cortex-M4

This section describes the trace sources.

ASYNC_MEMORY_FAULT

Context ID Register write. Fields:

FAULT unsigned int

Fault status present in ESR.ISS encoding for Serror in ARCH64 and present in DFSR encoding for external abort format in ARCH32.

PADDR unsigned int

Physical Address (or 0 if unavailable).

ATOMIC_END_ACCESS

Bus trace access for atomics. Fields:

ACCESS_FAIL bool

Memory access failed.

ADDR unsigned int

The virtual address of the access.

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

COMPARE_VALUE unsigned int

Compare value for CAS.

LOAD_VALUE unsigned int

Loaded values.

NSDESC unsigned int

The security state of the access.

OPERAND_VALUE unsigned int

Operation's operand.

OPERATION enum

Operation type.

PADDR unsigned int

The physical address of the access.

PRIV bool

Is this a privileged access?.

ATOMIC_START_ACCESS

Bus trace access for atomics. Fields:

ADDR unsigned int

The virtual address of the access.

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory

attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

COMPARE_VALUE unsigned int

Compare value for CAS.

NSDESC unsigned int

The security state of the access.

OPERAND_VALUE unsigned int

Operation's operand.

OPERATION enum

Operation type.

PADDR unsigned int

The physical address of the access.

PRIV bool

Is this a privileged access?.

ArchMsg.Error.exit_code

DISPLAY exit code: %{EXITCODE} component kind: %{KIND} reason: %{REASON}. Fields:

EXITCODE signed int

Exit code to be returned.

KIND string

Component kind that invoked the exit code trace.

REASON string

Reason why the exit code trace was invoked (optional).

ArchMsg.Warning.dap_csw_bad_size

A write to CM3DAP CSW has an invalid size field. DISPLAY Write %{DATA} to CM3DAP CSW has invalid size field. Fields:

DATA unsigned int

bits[2:0] are size.

ArchMsg.Warning.recursive_branch

An instruction is performing a branch that targets the same instruction. If this was intended then a WFI might be more effective DISPLAY recursive branch detected at PC=%{PC}. PRIMARY KEY PC. Fields:

PC unsigned int

Address of instruction causing the branch.

ArchMsg.Warning.recursive_exception

An instruction has generated an exception that targets the same instruction. DISPLAY recursive %{TYPE} exception detected at PC=%{PC} (CPSR=%{CPSR}). PRIMARY KEY PC. Fields:

CPSR unsigned int

Processor state.

PC unsigned int

Address of instruction causing the exception.

TYPE enum

Type of exception.

ArchMsg.Warning.secure_vector_fetch_from_nonsecure

DISPLAY Secure vector table is being fetched from Non-secure memory.

ArchMsg.Warning.sg_in_it_block

Arm recommends that software does not place SG (security gate instruction) inside an IT instruction block.

ArchMsg.warning_unpred_system_register_access

DISPLAY Unpredictable %{IS_WRITE:(read|write)} access to system register offset %{OFFSET}. Fields:

IS_WRITE unsigned int

Write Not Read.

OFFSET unsigned int

Register Offset.

BRANCH_MISPREDICT

Simulating branch mispredict. Fields:

PC unsigned int

Origin address (or 0 if unavailable).

BRA_DIR

Direct branches, to immediate address. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISET enum

The instruction set of the branch instruction.

IS_COND bool

Indicates if this is a conditional branch.

IS_HINTED bool

Whether the conditional instruction is hinted.

IS_ISB bool

Whether the instruction is ISB.

IS_LINK bool

Indicates if this is a branch with link.

PC unsigned int

The address of the branch instruction.

TARGET_ISET enum

The instruction set after the branch.

TARGET_PC unsigned int

The address the instruction branches to.

BRA_INDIR

Indirect branches, perhaps to a register. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISET enum

The instruction set of the branch instruction.

IS_COND bool

Indicates if this is a conditional branch.

IS_HINTED bool

Whether the conditional instruction is hinted.

IS_ISB bool

Whether the instruction is ISB.

IS_LINK bool

Indicates if this is a branch with link.

PC unsigned int

The address of the branch instruction.

TARGET_ISET enum

The instruction set after the branch.

TARGET_PC unsigned int

The address the instruction branches to.

BRA_INDIR_CCFAIL

Indirect branches fail, perhaps to a register.

BRA_RET_CCFAIL

Branche return fail, perhaps to a register.

CCFAIL

Conditional instruction condition check fail. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CCFAIL_UNC

Unconditional instruction condition check fail. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CCPASS

Conditional instruction condition check pass. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CCPASS_UNC

Unconditional instruction condition check pass. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CODE_CACHE_MAINT

Code cache maintenance. Fields:

END_INCL unsigned int

Inclusive end address for VA or PA requests.

NS bool

Non-secure world for PA.

START unsigned int

Start address for VA or PA requests.

TYPE enum

Request type.

COMPILE_BLOCK_END

Last instruction of basic block translated. Fields:

VADDR unsigned int

Address of next instruction after this basic block.

COMPILE_BLOCK_START

First instruction of basic block translated. Fields:

VADDR unsigned int

Address of first instruction in basic block.

COMPILE_INST

ARM instruction compiled. Fields:

DISASS string

Disassembly of the instruction.

ISSET enum

The instruction set of this instruction.

ITSTATE unsigned int

The ITSTATE current for the instruction.

OPCODE unsigned int

The opcode of the instruction.

PC unsigned int

The address of the instruction.

SIZE unsigned int

The size of the instruction in bytes.

CONTEXTIDR

Context ID Register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

NS bool

Secure or nonsecure banked register is accessed.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The new value written.

CONTEXT_SYNC

Called for every context synchronization event. Fields:

ARCH_CSE bool

Architectural Context Synchronization Event.

PC unsigned int

The address of the synchronization instruction.

SYSREG_SYNC bool

System register synchronization event.

CORE_ENDIAN

Core BE8 Big-Endian state changed. Fields:

BE8 bool

Core BE8 Big-Endian state.

CORE_INFO

Static processor attributes. Only triggered by a call to DumpState(). Fields:

ARCH_PROFILE enum

The architecture profile of the core.

CLUSTER_ID unsigned int

The cluster ID of this processor.

CORE_NUM unsigned int

The number of this core in an MP processor.

DETAILED_CORE_FEATURES enum

Detailed core features as a bitfield. Enum values are bitmasks to extract the info from the field.

FPU_VERSION enum

The VFP version implemented by the core.

MEM_ARCH enum

The memory architecture of the core.

MIDR_EL1 unsigned int

MIDR_EL1 value.

NUM_CORES unsigned int

The number of cores in this MP processor.

QUANTUM_SIZE unsigned int

The default quantum size of the core (in ticks).

SECURITY_FEATURES bool

Does the core have security features?.

CORE_LOADS

Processor load accesses. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ACQREL enum

Is this an acquire/release.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DATA unsigned int

The data read or written.

ELEMENT_SIZE unsigned int

Width of each element.

ENDIANNESS enum

Byte ordering of this access.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

LOCK enum

Normal, exclusive or locked access.

MECID unsigned int

Memory Encryption Context ID, if applicable.

MECID2 unsigned int

Memory Encryption Context ID of the second page, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PAGE_SIZE unsigned int

Page size as $\log_2(\text{size in bytes})$.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK, 2=Exclusive Failed, 3=Unsupported.

SIZE unsigned int

Width of the access in bytes. Only required if DATA is not traced.

TRANS bool

Is this a translated access.

VADDR unsigned int

The virtual address of the access.

CORE_REGS

Changes of the core registers R0 to R14. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number, 0 to 14.

OLD_VALUE unsigned int

The old value overwritten.

PHYS_ID enum

The physical register accessed.

VALUE unsigned int

The new value written to the register.

CORE_STALLED_TRANSITION

Detect when a core is prevented from progressing PChannel states. Fields:

CURRENT_STATE enum

Current power mode of core.

REASON string

The reason a transition was stalled, and what must be done to unblock it.

TARGET_STATE enum

Desired power mode of core after transition.

CORE_STORES

Processor store accesses. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ACQREL enum

Is this an acquire/release.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DATA unsigned int

The data read or written.

ELEMENT_SIZE unsigned int

Width of each element.

ENDIANNESS enum

Byte ordering of this access.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

LOCK enum

Normal, exclusive or locked access.

MECID unsigned int

Memory Encryption Context ID, if applicable.

MECID2 unsigned int

Memory Encryption Context ID of the second page, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PAGE_SIZE unsigned int

Page size as log2(size in bytes).

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK, 2=Exclusive Failed, 3=Unsupported.

SIZE unsigned int

Width of the access in bytes. Only required if DATA is not traced.

TRANS bool

Is this a translated access.

VADDR unsigned int

The virtual address of the access.

CRYPTO_SPEC

Every crypto instruction speculatively executed.

DATA_CACHE_SET_ALLOC_TAGS

Set allocation tags from DC GVA instruction. Fields:

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate

when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

NSDESC unsigned int

The security state of the access.

PADDR unsigned int

The physical address of the access.

PAS enum

The physical address space of the page.

RESPONSE enum

0=Aborted, 1=OK.

SIZE unsigned int

The size of memory in bytes.

TAG unsigned int

The value of the tag being set.

TAGGING_TYPE enum

Type of Memory tagging (Physical or Virtual).

TAG_VADDR unsigned int

The virtual address of the Virtual Tag if Virtual tagging is used; otherwise 0.

VADDR unsigned int

Virtual address used for setting allocation tags.

DATA_CACHE_ZERO

Zero and invalidate cache line from DC ZVA instruction. Fields:

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

NSDESC unsigned int

The security state of the access.

PADDR unsigned int

The physical address of the access.

PADDR_TAG unsigned int

If tag checked, the physical address tag for the access.

PAS enum

The physical address space of the page.

RESPONSE enum

0=Aborted, 1=OK.

SIZE unsigned int

The size of zero'd memory in bytes.

TAG_CHECKED bool

Is this access tag checked.

VADDR unsigned int

Virtual address of the zero'd cache line.

DEBUG_EVENT

Hardware debug support event. Fields:

EVENT enum

Description of event.

VALUE unsigned int

data value.

DMI_ALLOCATE

DMI allocated. Fields:

ACCESS_DIRECTION enum

READ / WRITE.

PADDR unsigned int

The physical address of the DMI.

PTR unsigned int

DMI range pointer.

SIDE enum

DSIDE / ISIDE.

SIZE unsigned int

The size of DMI.

VADDR unsigned int

The virtual address of the DMI..

DMI_HIT

DMI hit, access used DMI (only fire for small DMI regions i.e. less than a full page). Fields:

ACCESS_DIRECTION enum

READ / WRITE.

PADDR unsigned int

The physical address of the DMI.

SIDE enum

DSIDE / ISIDE.

SIZE unsigned int

The size of the DMI.

VADDR unsigned int

The virtual address of the DMI..

DMI_REVOKE

DMI revoked. Fields:

ACCESS_DIRECTION enum

READ / WRITE.

PADDR unsigned int

The physical address of the DMI.

SIDE enum

DSIDE / ISIDE.

SIZE unsigned int

The size of DMI.

VADDR unsigned int

The virtual address of the DMI..

DWT_MATCH

DWT comparator matches. Fields:

NUM unsigned int

DWT comparator number.

TYPE enum

DWT comparator configuration.

END_COMPILE

Compilation end. Fields:

END_OF_PAGE_COUNT unsigned int

Number of basic blocks exited due to page boundary since START_COMPILE.

FETCHFAIL_COUNT unsigned int

Number of basic blocks exited due to ifetch failure START_COMPILE.

INST_COUNT unsigned int

Number of instructions compiled since START_COMPILE.

NONSEQ_COUNT unsigned int

Number of basic blocks exited due to non-sequential instructions since START_COMPILE.

PAGE_STRADDLE_COUNT unsigned int

Number of basic blocks exited due to unaligned instructions crossing page since START_COMPILE.

EXCEPTION

Exceptions that are taken. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ISET enum

The instruction set of the processor when the exception occurred.

LR unsigned int

The value assigned to the link register.

PC unsigned int

The location where the exception occurred.

TARGET_ISET enum

The instruction set of the exception handler code.

TARGET_PC unsigned int

The address the exception branches to.

VECTOR enum

The exception vector.

EXCEPTION_END

Every exception completed.

EXCEPTION_ENTRY

Event marking the entry of an exception. All pushes of registers to the stack will follow this event. Fields:

PC unsigned int

The location where the exception occurred.

VECTOR enum

The exception that occurred.

EXCEPTION_RAISE

Every exception raised.

EXCEPTION_RETURN

Marks the end of an exception, but on an M core does not indicate a program flow change (branch). In most situations it will be followed by an indirect branch. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISET enum

The instruction set of the branch instruction.

PC unsigned int

The address of the branch instruction.

EXCEPTION_RETURN_PREBRANCH

Exception return event that must occur before PC and processor state have updated.

EXCEPTION_START

Every exception started.

EXCEPTION_VECTOR_FETCH

I-side vector fetch for M-class exception. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

DATA unsigned int

Result of fetch if OK.

RESPONSE enum

0=Aborted, 1=OK.

VADDR unsigned int

Where the vector is fetched from (including VTOR).

FP_STATE

Floating point state. Fields:

FPSR unsigned int

FPSR register value.

MASK unsigned int

The mask indicating updated cumulative bits.

FREQ_CHANGED

Core frequency changed. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The instruction count of this core.

NEW_FREQ signed int

The new frequency of this core (expressed in Hz).

OLD_FREQ signed int

The old frequency of this core (expressed in Hz).

HLT

HLT instruction occurred. Fields:

IMM16 unsigned int

The 16-bit immediate value encoded in HLT instruction.

INFO_EXCEPTION_REASON

Provide information of real cause of exception. Fields:

FaultCause enum

Which bit of which V7M/V8M-mainline FSR/HFSR would have been set.

PC unsigned int

The location where the exception occurred.

PHASE enum

What we are doing with the exception.

REASONS enum

Why we are doing it.

VECTOR enum

The exception that occurred.

INFO_FP_CONTEXT

Inform when ExecuteFPCheck writes CONTROL.FPCA and FPSCR etc. Fields:

WHAT enum

What we are doing.

INFO_STACKING

Provide information of real cause of exception. Fields:

PHASE enum

Are we starting or ending a sequence of stack writes.

REASONS enum

Why we are doing it.

INST

Every instruction executed. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DEBUG_STATE bool

The instruction is executed in debug state.

DISASS string

Disassembly of instruction.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISSET enum

The current instruction set.

ITSTATE unsigned int

The current ITSTATE.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

MODE enum

The mode the core is in.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

OPCODE unsigned int

The opcode of the instruction.

PADDR unsigned int

The physical address of the instruction.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the instruction.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the page.

PC unsigned int

The address of the instruction.

SECURITY_STATE enum

The core's security state.

SIZE unsigned int

The size of the instruction in bytes.

INST_END

Every instruction completed.

INST_START

Every instruction started. Fields:

DEBUG_STATE bool

The instruction is executed in debug state.

ISSET enum

The current instruction set.

MODE enum

The mode the core is in.

NS enum

The current Secure State.

PC unsigned int

The address of the conditional instruction.

SECURITY_STATE enum

The current Security State.

INST_STRADDLE

Instruction straddles boundary. Fields:

DEBUG_STATE bool

The instruction is executed in debug state.

ISET enum

The current instruction set.

MODE enum

The mode the core is in.

NS enum

The current Secure State.

PC unsigned int

The address of the conditional instruction.

SECURITY_STATE enum

The current Security State.

ISB

ISB instruction occurred. Fields:

PC unsigned int

The address of the ISB instruction.

ITM

Instrumentation Trace Macrocell. Fields:

ITM_PACKET_TYPE enum

ITM and DWT packets type.

PACKET_HEADER unsigned int

ITM Packet Header.

PACKET_PAYLOAD unsigned int

ITM Packet Payload.

LOAD_MULTIPLE_REGS_LIST

List of destination registers (not source) of a multiple registers load instruction. Fields:

REG string

Register to add into the list.

LOCAL_MONITOR

Local monitor activity. Fields:

PADDR unsigned int

Local Monitor Address.

State enum

State of the monitor (Open/Exclusive).

LOCKUP_CYCLE

This event is triggered if the core enters or stays in the lock-up state. Fields:

CAUSE enum

What causes the lockup?.

LOCKUP_ENTRY

This event is triggered if the core enters the lock-up state. Fields:

PC unsigned int

The PC when the lockup state is entered.

LOCKUP_EXIT

This event is triggered if the core leaves the lock-up state.

MODE_CHANGE

Mode change. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

MODE enum

The new mode.

OLD_MODE enum

The old mode.

MPU_TRANS

Address translation information. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

EXEC_PERM enum

Execution Permission.

INNERCACHE_RA bool

Is the inner cache allocate on read.

INNERCACHE_TYPE enum

Inner Caching scheme (NC/MB/WA).

INNERCACHE_WA bool

Is the inner cache allocate on write.

MEMTYPE enum

Memory type.

OUTERCACHE_RA bool

Is the outer cache allocate on read.

OUTERCACHE_TYPE enum

Outer Caching scheme (NC/MB/WA).

OUTERCACHE_WA bool

Is the outer cache allocate on write.

PADDR unsigned int

Address of the access.

PAGESIZE unsigned int

Page size as $\log_2(\text{size})$.

READ_PERM enum

Read Permission.

REG_NUM unsigned int

MPU register based region-number 0-15 (else 0xFF=none, 0xBB/0xDD=background/default).

SH enum

Shareability.

SIDE enum

Inst / Data.

WRITE_PERM enum

Write Permission.

PERIODIC

Called for every quantum. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The instruction count of this CPU.

PC unsigned int

The address of the next instruction to be executed on this CPU.

POTINDEX_UPDATE

Value of POT index for current pc changed. Fields:

NEW_POTINDEX unsigned int

New POTIndex.

OLD_POTINDEX unsigned int

Old POTIndex.

PPB_READ_ACCESS

Trace reads from the core to memory mapped registers in the PPB (private peripheral bus) address range. Fields:

ACCESS_SIZE unsigned int

Logarithm of the access size: 0=byte, 1=halfword, 2=word, 3=doubleword, ...

BURST_LEN unsigned int

The number of data transfers (beats) in this burst.

DATA unsigned int

The data transferred.

INNER_ATTRIBUTES unsigned int

The inner attributes: bit0=write allocate, bit1=read allocate, bit2=cachable, bit3=bufferable, bit4=shareable.

INST bool

Is an instruction fetch.

LOCK enum

Normal / Locked / Exclusive.

MANAGER_ID unsigned int

The AXI manager ID.

NS bool

Is a nonsecure access.

OUTER_ATTRIBUTES unsigned int

The outer attributes: bit0=write allocate, bit1=read allocate, bit2=cachable, bit3=bufferable, bit4=shareable.

PADDR unsigned int

Physical address of access.

PRIV bool

Is privileged access.

RESPONSE enum

Whether the transaction was successful, or an error occurred.

USER_FLAGS unsigned int

Core specific additional signals.

PPB_WRITE_ACCESS

Trace writes from the core to memory mapped registers in the PPB (private peripheral bus) address range. Fields:

ACCESS_SIZE unsigned int

Logarithm of the access size: 0=byte, 1=halfword, 2=word, 3=doubleword, ...

BURST_LEN unsigned int

The number of data transfers (beats) in this burst.

DATA unsigned int

The data transferred.

INNER_ATTRIBUTES unsigned int

The inner attributes: bit0=write allocate, bit1=read allocate, bit2=cachable, bit3=bufferable, bit4=shareable.

INST bool

Is an instruction fetch.

LOCK enum

Normal / Locked / Exclusive.

MANAGER_ID unsigned int

The AXI manager ID.

NS bool

Is a nonsecure access.

OUTER_ATTRIBUTES unsigned int

The outer attributes: bit0=write allocate, bit1=read allocate, bit2=cachable, bit3=bufferable, bit4=shareable.

PADDR unsigned int

Physical address of access.

PRIV bool

Is privileged access.

RESPONSE enum

Whether the transaction was successful, or an error occurred.

USER_FLAGS unsigned int

Core specific additional signals.

PRELOAD_DATA

Data preload from PLD instruction. Fields:

VADDR unsigned int

Virtual address of the data that should be preloaded.

PRELOAD_INST

Instruction preload from PLI instruction. Fields:

VADDR unsigned int

Virtual address of the instruction that should be preloaded.

PRE_CORE_LOAD

Trace just before a core load. Fields:

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

PRE_CORE_STORE

Trace just before a core store. Fields:

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

PSTATE_PACTIVE_OPMODE_TRANSITION

Trace pactive and Opmode mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Operating mode after the transition.

OLD_STATE enum

Operating mode before the transition.

PSTATE_PACTIVE_OPMODE_TRANSITION_DENIED

Trace the denied pactive and Opmode mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Operating mode after the transition.

OLD_STATE enum

Operating mode before the transition.

REASON string

Reason for the transition being denied.

PSTATE_PACTIVE_POWER_TRANSITION

Trace pactive and power mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Power mode after the transition.

OLD_STATE enum

Power mode before the transition.

PSTATE_PACTIVE_POWER_TRANSITION_DENIED

Trace the denied pactive and power mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Power mode after the transition.

OLD_STATE enum

Power mode before the transition.

REASON string

Reason for the transition being denied.

PSTATE_PACTIVE_THREAD_OPMODE_TRANSITION

Trace pactive and Opmode mode transitions for threads on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Operating mode after the transition.

OLD_STATE enum

Operating mode before the transition.

QUANTUM_START

Called at the start of every quantum.

RUN_STATE

Run state transition. Fields:

INST_COUNT unsigned int

Ticks count at point of transition.

NEW enum

New run state.

OLD enum

Old run state.

SEMIHOSTING_CALL

Call of a semihost function occurred. Fields:

PC unsigned int

The program counter after the semihosting call.

SEMIHOSTING_PRECALL

About to call semihost. Fields:

EL2 bool

Use EL2 translation regime.

NS bool

Is Non-Secure.

REG_WIDTH unsigned int

The current register width in bytes.

SIGNAL

External signal state change. Fields:

SIGNAL enum

Signal that changed.

STATE bool

Signal asserted state.

START_COMPILE

Compilation started. Fields:

VADDR unsigned int

Instruction where compilation begins.

STG_MULTIPLE_MICRO_OPS_ADDRESS_RANGE

Address range for MTE stores with multiple uops. Fields:

ADDR_END unsigned int

Final address.

ADDR_START unsigned int

Starting address.

STORE_MULTIPLE_REGS_ADDRESS_RANGE

Address range for stores with multiple registers. Fields:

ADDR_END unsigned int

Final address.

ADDR_START unsigned int

Starting address.

SUBOPTIMAL_LDST_RETIRED

suboptimal load/store. Fields:

ACCESS_SIZE unsigned int

Log2 of the access width in bytes used for alignment checking (1: half-word, 2: word ...).

OPTIMAL_ALIGN_SELECTOR unsigned int

Selector to decide the optimal alignment size (1: access width, 2: 4-bytes, 3: 8-bytes, ..., 12: 4-kbytes).

VADDR unsigned int

The virtual address of the access.

SYNC

Called for every synchronization, that is every quantum break. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The instruction count of this CPU.

LOCAL_QUANTUM unsigned int

The local quantum of this CPU.

LOCAL_TIME unsigned int

The local time of this CPU.

SYNC_REASON enum

Reason for sync.

SYSCALL

System call instruction executed. Fields:

IMM unsigned int

Immediate value of the system call instruction.

TYPE enum

System call type.

VADDR unsigned int

Instruction that caused the system call.

SYSREG_READ

Trace reads from system registers. Fields:

REG_NAME enum

Register number.

VALUE unsigned int

The value read.

SYSREG_UPDATE128

Triggers when the system updates a register. Fields:

REG enum

Register number.

UNKNOWN unsigned int

Bits of the register which became unknown.

VALUE unsigned int

Value written to the register.

SYSREG_UPDATE32

Triggers when the system updates a register. Fields:

REG enum

Register number.

UNKNOWN unsigned int

Bits of the register which became unknown.

VALUE unsigned int

Value written to the register.

SYSREG_UPDATE64

Triggers when the system updates a register. Fields:

REG enum

Register number.

UNKNOWN unsigned int

Bits of the register which became unknown.

VALUE unsigned int

Value written to the register.

SYSREG_WRITE

Trace write to system registers. Fields:

REG_ACCESS enum

Register access status.

REG_NAME enum

Register number.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

Value written to the register.

UNALIGNED_LDST_RETIRED

Processor unaligned load/store. Fields:

ACCESS_SIZE unsigned int

Log2 of the access width in bytes used for alignment checking (1=>half-word, 2->word ...).

NUMBER_OF_BEATS unsigned int

Number of accesses (beats) in this burst (Total size in bytes is ACCESS_SIZE * NUMBER_OF_BEATS).

VADDR unsigned int

The virtual address of the access.

VFP_D_REGS

VFP/NEON D 64 bit register write. Fields:

ALIASING enum

Type of register aliasing used.

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a write to the corresponding bit in the register. For example, 0x1 would indicate a write to VALUE[0].

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

VFP_D_REGS_READ

VFP/NEON D 64 bit register read. Fields:

ID unsigned int

The register number.

VALUE unsigned int

The current value of the register.

VFP_Q_REGS

VFP/NEON Q 128 bit register write. Fields:

ALIASING enum

Type of register aliasing used.

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a write to the corresponding byte in the register. For example, 0x1 would indicate a write to VALUE[7:0].

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

VFP_Q_REGS_READ

VFP/NEON Q 128 bit register read. Fields:

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a read to the corresponding byte in the register. For example, 0x1 would indicate a read to VALUE[7:0].

VALUE unsigned int

The current value of the register.

VFP_SYS_REGS

Writes to the VFP/NEON units system registers. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ID enum

Which VFP system register is written.

MASK unsigned int

The mask indicating updated cumulative bits.

OLD_VALUE unsigned int

The register's old value overwritten.

VALUE unsigned int

The new value written to the VFP system register.

VFP_S_REGS

VFP/NEON S 32 bit register write. Fields:

ALIASING enum

Type of register aliasing used.

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a write to the corresponding bit in the register. For example, 0x1 would indicate a write to VALUE[0].

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

VFP_S_REGS_READ

VFP/NEON S 32 bit register read. Fields:

ID unsigned int

The register number.

VALUE unsigned int

The current value of the register.

WAYPOINT

Signals end of basic block execution. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ISET enum

Origin instruction set.

IS_COND bool

Indicates if this is a conditional waypoint.

PC unsigned int

Origin address (or 0 if unavailable).

TAKEN bool

Indicates if this waypoint was taken.

TARGET unsigned int

Destination address.

TARGET_ISET enum

Destination instruction set.

WFE_END

WFE ended. Fields:

INST_COUNT unsigned int

Ticks count when leaving WFE.

WFE_EVENT_REGISTER

WFE event register status: set/clear, reason. Fields:

INST_COUNT unsigned int

Ticks count.

REASON enum

Reason for set/clear. Only REASON==0 (Cleared by WFE) clears the bit, all other reasons set it.

WFE_IGNORED

WFE ignored. Fields:

EVENT bool

This WFE was ignored because the event register was set.

INST_COUNT unsigned int

Ticks count when ignoring WFE.

TRAPPED bool

This WFE was trapped.

WFE_START

WFE entered. Fields:

INST_COUNT unsigned int

Ticks count when entering WFE.

WFI_END

WFI ended. Fields:

INST_COUNT unsigned int

Ticks count when leaving WFI.

WFI_IGNORED

WFI ignored. Fields:

DISABLED bool

This WFI was ignored because WFI is disabled.

INST_COUNT unsigned int

Ticks count when ignoring WFI.

REASON enum

specifies reason why WFI trace was ignored.

TRAPPED bool

This WFI was trapped.

WFI_START

WFI entered. Fields:

INST_COUNT unsigned int

Ticks count when entering WFI.

WFI_WAKEUP

WFI wakeup. Fields:

INST_COUNT unsigned int

Ticks count when WFI wakeup occurred.

REASON enum

Reason for wakeup.

XPSR

Changes to the xPSR register. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

OLD_VALUE unsigned int

The old xPSR value.

UNKNOWN unsigned int

Bits within the register that have unknown value.

VALUE unsigned int

The new xPSR value.

2.45 ARM_Cortex-M52

This section describes the trace sources.

ASYNC_MEMORY_FAULT

Context ID Register write. Fields:

FAULT unsigned int

Fault status present in ESR.ISS encoding for Serror in ARCH64 and present in DFSR encoding for external abort format in ARCH32.

PADDR unsigned int

Physical Address (or 0 if unavailable).

ATOMIC_END_ACCESS

Bus trace access for atomics. Fields:

ACCESS_FAIL bool

Memory access failed.

ADDR unsigned int

The virtual address of the access.

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

COMPARE_VALUE unsigned int

Compare value for CAS.

LOAD_VALUE unsigned int

Loaded values.

NSDESC unsigned int

The security state of the access.

OPERAND_VALUE unsigned int

Operation's operand.

OPERATION enum

Operation type.

PADDR unsigned int

The physical address of the access.

PRIV bool

Is this a privileged access?.

ATOMIC_SLAVE_ACCESS

Information on the atomic operation at slave side. Fields:

ACCESS_RESULT unsigned int

Result of atomic access.

COMPARE_VALUE unsigned int

Value to be compared with memory by CAS.

LOAD_VALUE unsigned int

Value to be loaded from memory as a result of atomic operation.

MANAGER unsigned int

ID of manager which initiated the transaction.

NS bool

Transaction is Non-Secure.

OPERAND_VALUE unsigned int

Incoming operand value provided by core.

OPERATION enum

Type of atomic operation.

PADDR unsigned int

Full physical address of TCM.

STORE_VALUE unsigned int

Value to be stored to memory as a result of atomic operation.

ATOMIC_START_ACCESS

Bus trace access for atomics. Fields:

ADDR unsigned int

The virtual address of the access.

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

COMPARE_VALUE unsigned int

Compare value for CAS.

NSDESC unsigned int

The security state of the access.

OPERAND_VALUE unsigned int

Operation's operand.

OPERATION enum

Operation type.

PADDR unsigned int

The physical address of the access.

PRIV bool

Is this a privileged access?.

ArchMsg.Error.exit_code

DISPLAY exit code: %{EXITCODE} component kind: %{KIND} reason: %{REASON}. Fields:

EXITCODE signed int

Exit code to be returned.

KIND string

Component kind that invoked the exit code trace.

REASON string

Reason why the exit code trace was invoked (optional).

ArchMsg.Warning.Unpredictably Indexed PM Event RegisterTAGS **UNPREDICTABLE** DISPLAY Event %{IsDirect:(Selector value|Register)} %{SEL} is >= %{N}. Fields:**IsDirect unsigned int**

Direct Access rather than indirect.

N unsigned int

Number of Accessible Registers.

SEL unsigned int

Selector.

ArchMsg.Warning.cache_contents_unknown

A cache has been activated without being invalidated and may contain **UNKNOWN** content. DISPLAY %{SIDE:(D|I)}-Cache was enabled in %{REGIME:(SECURE|NON_SECURE|HYP)} regime but was not invalidated since power-on: cache lines could contain **UNKNOWN** content %{SIDE: (Note: D-side caches must invalidate SECURE lines even if the cache is only used by NON_SECURE code in order to ensure that unexpected dirty lines tagged SECURE are not naturally evicted)}. Fields:

REGIME unsigned int

Current translation regime.

SIDE unsigned int

0 d-side cache or unified cache, 1 i-side.

ArchMsg.Warning.dap_csw_bad_size

A write to CM3DAP CSW has an invalid size field. DISPLAY Write %{DATA} to CM3DAP CSW has invalid size field. Fields:

DATA unsigned int

bits[2:0] are size.

ArchMsg.Warning.dcimvac_matches_watchpoint

TAGS UNPREDICTABLE DISPLAY Watchpoints matching an AArch32 DCIMVA are implementation defined.

ArchMsg.Warning.decode_fieldmismatch

TAGS UNPREDICTABLE DISPLAY Instruction is unpredictable with different fields %{FIELD1} and %{FIELD2}. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_frachitsnegative

TAGS UNPREDICTABLE DISPLAY VCVT Instruction is unpredictable with a negative number of fraction bits.

ArchMsg.Warning.decode_initblock

TAGS UNPREDICTABLE DISPLAY Instruction is unpredictable in an IT block.

ArchMsg.Warning.decode_initblocknotlast

TAGS UNPREDICTABLE DISPLAY Instruction is unpredictable in an IT block when not the last.

ArchMsg.Warning.decode_invalid_condition

TAGS UNPREDICTABLE DISPLAY Instruction is unpredictable when the condition is not AL.

ArchMsg.Warning.decode_invalidfieldcombination

TAGS UNPREDICTABLE DISPLAY Instruction is unpredictable with the values in %{FIELD1} and %{FIELD2}. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_invalidvalue

TAGS UNPREDICTABLE DISPLAY Instruction is unpredictable with the value in %{FIELD}. Fields:

FIELD enum

Field name.

ArchMsg.Warning.decode_mem_hint_unallocated

TAGS UNPREDICTABLE DISPLAY There is no memory hint allocated to this bit pattern.

ArchMsg.Warning.decode_registermatch

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with %{FIELD1} and %{FIELD2} as the same register. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_registermismatch

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with %{FIELD1} and %{FIELD2} as different registers. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_registeroutofrange

TAGS **UNPREDICTABLE** DISPLAY Floating-point instruction is unpredictable indexing beyond the end of the register bank.

ArchMsg.Warning.decode_sbzsbo

TAGS **UNPREDICTABLE** DISPLAY Reserved bits in the instruction are not canonical.

ArchMsg.Warning.decode_transactiontoobig

TAGS **UNPREDICTABLE** DISPLAY Attempt to perform a memory transaction of more than 128 bytes.

ArchMsg.Warning.decode_unpred_other

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable.

ArchMsg.Warning.decode_unpreduseofpc

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable when %{FIELD} is PC. Fields:

FIELD enum

Field name.

ArchMsg.Warning.decode_unpreduseofr13

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable when %{FIELD} is R13. Fields:

FIELD enum

Field name.

ArchMsg.Warning.decode_unpreduseofr13inreglist

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with R13 in register-list.

ArchMsg.Warning.decode_useofd16orhigher

TAGS **UNPREDICTABLE** DISPLAY Attempt to use register D16 or higher where it is not allowed.

ArchMsg.Warning.decode_writebackandbaseinlist

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with write back when the base register is in the transfer list.

ArchMsg.Warning.decode_zeroregistersinlist

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable when no registers are to be transferred.

ArchMsg.Warning.extension_unpred

Probably unpred in IT state or decod. Fields:

PC unsigned int

Address of this instruction.

ArchMsg.Warning.msr_zeroes_ltpsize

May be pre-MVE setup code.

ArchMsg.Warning.pmu_and_dwt

DISPLAY Trying to access a %{IS_PMU:(DWT|PMU)} register while %{IS_PMU:(PMU|DWT)} is also active. Fields:

IS_PMU bool

Whether trying to access the PMU or the DWT.

ArchMsg.Warning.recursive_branch

An instruction is performing a branch that targets the same instruction. If this was intended then a WFI might be more effective DISPLAY recursive branch detected at PC=%{PC}. PRIMARY KEY PC. Fields:

PC unsigned int

Address of instruction causing the branch.

ArchMsg.Warning.recursive_exception

An instruction has generated an exception that targets the same instruction. DISPLAY recursive %{TYPE} exception detected at PC=%{PC} (CPSR=%{CPSR}). PRIMARY KEY PC. Fields:

CPSR unsigned int

Processor state.

PC unsigned int

Address of instruction causing the exception.

TYPE enum

Type of exception.

ArchMsg.Warning.reentrant_vector_catch

TAGS **UNPREDICTABLE** DISPLAY Vector catch debug event on Prefetch or Data abort vector %{ADDR}. Fields:

ADDR unsigned int

vector address.

ArchMsg.Warning.reserved_it_state

TAGS **UNPREDICTABLE** DISPLAY Execution with reserved IT state %{ITSTATE}. Fields:

ITSTATE unsigned int
reserved IT state.

ArchMsg.Warning.secure_vector_fetch_from_nonsecure

DISPLAY Secure vector table is being fetched from Non-secure memory.

ArchMsg.Warning.sg_in_it_block

Arm recommends that software does not place SG (security gate instruction) inside an IT instruction block.

ArchMsg.Warning.unpred_FAR_on_LRCPC3_non_atomic_mem_access

TAGS **UNPREDICTABLE** Unpredictable FAR/FSTATUS on LRCPC3 pair instructions resulting in two separate memory accesses
DISPLAY Unpredictable FAR on LRCPC3 non atomic memory access at %{ADDR1} with FSTATUS %{FSTATUS1} or %{ADDR2} with FSTATUS %{FSTATUS2}. Fields:

ADDR1 unsigned int
Possible FAR address 1.

ADDR2 unsigned int
Possible FAR address 2.

FSTATUS1 unsigned int
Possible fault status value 1.

FSTATUS2 unsigned int
Possible fault status value 2.

ArchMsg.Warning.unpred_async_tag_check_fail_on_sve_speculative_load_do_update_ffr

Unpredictable set to FALSE of the FFR element associated to SVE speculative load with an async tag check fail TAGS **UNPREDICTABLE**.

ArchMsg.Warning.unpredictable_a32_breakpoint

TAGS **UNPREDICTABLE** DISPLAY Unpredictable breakpoint on a A32 instruction at %{ADDR1} Offset %{OFFSET} BAS %{BAS} for a %{IS_ADDRESS_MISMATCH_BREAKPOINT:(Mismatch|Match)} Breakpoint number %{N:d} Breakpoint address %{ADDR2}. Fields:

ADDR1 unsigned int
Address.

ADDR2 unsigned int
Breakpoint address.

BAS unsigned int
BAS field.

IS_ADDRESS_MISMATCH_BREAKPOINT bool
Is a Mismatch breakpoint, not a Match breakpoint.

N unsigned int

Breakpoint number.

OFFSET unsigned int

Always has a value of zero.

ArchMsg.Warning.unpredictable_banked_register_access

TAGS **UNPREDICTABLE** DISPLAY Access to banked register (R=%{R:d} SYSm=%{SYSm:2x}) is unpredictable. Fields:

R unsigned int

R bit.

SYSm unsigned int

SYSm.

ArchMsg.Warning.unpredictable_ccfail_undef

TAGS **UNPREDICTABLE** DISPLAY An instruction %{OPCODE} at VA=%{VADDR} failed its condition codes check but would otherwise have caused a non-data-dependent trap or undefined exception. Behaviour is implementation-defined choice of exception or nop. Fields:

OPCODE unsigned int

opcode of the instruction.

VADDR unsigned int

Virtual address of the instruction being executed.

ArchMsg.Warning.unpredictable_ici_bits

TAGS **UNPREDICTABLE** DISPLAY ICI bits do not apply or are architecturally unpredictable for instruction.

ArchMsg.Warning.unpredictable_t32_breakpoint

TAGS **UNPREDICTABLE** DISPLAY Unpredictable breakpoint on a T32 instruction at %{ADDR1} Offset %{OFFSET} BAS %{BAS} for a %{IS_ADDRESS_MISMATCH_BREAKPOINT:(Mismatch|Match)} Breakpoint number %{N:d} Breakpoint address %{ADDR2}. Fields:

ADDR1 unsigned int

Address.

ADDR2 unsigned int

Breakpoint address.

BAS unsigned int

BAS field.

IS_ADDRESS_MISMATCH_BREAKPOINT bool

Is a Mismatch breakpoint, not a Match breakpoint.

N unsigned int

Breakpoint number.

OFFSET unsigned int

Indicates which halfword of a 32-bit Thumb instruction the breakpoint is set on. 0 for the first halfword or if the instruction is 16 bits wide, or 1 for the second halfword.

ArchMsg.Warning.unpredictable_t32_vector_catch

TAGS **UNPREDICTABLE** DISPLAY Unpredictable Vector catch exception on a T32 instruction at **%{ADDR}** Offset **%{OFFSET}**. Fields:

ADDR unsigned int

Address.

OFFSET unsigned int

Indicates which halfword of a 32-bit Thumb instruction the vector catch exception is set on. 0 for the first halfword or if the instruction is 16 bits wide, or 1 for the second halfword.

ArchMsg.Warning.unpredictable_watchpoint_far

TAGS **UNPREDICTABLE** DISPLAY The value for FAR/EDWAR for hitting this watchpoint may be anywhere in the range **%{LOWER_BOUND}** to **%{UPPER_BOUND}**. Fields:

LOWER_BOUND unsigned int

the lowest address accessed by the instruction that triggered the watchpoint.

UPPER_BOUND unsigned int

the highest watchpointed address accessed by that instruction.

ArchMsg.Warning.warning_AdvSIMDExpandImmUnexpectedZero

TAGS **UNPREDICTABLE** DISPLAY AdvSIMDExpandImm may treat this immediate value as **UNPREDICTABLE**.

ArchMsg.Warning.warning_ConditionalSMC

TAGS **UNPREDICTABLE** DISPLAY SMC instruction has **UNPREDICTABLE** effects when conditional and when combined with traps.

ArchMsg.Warning.warning_access_crosses_page_boundary

TAGS **UNPREDICTABLE** DISPLAY Access crosses page boundary.

ArchMsg.Warning.warning_access_crosses_page_boundary_has_fault

TAGS **UNPREDICTABLE** DISPLAY Access that crosses page boundary has fault(s).

ArchMsg.Warning.warning_access_crosses_page_boundary_has_fault_on_both_pages

TAGS **UNPREDICTABLE** DISPLAY Access that crosses page boundary has a fault on each page. Fields:

ADDR1 unsigned int

FAR address on first page.

ADDR2 unsigned int

FAR address on second page.

FSTATUS1 unsigned int

Fault status on first page.

FSTATUS2 unsigned int

Fault status on second page.

ArchMsg.Warning.warning_access_crosses_page_boundary_spanning_different_memory

TAGS **UNPREDICTABLE** DISPLAY Access at address %{ADDR} crosses page from MEM_TYPE %{MEMTYPE_PAGE1} to %{MEMTYPE_PAGE2}. Fields:

ADDR unsigned int

address of access crossing page.

MEMTYPE_PAGE1 signed int

First Page's memory type (0-Normal, 1-Non-Normal).

MEMTYPE_PAGE2 signed int

Second Page's memory type (0-Normal, 1-Non-Normal).

ArchMsg.Warning.warning_access_wraps_around_memory

TAGS **UNPREDICTABLE** DISPLAY Access wraps around memory in %{WIDTH:d}bit mode. Fields:

WIDTH unsigned int

Address width.

ArchMsg.Warning.warning_bcr_linking_status

TAGS **UNPREDICTABLE** DISPLAY DBGBCR.{BT2,BT} combination gives unpredictable behaviour.

ArchMsg.Warning.warning_bcr_mask_reserved

TAGS **UNPREDICTABLE** DISPLAY DBGBCR_ELn.MASK is set to a reserved value.

ArchMsg.Warning.warning_bvr_masked_bit_not_zero

TAGS **UNPREDICTABLE** DISPLAY DBGBVRn_EL1.VA contains a bit masked by DBGBCRn_EL1.MASK that is not zero.

ArchMsg.Warning.warning_ccsidr_unimplemented_level

TAGS **UNPREDICTABLE** DISPLAY CCSIDR read while CSSELR %{CSSELR:x} points to an unimplemented cache level. Fields:

CSSELR unsigned int

current effective value of CSSELR.

ArchMsg.Warning.warning_change_to_ns_when_tge_set

TAGS **UNPREDICTABLE** DISPLAY Attempting to change NS to 1 when HCR.TGE = 1.

ArchMsg.Warning.warning_contiguous_bit_check_abort

TAGS **UNPREDICTABLE** DISPLAY An abort occurred whilst attempting to check TLB entry is contiguous at %{ENTRY_ADDR}. Fields:

ENTRY_ADDR unsigned int

address of conflicting TLB entry.

ArchMsg.Warning.warning_contiguous_bit_error

TAGS **UNPREDICTABLE** DISPLAY TLB entry at %{ENTRY_ADDR} was not contiguous with entry at %{CONTIG_ADDR} Contents are %{ENTRY_DATA} but expected %{CONTIG_DATA}. Fields:

CONTIG_ADDR unsigned int

address of the TLB first read, with which this entry is expected to be contiguous.

CONTIG_DATA unsigned int

expected contents based on the entry at CONTIG_ADDR.

ENTRY_ADDR unsigned int

address of conflicting TLB entry.

ENTRY_DATA unsigned int

contents of conflicting TLB entry.

ArchMsg.Warning.warning_contiguous_bit_error_D128

TAGS **UNPREDICTABLE** DISPLAY TLB entry at %{ENTRY_ADDR} was not contiguous with entry at %{CONTIG_ADDR} Contents are %{ENTRY_DATA_HI} %{ENTRY_DATA_LO} but expected %{CONTIG_DATA}. Fields:

CONTIG_ADDR unsigned int

address of the TLB first read, with which this entry is expected to be contiguous.

CONTIG_DATA unsigned int

expected contents based on the entry at CONTIG_ADDR.

ENTRY_ADDR unsigned int

address of conflicting TLB entry.

ENTRY_DATA_HI unsigned int

Higher bit contents of conflicting TLB entry.

ENTRY_DATA_LO unsigned int

Lower bit contents of conflicting TLB entry.

ArchMsg.Warning.warning_cp10_cp11_mismatch

TAGS **UNPREDICTABLE** DISPLAY Unpredictable configuration of CP10 and CP11 controls at EL %{EL:d}. Fields:

EL unsigned int

EL owning the control.

ArchMsg.Warning.warning_cp10_cp11_reserved_value

TAGS **UNPREDICTABLE** DISPLAY Unpredictable configuration of CPACR CP10 (%{CP10}) or CP11 (%{CP11}). Fields:

CP10 unsigned int

CP10 access.

CP11 unsigned int

CP11 access.

ArchMsg.Warning.warning_csselr_level_out_of_range

TAGS **UNPREDICTABLE** DISPLAY CSSELR.Level out of range, written value %{LEVEL} with only %{IMPLEMENTED:d} levels implemented. Fields:

IMPLEMENTED unsigned int

number of cache levels implemented.

LEVEL unsigned int

written value of CSSELR.Level.

TYPE enum

The type of cache selected by CSSELR.InD and TnD.

ArchMsg.Warning.warning_debug_flow_control_bits_not_obeyed

TAGS **UNPREDICTABLE** DISPLAY Flow control bits for %{REGISTER} were not obeyed, causing %{ERROR}. Fields:

ERROR enum

type of error resulting.

REGISTER enum

register under consideration.

ArchMsg.Warning.warning_debug_register_access_during_reset

TAGS **UNPREDICTABLE** DISPLAY Unpredictable %{IS_WRITE:(read|write)} access to debug register offset %{OFFSET} during reset. Fields:

IS_WRITE unsigned int

Write Not Read.

OFFSET unsigned int

Register Offset.

ArchMsg.Warning.warning_decode_cps_inconsistent_fields

TAGS **UNPREDICTABLE** DISPLAY The mode setting fields in the CPS instruction are inconsistent.

ArchMsg.Warning.warning_decode_invalid_state

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable in current exception-level/security state.

ArchMsg.Warning.warning_default_cacheable_mmu_on

TAGS **UNPREDICTABLE** DISPLAY Default cacheable is enabled (HCR.DC = 1) while MMU is enabled.

ArchMsg.Warning.warning_default_cacheable_vmmu_off

TAGS **UNPREDICTABLE** DISPLAY Default cacheable is enabled (HCR.DC = 1) while HCR.VM = 0.

ArchMsg.Warning.warning_deprecated_wvr_bit_2_set

TAGS **UNPREDICTABLE** DISPLAY DBGWVR%{N:d}_EL1=%{WVR:h} has bit 2 set, deprecated in ARMv8. Fields:

N unsigned int
index.

WVR unsigned int
DBGWVR.

ArchMsg.Warning.warning_eret_while_software_step_active_pending

Missing ISB between setting MDSCR_EL1.SS and ERET TAGS UNPREDICTABLE.

ArchMsg.Warning.warning_exclusive_to_non_normal

TAGS UNPREDICTABLE DISPLAY Exclusive Access to Non-Normal Memory %{ADDR}. Fields:

ADDR unsigned int
Address.

ArchMsg.Warning.warning_exclusive_to_non_writeback

TAGS UNPREDICTABLE DISPLAY Exclusive Access to Non-Writeback-Cacheable Memory %{ADDR}. Fields:

ADDR unsigned int
Address.

ArchMsg.Warning.warning_execute_from_device_memory

TAGS UNPREDICTABLE DISPLAY An attempt was made to execute from Device memory.

ArchMsg.Warning.warning_illegal_cpsr_mode

TAGS UNPREDICTABLE DISPLAY Writing an illegal or unimplement mode (%{NEW_MODE:h}) to CPSR was unpredictable. Fields:

NEW_MODE unsigned int
New value of CPSR.M.

ArchMsg.Warning.warning_illegal_srs_mode

TAGS UNPREDICTABLE DISPLAY Illegal or UNPREDICTABLE mode (%{MODE:h}) used for SRS instruction. Fields:

MODE unsigned int
mode for Banked SP.

ArchMsg.Warning.warning_implementation_defined_read_debug_register_in_SCS

Read Access to Debug registers from the processor is IMPLEMENTATION DEFINED TAGS IMP_DEF. Fields:

PPB_OFFSET unsigned int
debug register offset.

ArchMsg.Warning.warning_implementation_defined_sequential_security_transitions_supported

The behavior of sequential instruction fetches that cross from non-secure to secure memory and contain an SG instruction is CONSTRAINED UNPREDICTABLE TAGS UNPREDICTABLE.

ArchMsg.Warning.warning_implementation_defined_stack_limit_check_supported

It is **IMPLEMENTATION DEFINED** whether stack pointer limit checking is performed for this instruction TAGS IMP_DEF. Fields:

ADDRESS unsigned int

Address of instruction.

OPCODE unsigned int

instruction opcode.

ArchMsg.Warning.warning_implementation_defined_taggednonsharedmemorytype

TAGS IMP_DEF DISPLAY Memory attribute has Tagged, non shared memory type; it is **IMPLEMENTATION DEFINED** whether the memory location is treated as Tagged or non-Tagged.

ArchMsg.Warning.warning_implementation_defined_write_debug_register_in_SCS

Write Access to Debug registers from the processor is **IMPLEMENTATION DEFINED** TAGS IMP_DEF. Fields:

DATA unsigned int

data attempted to be written.

PPB_OFFSET unsigned int

debug register offset.

ArchMsg.Warning.warning_invalid_tcr_granule

TAGS **UNPREDICTABLE** DISPLAY TCR.TG%{TG_ID:(0|1)} bits have been set to a granule size %{{REQUEST:d}K} not implemented - using %{{SUBSTITUTE:d}K}. Fields:

REQUEST signed int

page size requested (or 0 for reserved).

SUBSTITUTE signed int

best guess available page size.

TG_ID bool

bits TG0 or TG1.

ArchMsg.Warning.warning_load_multiple_user_registers_from_user_mode

TAGS **UNPREDICTABLE** DISPLAY An LDM(user registers) instruction executed from user mode.

ArchMsg.Warning.warning_mte_in_mem_attr_not_supported

TAGS **UNPREDICTABLE** DISPLAY Memory attribute %{{ATTR}} has Tagged memory type but Armv8.5-MTE is not implemented. Fields:

ATTR unsigned int

Memory attribute.

ArchMsg.Warning.warning_reserved_breakpoint_state_match

TAGS **UNPREDICTABLE** DISPLAY %{{IS_BREAKPOINT:(Watchpoint|Breakpoint)}} programmed with a reserved combination of HMC, SSC and %{{IS_BREAKPOINT:(PAC|PMC)}}. Fields:

IS_BREAKPOINT unsigned int

Is a breakpoint, not a watchpoint.

ArchMsg.Warning.warning_shareability

TAGS **UNPREDICTABLE** DISPLAY Unpredictable: combination of the 1st and 2nd stages of translation is Normal Inner Non-Cacheable, Outer Non-Cacheable.

ArchMsg.Warning.warning_software_step_set_while_enabled

TAGS **UNPREDICTABLE** DISPLAY MDSCR_EL1.SS set to 1 while software step debug exceptions are enabled.

ArchMsg.Warning.warning_thumb_instruction_wraps_around_memory

TAGS **UNPREDICTABLE** DISPLAY Thumb instruction wraps around memory.

ArchMsg.Warning.warning_ttbr_sbz_bits_are_not_zero

TAGS **UNPREDICTABLE** DISPLAY %{TTBR}[%{MSB:d}:0] should be zero, but are not. Fields:

MSB unsigned int

x-1.

TTBR enum

which TTBR register.

ArchMsg.Warning.warning_unaligned_address_dbgdtrrx_write

TAGS **UNPREDICTABLE** DISPLAY Unaligned address %{ADDR} in X0/R0 when DBGDTRRX written in memory access mode. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unaligned_address_dbgdtrtx_read

TAGS **UNPREDICTABLE** DISPLAY Unaligned address %{ADDR} in X0/R0 when DBGDTRTX read in memory access mode. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unaligned_to_device

TAGS **UNPREDICTABLE** DISPLAY Unaligned Access to Device Memory %{ADDR}. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unaligned_to_strongly_ordered

TAGS **UNPREDICTABLE** DISPLAY Unaligned Access to Strongly Ordered Memory %{ADDR}. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unknown_sau_rnr

TAGS UNPREDICTABLE DISPLAY SAU_RNR was set to an unsupported value.

ArchMsg.Warning.warning_unpredictable_AIRCR_PRIS_and_BFHFNMINs

TAGS UNPREDICTABLE DISPLAY The effect of setting both AIRCR.BFHFNMINs and AIRCR.PRIS to 1 is UNPREDICTABLE.

ArchMsg.Warning.warning_unpredictable_AIRCR_VECTCLRACTIVE_when_not_in_debug

TAGS UNPREDICTABLE DISPLAY The effect of writing a 1 to AIRCR.VECTCLRACTIVE if the processor is not halted in Debug state is UNPREDICTABLE.

ArchMsg.Warning.warning_unpredictable_AIRCR_VECTRESET_and_SYSRESETREQ

TAGS UNPREDICTABLE DISPLAY When the processor is halted in Debug state, if a write to the register writes a 1 to both VECTRESET and SYSRESETREQ, the behavior is UNPREDICTABLE.

ArchMsg.Warning.warning_unpredictable_AIRCR_VECTRESET_when_not_in_debug

TAGS UNPREDICTABLE DISPLAY The effect of writing a 1 to AIRCR.VECTRESET if the processor is not halted in Debug state is UNPREDICTABLE.

ArchMsg.Warning.warning_unpredictable_AIRCR_incorrect_VKEY

TAGS UNPREDICTABLE DISPLAY The value 0x05FA must be written to AIRCR.VKEY, otherwise the register write is UNPREDICTABLE.

ArchMsg.Warning.warning_unpredictable_EXC_RETURN_Reserved_Bit

EXC_RETURN[23:7] are reserved with the special condition that all bits should be written as one. Values other than all 1s are UNPREDICTABLE. TAGS UNPREDICTABLE. Fields:

EXC_RETURN unsigned int

Special PC value.

SBO_MASK unsigned int

Which bits Should Be One to not UNPRED.

ArchMsg.Warning.warning_unpredictable_PMEVTYPEPn_ELO_VS_reserved_value

TAGS UNPREDICTABLE DISPLAY Unpredictable reserved value of PMEVTYPEn<n>_ELO.VS.

ArchMsg.Warning.warning_unpredictable_change_to_DEMCR_MON_STEP_at_insufficient_priority

TAGS UNPREDICTABLE DISPLAY The effect of changing DEMCR.MON_STEP at an execution priority that is lower than the priority of the DebugMonitor exception is **CONSTRAINED** UNPREDICTABLE.

ArchMsg.Warning.warning_unpredictable_change_to_priority_of_active_exception

TAGS UNPREDICTABLE DISPLAY Changing the priority of an active exception is UNPREDICTABLE.

ArchMsg.Warning.warning_unpredictable_clear_lspact

TAGS UNPREDICTABLE DISPLAY UNPREDICTABLE state on context switch. Fields:

DRVEXC_TAKEN unsigned int

Derived exception taken, not pended.

REACHED_LAST_STORE unsigned int

Only the last faulted.

ArchMsg.Warning.warning_unpredictable_dbg_exit_unaligned_dlr

TAGS **UNPREDICTABLE** DISPLAY Debug exit to AArch32 with DLR[0] set to 1. PC[0] can be set to 0 or to DLR[0].

ArchMsg.Warning.warning_unpredictable_exceptionCatch

TAGS **UNPREDICTABLE** DISPLAY Generation of an exception catch event is unpredictable.

ArchMsg.Warning.warning_unpredictable_exception_return_inconsistent_state

TAGS **UNPREDICTABLE** DISPLAY state on exception return is unpredictable.

ArchMsg.Warning.warning_unpredictable_exception_return_instruction

TAGS **UNPREDICTABLE** DISPLAY exception return instruction is unpredictable.

ArchMsg.Warning.warning_unpredictable_in_debug_state

TAGS **UNPREDICTABLE** DISPLAY Instruction is **UNPREDICTABLE** when executed in debug state.

ArchMsg.Warning.warning_unpredictable_mair_encoding

TAGS **UNPREDICTABLE** DISPLAY Memory attribute %{ATTR} has invalid MAIR encoding; this may lead to unpredictable results. Fields:

ATTR unsigned int

Memory attribute.

ArchMsg.Warning.warning_unpredictable_pmu_counter_access

TAGS **UNPREDICTABLE** DISPLAY Access to PMU counters from non-secure EL0 or EL1 is **UNPREDICTABLE** with MDCR_EL2.HPMN set to 0.

ArchMsg.Warning.warning_unpredictable_prioritization_breakpoint_match_vector_catch

TAGS **UNPREDICTABLE** DISPLAY Two events with the same priority occurred at the same instruction: Address Matching Vector Catch debug event, and Breakpoint debug event. It is constrained unpredictable which is taken.

ArchMsg.Warning.warning_unpredictable_stack_selection

TAGS **UNPREDICTABLE** DISPLAY **UNPREDICTABLE** state on context switch.

ArchMsg.Warning.warning_unpredictable_tsize_out_of_range

TAGS **UNPREDICTABLE** DISPLAY %{TCR}.T%{TTBR:d}SZ (%{TSIZE:d} bits) is out of range. Must be between 25 and %{TSIZE_MAX:d}. Fields:

TCR enum

Which TCR.

TSIZE signed int

Value of TSize.

TSIZE_MAX signed int

The maximum allowed value.

TTBR unsigned int

Which TTBR region.

ArchMsg.Warning.warning_unpredictable_unaligned_pc_as_base_register

TAGS **UNPREDICTABLE** DISPLAY The PC value must be word-aligned, otherwise the behavior of the instruction is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_unaligned_pop_stack

TAGS **UNPREDICTABLE** DISPLAY **CONSTRAINED UNPREDICTABLE** unaligned stack frame %{FRAME_ADDR:x} in exception-return. Fields:

FRAME_ADDR unsigned int

unaligned stack frame address.

ArchMsg.Warning.warning_unpredictable_unaligned_to_non_writeback

TAGS **UNPREDICTABLE** DISPLAY Unaligned Access to Non-Writeback-Cacheable Memory.

ArchMsg.Warning.warning_unpredictable_vmsa_memattrib

TAGS **UNPREDICTABLE** DISPLAY Unpredictable vmsa memory attribute with texcb value programmed to %{TEXCB} when tex remap is %{TEX_REMAP}. Fields:

TEXCB unsigned int

texcb value.

TEX_REMAP bool

use tex remap.

ArchMsg.Warning.warning_unpredictable_vtcr_t0sz_sl0

TAGS **UNPREDICTABLE** DISPLAY The combination of VTCR.T0SZ=%{T0SZ:d} and VTCR.SL0=%{SL0:d} is unpredictable. Fields:

SL0 signed int

VTCR.SL0, starting level for stage 2 translation table walks.

T0SZ signed int

VTCR.T0SZ, required input address range.

ArchMsg.Warning.warning_unpredictable_write_DHCSR

TAGS **UNPREDICTABLE** DISPLAY **UNPREDICTABLE** set of bit-changes in DHCSR. Fields:

NEW unsigned int

data attempted to be written.

OLD unsigned int

previous value.

ArchMsg.Warning.warning_unpredictable_write_SHCSR

TAGS **UNPREDICTABLE** DISPLAY **UNPREDICTABLE** setting a pending bit to 1 for an exception with priority >= execution priority.

ArchMsg.Warning.warning_unsupported_access_to_memory_mapped_register

TAGS **UNPREDICTABLE** DISPLAY Access to memory mapped register is unsupported.

ArchMsg.Warning.warning_user_jmcr_access

TAGS UNPREDICTABLE DISPLAY User %{WRITE:(write|read)} access to JMCR is unpredictable.
Fields:

WRITE unsigned int
access is write.

ArchMsg.Warning.warning_wcr_mask_and_bas

TAGS UNPREDICTABLE DISPLAY DBGWCR_EL1.MASK is non-zero and BAS is not 0xff.

ArchMsg.Warning.warning_wcr_mask_reserved

TAGS UNPREDICTABLE DISPLAY DBGWCR_ELn.MASK is set to a reserved value.

ArchMsg.Warning.warning_wcr_non_configuous_bas

TAGS UNPREDICTABLE DISPLAY DBGWCR_EL1.BAS has non-contiguous set of ones.

ArchMsg.Warning.warning_wvr_masked_bit_not_zero

TAGS UNPREDICTABLE DISPLAY DBGWVRn_EL1.VA contains a bit masked by
DBGWCRn_EL1.MASK that is not zero.

ArchMsg.warning_non_architectural_beat_order

DISPLAY Current set of committed beats does not give legal ECI value.

ArchMsg.warning_unpred_system_register_access

DISPLAY Unpredictable %{IS_WRITE:(read|write)} access to system register offset
%{OFFSET}. Fields:

IS_WRITE unsigned int
Write Not Read.

OFFSET unsigned int
Register Offset.

BEAT_EXCEPTION

Exception during of one beat of an M. Fields:

BEAT unsigned int
Beat number 0..3.

EXC unsigned int
Internal representation of the excep.

EXCEPTION_PC unsigned int
Address of older in-flight instructi.

INST unsigned int
This instruction as FirstHalfWord:Se.

INST_ADDR unsigned int
Address of faulting instruction inst.

BEAT_START

Start of one beat of an MVE Vector i. Fields:

BEAT unsigned int

Beat number 0..3.

DISASS string

Disassembly of instruction.

ELMT_MASK unsigned int

Element Mask.

EXCEPTION_PC unsigned int

Address of older in-flight instructi.

INST unsigned int

This instruction as FirstHalfWord:Se.

INST_ADDR unsigned int

Address of this instruction.

LTP_MASK unsigned int

Element Mask from Loop Tail Predicat.

VPT_MASK unsigned int

Element Mask from VPR.

BRANCH_MISPREDICT

Simulating branch mispredict. Fields:

PC unsigned int

Origin address (or 0 if unavailable).

BRA_DIR

Direct branches, to immediate address. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISSET enum

The instruction set of the branch instruction.

IS_COND bool

Indicates if this is a conditional branch.

IS_HINTED bool

Whether the conditional instruction is hinted.

IS_ISB bool

Whether the instruction is ISB.

IS_LINK bool

Indicates if this is a branch with link.

PC unsigned int

The address of the branch instruction.

TARGET_ISET enum

The instruction set after the branch.

TARGET_PC unsigned int

The address the instruction branches to.

BRA_INDIR

Indirect branches, perhaps to a register. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISET enum

The instruction set of the branch instruction.

IS_COND bool

Indicates if this is a conditional branch.

IS_HINTED bool

Whether the conditional instruction is hinted.

IS_ISB bool

Whether the instruction is ISB.

IS_LINK bool

Indicates if this is a branch with link.

PC unsigned int

The address of the branch instruction.

TARGET_ISET enum

The instruction set after the branch.

TARGET_PC unsigned int

The address the instruction branches to.

BRA_INDIR_CCFAIL

Indirect branches fail, perhaps to a register.

BRA_LOB

Low Overhead branch. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISSET enum

The instruction set of the branch instruction.

IS_COND bool

Indicates if this is a conditional branch.

IS_HINTED bool

Whether the conditional instruction is hinted.

IS_ISB bool

Whether the instruction is ISB.

IS_LINK bool

Indicates if this is a branch with link.

PC unsigned int

The address of the branch instruction.

TARGET_ISET enum

The instruction set after the branch.

TARGET_PC unsigned int

The address the instruction branches to.

BRA_RET_CCFAIL

Branch return fail, perhaps to a register.

CACHE_MAINTENANCE_OP

Cache Maintenance Operation. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

DATA unsigned int

Specified MVA or Set/way.

FUNCTION enum

Clean / Invalidate.

SCOPE enum

Affected region.

SIDE enum

Inst / Data.

CCFAIL

Conditional instruction condition check fail. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CCFAIL_UNC

Unconditional instruction condition check fail. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CCPASS

Conditional instruction condition check pass. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CCPASS_UNC

Unconditional instruction condition check pass. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CODE_CACHE_MAINT

Code cache maintenance. Fields:

END_INCL unsigned int

Inclusive end address for VA or PA requests.

NS bool

Non-secure world for PA.

START unsigned int

Start address for VA or PA requests.

TYPE enum

Request type.

COMPILE_BLOCK_END

Last instruction of basic block translated. Fields:

VADDR unsigned int

Address of next instruction after this basic block.

COMPILE_BLOCK_START

First instruction of basic block translated. Fields:

VADDR unsigned int

Address of first instruction in basic block.

COMPILE_INST

ARM instruction compiled. Fields:

DISASS string

Disassembly of the instruction.

ISSET enum

The instruction set of this instruction.

ITSTATE unsigned int

The ITSTATE current for the instruction.

OPCODE unsigned int

The opcode of the instruction.

PC unsigned int

The address of the instruction.

SIZE unsigned int

The size of the instruction in bytes.

CONTEXTIDR

Context ID Register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

NS bool

Secure or nonsecure banked register is accessed.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The new value written.

CONTEXT_SYNC

Called for every context synchronization event. Fields:

ARCH_CSE bool

Architectural Context Synchronization Event.

PC unsigned int

The address of the synchronization instruction.

SYSREG_SYNC bool

System register synchronization event.

CORE_ENDIAN

Core BE8 Big-Endian state changed. Fields:

BE8 bool

Core BE8 Big-Endian state.

CORE_INFO

Static processor attributes. Only triggered by a call to DumpState(). Fields:

ARCH_PROFILE enum

The architecture profile of the core.

CLUSTER_ID unsigned int

The cluster ID of this processor.

CORE_NUM unsigned int

The number of this core in an MP processor.

DETAILED_CORE_FEATURES enum

Detailed core features as a bitfield. Enum values are bitmasks to extract the info from the field.

FPU_VERSION enum

The VFP version implemented by the core.

MEM_ARCH enum

The memory architecture of the core.

MIDR_EL1 unsigned int

MIDR_EL1 value.

NUM_CORES unsigned int

The number of cores in this MP processor.

QUANTUM_SIZE unsigned int

The default quantum size of the core (in ticks).

SECURITY_FEATURES bool

Does the core have security features?.

CORE_LOADS

Processor load accesses. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ACQREL enum

Is this an acquire/release.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DATA unsigned int

The data read or written.

ELEMENT_SIZE unsigned int

Width of each element.

ENDIANNESS enum

Byte ordering of this access.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

LOCK enum

Normal, exclusive or locked access.

MECID unsigned int

Memory Encryption Context ID, if applicable.

MECID2 unsigned int

Memory Encryption Context ID of the second page, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PAGE_SIZE unsigned int

Page size as $\log_2(\text{size in bytes})$.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK, 2=Exclusive Failed, 3=Unsupported.

SIZE unsigned int

Width of the access in bytes. Only required if DATA is not traced.

TRANS bool

Is this a translated access.

VADDR unsigned int

The virtual address of the access.

CORE_REGS

Changes of the core registers R0 to R14. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number, 0 to 14.

OLD_VALUE unsigned int

The old value overwritten.

PHYS_ID enum

The physical register accessed.

VALUE unsigned int

The new value written to the register.

CORE_STALLED_TRANSITION

Detect when a core is prevented from progressing PChannel states. Fields:

CURRENT_STATE enum

Current power mode of core.

REASON string

The reason a transition was stalled, and what must be done to unblock it.

TARGET_STATE enum

Desired power mode of core after transition.

CORE_STORES

Processor store accesses. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ACQREL enum

Is this an acquire/release.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DATA unsigned int

The data read or written.

ELEMENT_SIZE unsigned int

Width of each element.

ENDIANNESS enum

Byte ordering of this access.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

LOCK enum

Normal, exclusive or locked access.

MECID unsigned int

Memory Encryption Context ID, if applicable.

MECID2 unsigned int

Memory Encryption Context ID of the second page, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PAGE_SIZE unsigned int

Page size as $\log_2(\text{size in bytes})$.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK, 2=Exclusive Failed, 3=Unsupported.

SIZE unsigned int

Width of the access in bytes. Only required if DATA is not traced.

TRANS bool

Is this a translated access.

VADDR unsigned int

The virtual address of the access.

CRYPTO_SPEC

Every crypto instruction speculatively executed.

DATA_CACHE_SET_ALLOC_TAGS

Set allocation tags from DC GVA instruction. Fields:

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

NSDESC unsigned int

The security state of the access.

PADDR unsigned int

The physical address of the access.

PAS enum

The physical address space of the page.

RESPONSE enum

0=Aborted, 1=OK.

SIZE unsigned int

The size of memory in bytes.

TAG unsigned int

The value of the tag being set.

TAGGING_TYPE enum

Type of Memory tagging (Physical or Virtual).

TAG_VADDR unsigned int

The virtual address of the Virtual Tag if Virtual tagging is used; otherwise 0.

VADDR unsigned int

Virtual address used for setting allocation tags.

DATA_CACHE_ZERO

Zero and invalidate cache line from DC ZVA instruction. Fields:

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

NSDESC unsigned int

The security state of the access.

PADDR unsigned int

The physical address of the access.

PADDR_TAG unsigned int

If tag checked, the physical address tag for the access.

PAS enum

The physical address space of the page.

RESPONSE enum

0=Aborted, 1=OK.

SIZE unsigned int

The size of zero'd memory in bytes.

TAG_CHECKED bool

Is this access tag checked.

VADDR unsigned int

Virtual address of the zero'd cache line.

DEBUG_EVENT

Hardware debug support event. Fields:

EVENT enum

Description of event.

VALUE unsigned int

data value.

DMI_ALLOCATE

DMI allocated. Fields:

ACCESS_DIRECTION enum

READ / WRITE.

PADDR unsigned int

The physical address of the DMI.

PTR unsigned int

DMI range pointer.

SIDE enum

DSIDE / ISIDE.

SIZE unsigned int

The size of DMI.

VADDR unsigned int

The virtual address of the DMI..

DMI_HIT

DMI hit, access used DMI (only fire for small DMI regions i.e. less than a full page). Fields:

ACCESS_DIRECTION enum

READ / WRITE.

PADDR unsigned int

The physical address of the DMI.

SIDE enum

DSIDE / ISIDE.

SIZE unsigned int

The size of the DMI.

VADDR unsigned int

The virtual address of the DMI..

DMI_REVOKE

DMI revoked. Fields:

ACCESS_DIRECTION enum

READ / WRITE.

PADDR unsigned int

The physical address of the DMI.

SIDE enum

DSIDE / ISIDE.

SIZE unsigned int

The size of DMI.

VADDR unsigned int

The virtual address of the DMI..

DTCM_ACCESS

Trace TCM accesses.

DTCM_ACCESS_RAZ_WI

Trace TCM accesses. Fields:

ADDR unsigned int

Address of the **RAZ/WI** access.

DWT_MATCH

DWT comparator matches. Fields:

NUM unsigned int

DWT comparator number.

TYPE enum

DWT comparator configuration.

END_COMPILE

Compilation end. Fields:

END_OF_PAGE_COUNT unsigned int

Number of basic blocks exited due to page boundary since **START_COMPILE**.

FETCHFAIL_COUNT unsigned int

Number of basic blocks exited due to ifetch failure **START_COMPILE**.

INST_COUNT unsigned int

Number of instructions compiled since **START_COMPILE**.

NONSEQ_COUNT unsigned int

Number of basic blocks exited due to non-sequential instructions since **START_COMPILE**.

PAGE_STRADDLE_COUNT unsigned int

Number of basic blocks exited due to unaligned instructions crossing page since **START_COMPILE**.

EXCEPTION

Exceptions that are taken. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ISSET enum

The instruction set of the processor when the exception occurred.

LR unsigned int

The value assigned to the link register.

NS enum

The core's non-secure bit.

PC unsigned int

The location where the exception occurred.

TARGET_ISSET enum

The instruction set of the exception handler code.

TARGET_PC unsigned int

The address the exception branches to.

VECTOR enum

The exception vector.

EXCEPTION_END

Every exception completed. Fields:

NS enum

The core's non-secure bit.

EXCEPTION_ENTRY

Event marking the entry of an exception. All pushes of registers to the stack will follow this event. Fields:

PC unsigned int

The location where the exception occurred.

VECTOR enum

The exception that occurred.

EXCEPTION_RAISE

Every exception raised.

EXCEPTION_RETURN

Marks the end of an exception, but on an M core does not indicate a program flow change (branch). In most situations it will be followed by an indirect branch. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISSET enum

The instruction set of the branch instruction.

PC unsigned int

The address of the branch instruction.

EXCEPTION_RETURN_PREBRANCH

Exception return event that must occur before PC and processor state have updated.

EXCEPTION_START

Every exception started. Fields:

NS enum

The core's non-secure bit.

EXCEPTION_VECTOR_FETCH

I-side vector fetch for M-class exception. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

DATA unsigned int

Result of fetch if OK.

RESPONSE enum

0=Aborted, 1=OK.

VADDR unsigned int

Where the vector is fetched from (including VTOR).

FP_STATE

Floating point state. Fields:

FPSR unsigned int

FPSR register value.

MASK unsigned int

The mask indicating updated cumulative bits.

FREQ_CHANGED

Core frequency changed. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The instruction count of this core.

NEW_FREQ signed int

The new frequency of this core (expressed in Hz).

OLD_FREQ signed int

The old frequency of this core (expressed in Hz).

HLT

HLT instruction occurred. Fields:

IMM16 unsigned int

The 16-bit immediate value encoded in HLT instruction.

INFO_EXCEPTION_REASON

Provide information of real cause of exception. Fields:

FaultCause enum

Which bit of which V7M/V8M-mainline FSR/HFSR would have been set.

PC unsigned int

The location where the exception occurred.

PHASE enum

What we are doing with the exception.

REASONS enum

Why we are doing it.

SecurityState enum

Privacy State of exception.

VECTOR enum

The exception that occurred.

INFO_FP_CONTEXT

Inform when ExecuteFPCheck writes CONTROL.FPCA and FPSCR etc. Fields:

WHAT enum

What we are doing.

INFO_STACKING

Provide information of real cause of exception. Fields:

PHASE enum

Are we starting or ending a sequence of stack writes.

REASONS enum

Why we are doing it.

INST

Every instruction executed. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DEBUG_STATE bool

The instruction is executed in debug state.

DISASS string

Disassembly of instruction.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISET enum

The current instruction set.

ITSTATE unsigned int

The current ITSTATE.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

MODE enum

The mode the core is in.

NS unsigned int

The core's non-secure bit.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

OPCODE unsigned int

The opcode of the instruction.

PADDR unsigned int

The physical address of the instruction.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the instruction.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the page.

PC unsigned int

The address of the instruction.

SECURITY_STATE enum

The core's security state.

SIZE unsigned int

The size of the instruction in bytes.

INST_END

Every instruction completed.

INST_START

Every instruction started. Fields:

DEBUG_STATE bool

The instruction is executed in debug state.

ISET enum

The current instruction set.

MODE enum

The mode the core is in.

NS enum

The current Secure State.

PC unsigned int

The address of the conditional instruction.

SECURITY_STATE enum

The current Security State.

INST_STRADDLE

Instruction straddles boundary. Fields:

DEBUG_STATE bool

The instruction is executed in debug state.

ISET enum

The current instruction set.

MODE enum

The mode the core is in.

NS enum

The current Secure State.

PC unsigned int

The address of the conditional instruction.

SECURITY_STATE enum

The current Security State.

ISB

ISB instruction occurred. Fields:

PC unsigned int

The address of the ISB instruction.

ITCM_ACCESS

Trace TCM accesses.

ITCM_ACCESS_RAZ_WI

Trace TCM accesses. Fields:

ADDR unsigned int

Address of the **RAZ/WI** access.

ITM

Instrumentation Trace Macrocell. Fields:

ITM_PACKET_TYPE enum

ITM and DWT packets type.

PACKET_HEADER unsigned int

ITM Packet Header.

PACKET_PAYLOAD unsigned int

ITM Packet Payload.

LOAD_MULTIPLE_REGS_LIST

List of destination registers (not source) of a multiple registers load instruction. Fields:

REG string

Register to add into the list.

LOCAL_MONITOR

Local monitor activity. Fields:

PADDR unsigned int

Local Monitor Address.

State enum

State of the monitor (Open/Exclusive).

LOCKUP_CYCLE

This event is triggered if the core enters or stays in the lock-up state. Fields:

CAUSE enum

What causes the lockup?.

LOCKUP_ENTRY

This event is triggered if the core enters the lock-up state. Fields:

PC unsigned int

The PC when the lockup state is entered.

LOCKUP_EXIT

This event is triggered if the core leaves the lock-up state.

LO_BRANCH_INFO

Low Overhead Branch cache Info. Fields:

BF unsigned int

Value originates from a BF instructi.

CAUSE enum

What caused the update.

END_ADDR unsigned int

Address of the instruction after the.

JUMP_ADDR unsigned int

Jump address.

LF unsigned int

Link/forever.

T16IND unsigned int

T16 indirect branch.

VALID unsigned int

Entry is valid.

LO_BRANCH_INSTR

Low Overhead Branch Instruction. Fields:

EFFECT enum

Did it set LO_BRANCH_INFO and/or branch.

INSTR enum

What sort of instruction (BF/..BFCSEL/LE/WLS/...).

MODE_CHANGE

Mode change. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

MODE enum

The new mode.

OLD_MODE enum

The old mode.

MPU_TRANS

Address translation information. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

EXEC_PERM enum

Execution Permission.

IDAU_REG_NUM enum

IDAU region-number or code.

INNERCACHE_RA bool

Is the inner cache allocate on read.

INNERCACHE_TYPE enum

Inner Caching scheme (NC/MB/WA).

INNERCACHE_WA bool

Is the inner cache allocate on write.

MEMTYPE enum

Memory type.

NSDESC enum

Security-attributes of the transaction.

OUTERCACHE_RA bool

Is the outer cache allocate on read.

OUTERCACHE_TYPE enum

Outer Caching scheme (NC/MB/WA).

OUTERCACHE_WA bool

Is the outer cache allocate on write.

PADDR unsigned int

Address of the access.

PAGESIZE unsigned int

Page size as $\log_2(\text{size})$.

PAS enum

PAS accessed is secure or nonsecure memory.

READ_PERM enum

Read Permission.

REG_NUM unsigned int

MPU register based region-number 0-15 (else 0xFF=none, 0xBB/0xDD=background/default).

SAU_REG_NUM enum

SAU region-number or code.

SAU_RGN_TYPE enum

Privacy Region Type.

SH enum

Shareability.

SIDE enum

Inst / Data.

WRITE_PERM enum

Write Permission.

PERIODIC

Called for every quantum. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The instruction count of this CPU.

PC unsigned int

The address of the next instruction to be executed on this CPU.

PMU_COUNTER_OVERFLOW

PMU counter overflow. Fields:

COUNTER_GROUP enum

Counter group.

COUNTER_RANGE enum

Counter range.

EVENT_ID unsigned int

Event Identifier, field is valid only for Event counter group.

INDEX unsigned int

Counter index (as selected by PMSELR), field is valid only for Event/Cycle counter groups.

INTERRUPT bool

Is interrupt enabled on overflow for this counter.

PMU_M_EVENT_AUT_RETIRED

PMU Event AUT_RETIRED.

PMU_M_EVENT_BF_CANCEL

PMU Event BF_CANCEL.

PMU_M_EVENT_BF_RETIRED

PMU Event BF_RETIRED.

PMU_M_EVENT_DWT_CMPMATCH0

PMU Event DWT_CMPMATCH0.

PMU_M_EVENT_DWT_CMPMATCH1

PMU Event DWT_CMPMATCH1.

PMU_M_EVENT_DWT_CMPMATCH2

PMU Event DWT_CMPMATCH2.

PMU_M_EVENT_DWT_CMPMATCH3

PMU Event DWT_CMPMATCH3.

PMU_M_EVENT_DWT_CMPMATCH4

PMU Event DWT_CMPMATCH4.

PMU_M_EVENT_DWT_CMPMATCH5

PMU Event DWT_CMPMATCH5.

PMU_M_EVENT_DWT_CMPMATCH6

PMU Event DWT_CMPMATCH6.

PMU_M_EVENT_DWT_CMPMATCH7

PMU Event DWT_CMPMATCH7.

PMU_M_EVENT_INST_SPEC

PMU Event INST_SPEC.

PMU_M_EVENT_LE_CANCEL

PMU Event LE_CANCEL.

PMU_M_EVENT_LE_RETIRED

PMU Event LE_RETIRED.

PMU_M_EVENT_MVE_FP_HP_RETIRED

PMU Event MVE_FP_HP_RETIRED.

PMU_M_EVENT_MVE_FP_HP_SPEC

PMU Event MVE_FP_HP_SPEC.

PMU_M_EVENT_MVE_FP_MAC_RETIRED

PMU Event MVE_FP_MAC_RETIRED.

PMU_M_EVENT_MVE_FP_MAC_SPEC

PMU Event MVE_FP_MAC_SPEC.

PMU_M_EVENT_MVE_FP_RETIRED

PMU Event MVE_FP_RETIRED.

PMU_M_EVENT_MVE_FP_SPEC

PMU Event MVE_FP_SPEC.

PMU_M_EVENT_MVE_FP_SP_RETIRED

PMU Event MVE_FP_SP_RETIRED.

PMU_M_EVENT_MVE_FP_SP_SPEC

PMU Event MVE_FP_SP_SPEC.

PMU_M_EVENT_MVE_INST_RETIRED

PMU Event MVE_INST_RETIRED.

PMU_M_EVENT_MVE_INST_SPEC

PMU Event MVE_INST_SPEC.

PMU_M_EVENT_MVE_INT_MAC_RETIRED

PMU Event MVE_INT_MAC_RETIRED.

PMU_M_EVENT_MVE_INT_MAC_SPEC

PMU Event MVE_INT_MAC_SPEC.

PMU_M_EVENT_MVE_INT_RETIRED

PMU Event MVE_INT_RETIRED.

PMU_M_EVENT_MVE_INT_SPEC

PMU Event MVE_INT_SPEC.

PMU_M_EVENT_MVE_LDST_MULTI_RETIRED

PMU Event MVE_LDST_MULTI_RETIRED.

PMU_M_EVENT_MVE_LDST_MULTI_SPEC

PMU Event MVE_LDST_MULTI_SPEC.

PMU_M_EVENT_MVE_LDST_RETIRE

PMU Event MVE_LDST_RETIRE.

PMU_M_EVENT_MVE_LDST_SPEC

PMU Event MVE_LDST_SPEC.

PMU_M_EVENT_MVE_LD_MULTI_RETIRE

PMU Event MVE_LD_MULTI_RETIRE.

PMU_M_EVENT_MVE_LD_MULTI_SPEC

PMU Event MVE_LD_MULTI_SPEC.

PMU_M_EVENT_MVE_LD_RETIRE

PMU Event MVE_LD_RETIRE.

PMU_M_EVENT_MVE_LD_SPEC

PMU Event MVE_LD_SPEC.

PMU_M_EVENT_MVE_PRED

PMU Event MVE_PRED.

PMU_M_EVENT_MVE_ST_MULTI_RETIRE

PMU Event MVE_ST_MULTI_RETIRE.

PMU_M_EVENT_MVE_ST_MULTI_SPEC

PMU Event MVE_ST_MULTI_SPEC.

PMU_M_EVENT_MVE_ST_RETIRE

PMU Event MVE_ST_RETIRE.

PMU_M_EVENT_MVE_ST_SPEC

PMU Event MVE_ST_SPEC.

PMU_M_EVENT_MVE_VREDUCE_FP_RETIRE

PMU Event MVE_VREDUCE_FP_RETIRE.

PMU_M_EVENT_MVE_VREDUCE_FP_SPEC

PMU Event MVE_VREDUCE_FP_SPEC.

PMU_M_EVENT_MVE_VREDUCE_INT_RETIRE

PMU Event MVE_VREDUCE_INT_RETIRE.

PMU_M_EVENT_MVE_VREDUCE_INT_SPEC

PMU Event MVE_VREDUCE_INT_SPEC.

PMU_M_EVENT_MVE_VREDUCE_RETIRE

PMU Event MVE_VREDUCE_RETIRE.

PMU_M_EVENT_MVE_VREDUCE_SPEC

PMU Event MVE_VREDUCE_SPEC.

PMU_M_EVENT_PAC_RETIRE

PMU Event PAC_RETIRE.

PMU_M_EVENT_SE_CALL_NS

PMU Event SE_CALL_NS.

PMU_M_EVENT_SE_CALL_S

PMU Event SE_CALL_S.

POTINDEX_UPDATE

Value of POT index for current pc changed. Fields:

NEW_POTINDEX unsigned int

New POTIndex.

OLD_POTINDEX unsigned int

Old POTIndex.

PPB_READ_ACCESS

Trace reads from the core to memory mapped registers in the PPB (private peripheral bus) address range. Fields:

ACCESS_SIZE unsigned int

Logarithm of the access size: 0=byte, 1=halfword, 2=word, 3=doubleword, ...

BURST_LEN unsigned int

The number of data transfers (beats) in this burst.

DATA unsigned int

The data transferred.

INNER_ATTRIBUTES unsigned int

The inner attributes: bit0=write allocate, bit1=read allocate, bit2=cachable, bit3=bufferable, bit4=shareable.

INST bool

Is an instruction fetch.

LOCK enum

Normal / Locked / Exclusive.

MANAGER_ID unsigned int

The AXI manager ID.

NS bool

Is a nonsecure access.

OUTER_ATTRIBUTES unsigned int

The outer attributes: bit0=write allocate, bit1=read allocate, bit2=cachable, bit3=bufferable, bit4=shareable.

PADDR unsigned int

Physical address of access.

PRIV bool

Is privileged access.

RESPONSE enum

Whether the transaction was successful, or an error occurred.

USER_FLAGS unsigned int

Core specific additional signals.

PPB_WRITE_ACCESS

Trace writes from the core to memory mapped registers in the PPB (private peripheral bus) address range. Fields:

ACCESS_SIZE unsigned int

Logarithm of the access size: 0=byte, 1=halfword, 2=word, 3=doubleword, ...

BURST_LEN unsigned int

The number of data transfers (beats) in this burst.

DATA unsigned int

The data transferred.

INNER_ATTRIBUTES unsigned int

The inner attributes: bit0=write allocate, bit1=read allocate, bit2=cachable, bit3=bufferable, bit4=shareable.

INST bool

Is an instruction fetch.

LOCK enum

Normal / Locked / Exclusive.

MANAGER_ID unsigned int

The AXI manager ID.

NS bool

Is a nonsecure access.

OUTER_ATTRIBUTES unsigned int

The outer attributes: bit0=write allocate, bit1=read allocate, bit2=cachable, bit3=bufferable, bit4=shareable.

PADDR unsigned int

Physical address of access.

PRIV bool

Is privileged access.

RESPONSE enum

Whether the transaction was successful, or an error occurred.

USER_FLAGS unsigned int

Core specific additional signals.

PRELOAD_DATA

Data preload from PLD instruction. Fields:

VADDR unsigned int

Virtual address of the data that should be preloaded.

PRELOAD_INST

Instruction preload from PLI instruction. Fields:

VADDR unsigned int

Virtual address of the instruction that should be preloaded.

PRE_CORE_LOAD

Trace just before a core load. Fields:

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

PRE_CORE_STORE

Trace just before a core store. Fields:

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

PSTATE_PACTIVE_OPMODE_TRANSITION

Trace pactive and Opmode mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Operating mode after the transition.

OLD_STATE enum

Operating mode before the transition.

PSTATE_PACTIVE_OPMODE_TRANSITION_DENIED

Trace the denied pactive and Opmode mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Operating mode after the transition.

OLD_STATE enum

Operating mode before the transition.

REASON string

Reason for the transition being denied.

PSTATE_PACTIVE_POWER_TRANSITION

Trace pactive and power mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Power mode after the transition.

OLD_STATE enum

Power mode before the transition.

PSTATE_PACTIVE_POWER_TRANSITION_DENIED

Trace the denied pactive and power mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Power mode after the transition.

OLD_STATE enum

Power mode before the transition.

REASON string

Reason for the transition being denied.

PSTATE_PACTIVE_THREAD_OPMODE_TRANSITION

Trace pactive and Opmode mode transitions for threads on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Operating mode after the transition.

OLD_STATE enum

Operating mode before the transition.

QUANTUM_START

Called at the start of every quantum.

RUN_STATE

Run state transition. Fields:

INST_COUNT unsigned int

Ticks count at point of transition.

NEW enum

New run state.

OLD enum

Old run state.

SEMIHOSTING_CALL

Call of a semihost function occurred. Fields:

PC unsigned int

The program counter after the semihosting call.

SEMIHOSTING_PRECALL

About to call semihost. Fields:

EL2 bool

Use EL2 translation regime.

NS bool

Is Non-Secure.

REG_WIDTH unsigned int

The current register width in bytes.

SIGNAL

External signal state change. Fields:

SIGNAL enum

Signal that changed.

STATE bool

Signal asserted state.

START_COMPILE

Compilation started. Fields:

VADDR unsigned int

Instruction where compilation begins.

STG_MULTIPLE_MICRO_OPS_ADDRESS_RANGE

Address range for MTE stores with multiple uops. Fields:

ADDR_END unsigned int

Final address.

ADDR_START unsigned int

Starting address.

STORE_MULTIPLE_REGS_ADDRESS_RANGE

Address range for stores with multiple registers. Fields:

ADDR_END unsigned int

Final address.

ADDR_START unsigned int

Starting address.

SUBOPTIMAL_LDST_RETIRED

suboptimal load/store. Fields:

ACCESS_SIZE unsigned int

Log2 of the access width in bytes used for alignment checking (1: half-word, 2: word ...).

OPTIMAL_ALIGN_SELECTOR unsigned int

Selector to decide the optimal alignment size (1: access width, 2: 4-bytes, 3: 8-bytes, ..., 12: 4-kbytes).

VADDR unsigned int

The virtual address of the access.

SYNC

Called for every synchronization, that is every quantum break. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The instruction count of this CPU.

LOCAL_QUANTUM unsigned int

The local quantum of this CPU.

LOCAL_TIME unsigned int

The local time of this CPU.

SYNC_REASON enum

Reason for sync.

SYSCALL

System call instruction executed. Fields:

IMM unsigned int

Immediate value of the system call instruction.

TYPE enum

System call type.

VADDR unsigned int

Instruction that caused the system call.

SYSREG_READ

Trace reads from system registers. Fields:

REG_NAME enum

Register number.

VALUE unsigned int

The value read.

SYSREG_UPDATE128

Triggers when the system updates a register. Fields:

REG enum

Register number.

UNKNOWN unsigned int

Bits of the register which became unknown.

VALUE unsigned int

Value written to the register.

SYSREG_UPDATE32

Triggers when the system updates a register. Fields:

REG enum

Register number.

UNKNOWN unsigned int

Bits of the register which became unknown.

VALUE unsigned int

Value written to the register.

SYSREG_UPDATE64

Triggers when the system updates a register. Fields:

REG enum

Register number.

UNKNOWN unsigned int

Bits of the register which became unknown.

VALUE unsigned int

Value written to the register.

SYSREG_WRITE

Trace write to system registers. Fields:

REG_ACCESS enum

Register access status.

REG_NAME enum

Register number.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

Value written to the register.

UNALIGNED_LDST_RETIRED

Processor unaligned load/store. Fields:

ACCESS_SIZE unsigned int

Log2 of the access width in bytes used for alignment checking (1=>half-word, 2->word ...).

NUMBER_OF_BEATS unsigned int

Number of accesses (beats) in this burst (Total size in bytes is ACCESS_SIZE * NUMBER_OF_BEATS).

VADDR unsigned int

The virtual address of the access.

VFP_D_REGS

VFP/NEON D 64 bit register write. Fields:

ALIASING enum

Type of register aliasing used.

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a write to the corresponding bit in the register. For example, 0x1 would indicate a write to VALUE[0].

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

VFP_D_REGS_READ

VFP/NEON D 64 bit register read. Fields:

ID unsigned int

The register number.

VALUE unsigned int

The current value of the register.

VFP_Q_REGS

VFP/NEON Q 128 bit register write. Fields:

ALIASING enum

Type of register aliasing used.

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a write to the corresponding byte in the register. For example, 0x1 would indicate a write to VALUE[7:0].

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

VFP_Q_REGS_READ

VFP/NEON Q 128 bit register read. Fields:

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a read to the corresponding byte in the register. For example, 0x1 would indicate a read to VALUE[7:0].

VALUE unsigned int

The current value of the register.

VFP_SYS_REGS

Writes to the VFP/NEON units system registers. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ID enum

Which VFP system register is written.

MASK unsigned int

The mask indicating updated cumulative bits.

OLD_VALUE unsigned int

The register's old value overwritten.

VALUE unsigned int

The new value written to the VFP system register.

VFP_S_REGS

VFP/NEON S 32 bit register write. Fields:

ALIASING enum

Type of register aliasing used.

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a write to the corresponding bit in the register. For example, 0x1 would indicate a write to VALUE[0].

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

VFP_S_REGS_READ

VFP/NEON S 32 bit register read. Fields:

ID unsigned int

The register number.

VALUE unsigned int

The current value of the register.

WAYPOINT

Signals end of basic block execution. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ISET enum

Origin instruction set.

IS_COND bool

Indicates if this is a conditional waypoint.

PC unsigned int

Origin address (or 0 if unavailable).

TAKEN bool

Indicates if this waypoint was taken.

TARGET unsigned int

Destination address.

TARGET_ISET enum

Destination instruction set.

WFE_END

WFE ended. Fields:

INST_COUNT unsigned int

Ticks count when leaving WFE.

WFE_EVENT_REGISTER

WFE event register status: set/clear, reason. Fields:

INST_COUNT unsigned int

Ticks count.

REASON enum

Reason for set/clear. Only REASON==0 (Cleared by WFE) clears the bit, all other reasons set it.

WFE_IGNORED

WFE ignored. Fields:

EVENT bool

This WFE was ignored because the event register was set.

INST_COUNT unsigned int

Ticks count when ignoring WFE.

TRAPPED bool

This WFE was trapped.

WFE_START

WFE entered. Fields:

INST_COUNT unsigned int

Ticks count when entering WFE.

WFI_END

WFI ended. Fields:

INST_COUNT unsigned int

Ticks count when leaving WFI.

WFI_IGNORED

WFI ignored. Fields:

DISABLED bool

This WFI was ignored because WFI is disabled.

INST_COUNT unsigned int

Ticks count when ignoring WFI.

REASON enum

specifies reason why WFI trace was ignored.

TRAPPED bool

This WFI was trapped.

WFI_START

WFI entered. Fields:

INST_COUNT unsigned int

Ticks count when entering WFI.

WFI_WAKEUP

WFI wakeup. Fields:

INST_COUNT unsigned int

Ticks count when WFI wakeup occurred.

REASON enum

Reason for wakeup.

XPSR

Changes to the xPSR register. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

OLD_VALUE unsigned int

The old xPSR value.

UNKNOWN unsigned int

Bits within the register that have unknown value.

VALUE unsigned int

The new xPSR value.

2.46 ARM_Cortex-M55

This section describes the trace sources.

ASYNC_MEMORY_FAULT

Context ID Register write. Fields:

FAULT unsigned int

Fault status present in ESR.ISS encoding for Serror in ARCH64 and present in DFSR encoding for external abort format in ARCH32.

PADDR unsigned int

Physical Address (or 0 if unavailable).

ATOMIC_END_ACCESS

Bus trace access for atomics. Fields:

ACCESS_FAIL bool

Memory access failed.

ADDR unsigned int

The virtual address of the access.

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate

when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

COMPARE_VALUE unsigned int

Compare value for CAS.

LOAD_VALUE unsigned int

Loaded values.

NSDESC unsigned int

The security state of the access.

OPERAND_VALUE unsigned int

Operation's operand.

OPERATION enum

Operation type.

PADDR unsigned int

The physical address of the access.

PRIV bool

Is this a privileged access?.

ATOMIC_SLAVE_ACCESS

Information on the atomic operation at slave side. Fields:

ACCESS_RESULT unsigned int

Result of atomic access.

COMPARE_VALUE unsigned int

Value to be compared with memory by CAS.

LOAD_VALUE unsigned int

Value to be loaded from memory as a result of atomic operation.

MANAGER unsigned int

ID of manager which initiated the transaction.

NS bool

Transaction is Non-Secure.

OPERAND_VALUE unsigned int

Incoming operand value provided by core.

OPERATION enum

Type of atomic operation.

PADDR unsigned int

Full physical address of TCM.

STORE_VALUE unsigned int

Value to be stored to memory as a result of atomic operation.

ATOMIC_START_ACCESS

Bus trace access for atomics. Fields:

ADDR unsigned int

The virtual address of the access.

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

COMPARE_VALUE unsigned int

Compare value for CAS.

NSDESC unsigned int

The security state of the access.

OPERAND_VALUE unsigned int

Operation's operand.

OPERATION enum

Operation type.

PADDR unsigned int

The physical address of the access.

PRIV bool

Is this a privileged access?.

ArchMsg.Error.exit_code

DISPLAY exit code: %{EXITCODE} component kind: %{KIND} reason: %{REASON}. Fields:

EXITCODE signed int

Exit code to be returned.

KIND string

Component kind that invoked the exit code trace.

REASON string

Reason why the exit code trace was invoked (optional).

ArchMsg.Warning.Unpredictably Indexed PM Event Register

TAGS UNPREDICTABLE DISPLAY Event %{IsDirect:(Selector value|Register)} %{SEL} is >= %{N}. Fields:

IsDirect unsigned int

Direct Access rather than indirect.

N unsigned int

Number of Accessible Registers.

SEL unsigned int

Selector.

ArchMsg.Warning.cache_contents_unknown

A cache has been activated without being invalidated and may contain **UNKNOWN** content. DISPLAY **%{SIDE:(D|I)}**-Cache was enabled in **%{REGIME:(SECURE|NON_SECURE|HYP)}** regime but was not invalidated since power-on: cache lines could contain **UNKNOWN** content **%{SIDE:(** Note: D-side caches must invalidate SECURE lines even if the cache is only used by NON_SECURE code in order to ensure that unexpected dirty lines tagged SECURE are not naturally evicted**)}**. Fields:

REGIME unsigned int

Current translation regime.

SIDE unsigned int

0 d-side cache or unified cache, 1 i-side.

ArchMsg.Warning.dap_csw_bad_size

A write to CM3DAP CSW has an invalid size field. DISPLAY Write **%{DATA}** to CM3DAP CSW has invalid size field. Fields:

DATA unsigned int

bits[2:0] are size.

ArchMsg.Warning.dcimvac_matches_watchpoint

TAGS **UNPREDICTABLE** DISPLAY Watchpoints matching an AArch32 DCIMVA are implementation defined.

ArchMsg.Warning.decode_fieldmismatch

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with different fields **%{FIELD1}** and **%{FIELD2}**. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_frachitsnegative

TAGS **UNPREDICTABLE** DISPLAY VCVT Instruction is unpredictable with a negative number of fraction bits.

ArchMsg.Warning.decode_initblock

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable in an IT block.

ArchMsg.Warning.decode_initblocknotlast

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable in an IT block when not the last.

ArchMsg.Warning.decode_invalid_condition

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable when the condition is not AL.

ArchMsg.Warning.decode_invalidfieldcombination

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with the values in %{FIELD1} and %{FIELD2}. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_invalidvalue

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with the value in %{FIELD}. Fields:

FIELD enum

Field name.

ArchMsg.Warning.decode_mem_hint_unallocated

TAGS **UNPREDICTABLE** DISPLAY There is no memory hint allocated to this bit pattern.

ArchMsg.Warning.decode_registermatch

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with %{FIELD1} and %{FIELD2} as the same register. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_registermismatch

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with %{FIELD1} and %{FIELD2} as different registers. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_registeroutofrange

TAGS **UNPREDICTABLE** DISPLAY Floating-point instruction is unpredictable indexing beyond the end of the register bank.

ArchMsg.Warning.decode_sbzsbo

TAGS **UNPREDICTABLE** DISPLAY Reserved bits in the instruction are not canonical.

ArchMsg.Warning.decode_transactiontoobig

TAGS **UNPREDICTABLE** DISPLAY Attempt to perform a memory transaction of more than 128 bytes.

ArchMsg.Warning.decode_unpred_other

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable.

ArchMsg.Warning.decode_unpreduseofpc

TAGS UNPREDICTABLE DISPLAY Instruction is unpredictable when %{FIELD} is PC. Fields:

FIELD enum

Field name.

ArchMsg.Warning.decode_unpreduseofr13

TAGS UNPREDICTABLE DISPLAY Instruction is unpredictable when %{FIELD} is R13. Fields:

FIELD enum

Field name.

ArchMsg.Warning.decode_unpreduseofr13inreglist

TAGS UNPREDICTABLE DISPLAY Instruction is unpredictable with R13 in register-list.

ArchMsg.Warning.decode_useofd16orhigher

TAGS UNPREDICTABLE DISPLAY Attempt to use register D16 or higher where it is not allowed.

ArchMsg.Warning.decode_writebackandbaseinlist

TAGS UNPREDICTABLE DISPLAY Instruction is unpredictable with write back when the base register is in the transfer list.

ArchMsg.Warning.decode_zeroregistersinlist

TAGS UNPREDICTABLE DISPLAY Instruction is unpredictable when no registers are to be transferred.

ArchMsg.Warning.extension_unpred

Probably unpred in IT state or decod. Fields:

PC unsigned int

Address of this instruction.

ArchMsg.Warning.msr_zeroes_ltpsize

May be pre-MVE setup code.

ArchMsg.Warning.pmu_and_dwt

DISPLAY Trying to access a %{IS_PMU:(DWT|PMU)} register while %{IS_PMU:(PMU|DWT)} is also active. Fields:

IS_PMU bool

Whether trying to access the PMU or the DWT.

ArchMsg.Warning.recursive_branch

An instruction is performing a branch that targets the same instruction. If this was intended then a WFI might be more effective DISPLAY recursive branch detected at PC=%{PC}. PRIMARY KEY PC. Fields:

PC unsigned int

Address of instruction causing the branch.

ArchMsg.Warning.recursive_exception

An instruction has generated an exception that targets the same instruction. DISPLAY recursive %{TYPE} exception detected at PC=%{PC} (CPSR=%{CPSR}). PRIMARY KEY PC. Fields:

CPSR unsigned int

Processor state.

PC unsigned int

Address of instruction causing the exception.

TYPE enum

Type of exception.

ArchMsg.Warning.reentrant_vector_catch

TAGS **UNPREDICTABLE** DISPLAY Vector catch debug event on Prefetch or Data abort vector %{ADDR}. Fields:

ADDR unsigned int

vector address.

ArchMsg.Warning.reserved_it_state

TAGS **UNPREDICTABLE** DISPLAY Execution with reserved IT state %{ITSTATE}. Fields:

ITSTATE unsigned int

reserved IT state.

ArchMsg.Warning.secure_vector_fetch_from_nonsecure

DISPLAY Secure vector table is being fetched from Non-secure memory.

ArchMsg.Warning.sg_in_it_block

Arm recommends that software does not place SG (security gate instruction) inside an IT instruction block.

ArchMsg.Warning.unpred_FAR_on_LRCPC3_non_atomic_mem_access

TAGS **UNPREDICTABLE** Unpredictable FAR/FSTATUS on LRCPC3 pair instructions resulting in two separate memory accessesDISPLAY Unpredictable FAR on LRCPC3 non atomic memory access at %{ADDR1} with FSTATUS %{FSTATUS1} or %{ADDR2} with FSTATUS %{FSTATUS2}. Fields:

ADDR1 unsigned int

Possible FAR address 1.

ADDR2 unsigned int

Possible FAR address 2.

FSTATUS1 unsigned int

Possible fault status value 1.

FSTATUS2 unsigned int

Possible fault status value 2.

ArchMsg.Warning.unpred_async_tag_check_fail_on_sve_speculative_load_do_update_ffr

Unpredictable set to FALSE of the FFR element associated to SVE speculative load with an async tag check fail TAGS **UNPREDICTABLE**.

ArchMsg.Warning.unpredictable_a32_breakpoint

TAGS **UNPREDICTABLE** DISPLAY Unpredictable breakpoint on a A32 instruction at %{ADDR1} Offset %{OFFSET} BAS %{BAS} for a %{IS_ADDRESS_MISMATCH_BREAKPOINT:(Mismatch|Match)} Breakpoint number %{N:d} Breakpoint address %{ADDR2}. Fields:

ADDR1 unsigned int

Address.

ADDR2 unsigned int

Breakpoint address.

BAS unsigned int

BAS field.

IS_ADDRESS_MISMATCH_BREAKPOINT bool

Is a Mismatch breakpoint, not a Match breakpoint.

N unsigned int

Breakpoint number.

OFFSET unsigned int

Always has a value of zero.

ArchMsg.Warning.unpredictable_banked_register_access

TAGS **UNPREDICTABLE** DISPLAY Access to banked register (R=%{R:d} SYSm=%{SYSM:2x}) is unpredictable. Fields:

R unsigned int

R bit.

SYSM unsigned int

SYSm.

ArchMsg.Warning.unpredictable_ccfail_undef

TAGS **UNPREDICTABLE** DISPLAY An instruction %{OPCODE} at VA=%{VADDR} failed its condition codes check but would otherwise have caused a non-data-dependent trap or undefined exception. Behaviour is implementation-defined choice of exception or nop. Fields:

OPCODE unsigned int

opcode of the instruction.

VADDR unsigned int

Virtual address of the instruction being executed.

ArchMsg.Warning.unpredictable_ici_bits

TAGS **UNPREDICTABLE** DISPLAY ICI bits do not apply or are architecturally unpredictable for instruction.

ArchMsg.Warning.unpredictable_t32_breakpoint

TAGS **UNPREDICTABLE** DISPLAY Unpredictable breakpoint on a T32 instruction at %{ADDR1} Offset %{OFFSET} BAS %{BAS} for a %{IS_ADDRESS_MISMATCH_BREAKPOINT:(Mismatch|Match)} Breakpoint number %{N:d} Breakpoint address %{ADDR2}. Fields:

ADDR1 unsigned int

Address.

ADDR2 unsigned int

Breakpoint address.

BAS unsigned int

BAS field.

IS_ADDRESS_MISMATCH_BREAKPOINT bool

Is a Mismatch breakpoint, not a Match breakpoint.

N unsigned int

Breakpoint number.

OFFSET unsigned int

Indicates which halfword of a 32-bit Thumb instruction the breakpoint is set on. 0 for the first halfword or if the instruction is 16 bits wide, or 1 for the second halfword.

ArchMsg.Warning.unpredictable_t32_vector_catch

TAGS **UNPREDICTABLE** DISPLAY Unpredictable Vector catch exception on a T32 instruction at %{ADDR} Offset %{OFFSET}. Fields:

ADDR unsigned int

Address.

OFFSET unsigned int

Indicates which halfword of a 32-bit Thumb instruction the vector catch exception is set on. 0 for the first halfword or if the instruction is 16 bits wide, or 1 for the second halfword.

ArchMsg.Warning.unpredictable_watchpoint_far

TAGS **UNPREDICTABLE** DISPLAY The value for FAR/EDWAR for hitting this watchpoint may be anywhere in the range %{LOWER_BOUND} to %{UPPER_BOUND}. Fields:

LOWER_BOUND unsigned int

the lowest address accessed by the instruction that triggered the watchpoint.

UPPER_BOUND unsigned int

the highest watchpointed address accessed by that instruction.

ArchMsg.Warning.warning_AdvSIMDExpandImmUnexpectedZero

TAGS **UNPREDICTABLE** DISPLAY AdvSIMDExpandImm may treat this immediate value as **UNPREDICTABLE**.

ArchMsg.Warning.warning_ConditionalSMC

TAGS **UNPREDICTABLE** DISPLAY SMC instruction has **UNPREDICTABLE** effects when conditional and when combined with traps.

ArchMsg.Warning.warning_access_crosses_page_boundary

TAGS **UNPREDICTABLE** DISPLAY Access crosses page boundary.

ArchMsg.Warning.warning_access_crosses_page_boundary_has_fault

TAGS **UNPREDICTABLE** DISPLAY Access that crosses page boundary has fault(s).

ArchMsg.Warning.warning_access_crosses_page_boundary_has_fault_on_both_pages

TAGS **UNPREDICTABLE** DISPLAY Access that crosses page boundary has a fault on each page.
Fields:

ADDR1 unsigned int

FAR address on first page.

ADDR2 unsigned int

FAR address on second page.

FSTATUS1 unsigned int

Fault status on first page.

FSTATUS2 unsigned int

Fault status on second page.

ArchMsg.Warning.warning_access_crosses_page_boundary_spanning_different_memory

TAGS **UNPREDICTABLE** DISPLAY Access at address %{ADDR} crosses page from MEM_TYPE %{MEMTYPE_PAGE1} to %{MEMTYPE_PAGE2}. Fields:

ADDR unsigned int

address of access crossing page.

MEMTYPE_PAGE1 signed int

First Page's memory type (0-Normal, 1-Non-Normal).

MEMTYPE_PAGE2 signed int

Second Page's memory type (0-Normal, 1-Non-Normal).

ArchMsg.Warning.warning_access_wraps_around_memory

TAGS **UNPREDICTABLE** DISPLAY Access wraps around memory in %{WIDTH:d}bit mode.
Fields:

WIDTH unsigned int

Address width.

ArchMsg.Warning.warning_bcr_linking_status

TAGS **UNPREDICTABLE** DISPLAY DBGBCR.{BT2,BT} combination gives unpredictable behaviour.

ArchMsg.Warning.warning_bcr_mask_reserved

TAGS **UNPREDICTABLE** DISPLAY DBGBCR_ELn.MASK is set to a reserved value.

ArchMsg.Warning.warning_bvr_masked_bit_not_zero

TAGS **UNPREDICTABLE** DISPLAY DBGBVRn_EL1.VA contains a bit masked by DBGBCRn_EL1.MASK that is not zero.

ArchMsg.Warning.warning_ccsidr_unimplemented_level

TAGS **UNPREDICTABLE** DISPLAY CCSIDR read while CSSELR %{CSSELR:x} points to an unimplemented cache level. Fields:

CSSELR unsigned int

current effective value of CSSELR.

ArchMsg.Warning.warning_change_to_ns_when_tge_set

TAGS **UNPREDICTABLE** DISPLAY Attempting to change NS to 1 when HCR.TGE = 1.

ArchMsg.Warning.warning_contiguous_bit_check_abort

TAGS **UNPREDICTABLE** DISPLAY An abort occurred whilst attempting to check TLB entry is contiguous at %{ENTRY_ADDR}. Fields:

ENTRY_ADDR unsigned int

address of conflicting TLB entry.

ArchMsg.Warning.warning_contiguous_bit_error

TAGS **UNPREDICTABLE** DISPLAY TLB entry at %{ENTRY_ADDR} was not contiguous with entry at %{CONTIG_ADDR} Contents are %{ENTRY_DATA} but expected %{CONTIG_DATA}. Fields:

CONTIG_ADDR unsigned int

address of the TLB first read, with which this entry is expected to be contiguous.

CONTIG_DATA unsigned int

expected contents based on the entry at CONTIG_ADDR.

ENTRY_ADDR unsigned int

address of conflicting TLB entry.

ENTRY_DATA unsigned int

contents of conflicting TLB entry.

ArchMsg.Warning.warning_contiguous_bit_error_D128

TAGS **UNPREDICTABLE** DISPLAY TLB entry at %{ENTRY_ADDR} was not contiguous with entry at %{CONTIG_ADDR} Contents are %{ENTRY_DATA_HI} %{ENTRY_DATA_LO} but expected %{CONTIG_DATA}. Fields:

CONTIG_ADDR unsigned int

address of the TLB first read, with which this entry is expected to be contiguous.

CONTIG_DATA unsigned int

expected contents based on the entry at CONTIG_ADDR.

ENTRY_ADDR unsigned int

address of conflicting TLB entry.

ENTRY_DATA_HI unsigned int

Higher bit contents of conflicting TLB entry.

ENTRY_DATA_LO unsigned int

Lower bit contents of conflicting TLB entry.

ArchMsg.Warning.warning_cp10_cp11_mismatch

TAGS **UNPREDICTABLE** DISPLAY Unpredictable configuration of CP10 and CP11 controls at EL
%{EL:d}. Fields:

EL unsigned int

EL owning the control.

ArchMsg.Warning.warning_cp10_cp11_reserved_value

TAGS **UNPREDICTABLE** DISPLAY Unpredictable configuration of CPACR CP10 (%{CP10}) or
CP11 (%{CP11}). Fields:

CP10 unsigned int

CP10 access.

CP11 unsigned int

CP11 access.

ArchMsg.Warning.warning_csselr_level_out_of_range

TAGS **UNPREDICTABLE** DISPLAY CSSELR.Level out of range, written value %{LEVEL} with only
%{IMPLEMENTED:d} levels implemented. Fields:

IMPLEMENTED unsigned int

number of cache levels implemented.

LEVEL unsigned int

written value of CSSELR.Level.

TYPE enum

The type of cache selected by CSSELR.InD and TnD.

ArchMsg.Warning.warning_debug_flow_control_bits_not_obeyed

TAGS **UNPREDICTABLE** DISPLAY Flow control bits for %{REGISTER} were not obeyed, causing
%{ERROR}. Fields:

ERROR enum

type of error resulting.

REGISTER enum

register under consideration.

ArchMsg.Warning.warning_debug_register_access_during_reset

TAGS **UNPREDICTABLE** DISPLAY Unpredictable %{IS_WRITE:(read|write)} access to debug
register offset %{OFFSET} during reset. Fields:

IS_WRITE unsigned int

Write Not Read.

OFFSET unsigned int

Register Offset.

ArchMsg.Warning.warning_decode_cps_inconsistent_fields

TAGS **UNPREDICTABLE** DISPLAY The mode setting fields in the CPS instruction are inconsistent.

ArchMsg.Warning.warning_decode_invalid_state

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable in current exception-level/security state.

ArchMsg.Warning.warning_default_cacheable_mmu_on

TAGS **UNPREDICTABLE** DISPLAY Default cacheable is enabled (HCR.DC = 1) while MMU is enabled.

ArchMsg.Warning.warning_default_cacheable_vmmu_off

TAGS **UNPREDICTABLE** DISPLAY Default cacheable is enabled (HCR.DC = 1) while HCR.VM = 0.

ArchMsg.Warning.warning_deprecated_wvr_bit_2_set

TAGS **UNPREDICTABLE** DISPLAY DBGWVR%{N:d}_EL1=%{WVR:h} has bit 2 set, deprecated in ARMv8. Fields:

N unsigned int

index.

WVR unsigned int

DBGWVR.

ArchMsg.Warning.warning_eret_while_software_step_active_pending

Missing ISB between setting MDSCR_EL1.SS and ERET TAGS **UNPREDICTABLE**.

ArchMsg.Warning.warning_exclusive_to_non_normal

TAGS **UNPREDICTABLE** DISPLAY Exclusive Access to Non-Normal Memory %{ADDR}. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_exclusive_to_non_writeback

TAGS **UNPREDICTABLE** DISPLAY Exclusive Access to Non-Writeback-Cacheable Memory %{ADDR}. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_execute_from_device_memory

TAGS **UNPREDICTABLE** DISPLAY An attempt was made to execute from Device memory.

ArchMsg.Warning.warning_illegal_cpsr_mode

TAGS **UNPREDICTABLE** DISPLAY Writing an illegal or unimplement mode (%{NEW_MODE:h}) to CPSR was unpredictable. Fields:

NEW_MODE unsigned int

New value of CPSR.M.

ArchMsg.Warning.warning_illegal_srs_mode

TAGS **UNPREDICTABLE** DISPLAY Illegal or **UNPREDICTABLE** mode (%{MODE:h}) used for SRS instruction. Fields:

MODE unsigned int

mode for Banked SP.

ArchMsg.Warning.warning_implementation_defined_read_debug_register_in_SCS

Read Access to Debug registers from the processor is **IMPLEMENTATION DEFINED** TAGS **IMP_DEF**. Fields:

PPB_OFFSET unsigned int

debug register offset.

ArchMsg.Warning.warning_implementation_defined_sequential_security_transitions_supported

The behavior of sequential instruction fetches that cross from non-secure to secure memory and contain an SG instruction is **CONSTRAINED UNPREDICTABLE** TAGS **UNPREDICTABLE**.

ArchMsg.Warning.warning_implementation_defined_stack_limit_check_supported

It is **IMPLEMENTATION DEFINED** whether stack pointer limit checking is performed for this instruction TAGS **IMP_DEF**. Fields:

ADDRESS unsigned int

Address of instruction.

OPCODE unsigned int

instruction opcode.

ArchMsg.Warning.warning_implementation_defined_taggednonsharedmemorytype

TAGS **IMP_DEF** DISPLAY Memory attribute has Tagged, non shared memory type; it is **IMPLEMENTATION DEFINED** whether the memory location is treated as Tagged or non-Tagged.

ArchMsg.Warning.warning_implementation_defined_write_debug_register_in_SCS

Write Access to Debug registers from the processor is **IMPLEMENTATION DEFINED** TAGS **IMP_DEF**. Fields:

DATA unsigned int

data attempted to be written.

PPB_OFFSET unsigned int

debug register offset.

ArchMsg.Warning.warning_invalid_tcr_granule

TAGS **UNPREDICTABLE** DISPLAY TCR.TG%{TG_ID:(0|1)} bits have been set to a granule size %{REQUEST:d}K not implemented - using %{SUBSTITUTE:d}K. Fields:

REQUEST signed int

page size requested (or 0 for reserved).

SUBSTITUTE signed int

best guess available page size.

TG_ID bool

bits TG0 or TG1.

ArchMsg.Warning.warning_load_multiple_user_registers_from_user_mode

TAGS UNPREDICTABLE DISPLAY An LDM(user registers) instruction executed from user mode.

ArchMsg.Warning.warning_mte_in_mem_attr_not_supported

TAGS UNPREDICTABLE DISPLAY Memory attribute %{ATTR} has Tagged memory type but Armv8.5-MTE is not implemented. Fields:

ATTR unsigned int

Memory attribute.

ArchMsg.Warning.warning_reserved_breakpoint_state_match

TAGS UNPREDICTABLE DISPLAY %{IS_BREAKPOINT:(Watchpoint|Breakpoint)} programmed with a reserved combination of HMC, SSC and %{IS_BREAKPOINT:(PAC|PMC)}. Fields:

IS_BREAKPOINT unsigned int

Is a breakpoint, not a watchpoint.

ArchMsg.Warning.warning_shareability

TAGS UNPREDICTABLE DISPLAY Unpredictable: combination of the 1st and 2nd stages of translation is Normal Inner Non-Cacheable, Outer Non-Cacheable.

ArchMsg.Warning.warning_software_step_set_while_enabled

TAGS UNPREDICTABLE DISPLAY MDSCR_EL1.SS set to 1 while software step debug exceptions are enabled.

ArchMsg.Warning.warning_thumb_instruction_wraps_around_memory

TAGS UNPREDICTABLE DISPLAY Thumb instruction wraps around memory.

ArchMsg.Warning.warning_ttbr_sbz_bits_are_not_zero

TAGS UNPREDICTABLE DISPLAY %{TTBR}[%{MSB:d};0] should be zero, but are not. Fields:

MSB unsigned int

x-1.

TTBR enum

which TTBR register.

ArchMsg.Warning.warning_unaligned_address_dbgdtrrx_write

TAGS UNPREDICTABLE DISPLAY Unaligned address %{ADDR} in X0/R0 when DBGDTRRX written in memory access mode. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unaligned_address_dbgdtrtx_read

TAGS **UNPREDICTABLE** DISPLAY Unaligned address %{ADDR} in X0/R0 when DBGDTRTX read in memory access mode. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unaligned_to_device

TAGS **UNPREDICTABLE** DISPLAY Unaligned Access to Device Memory %{ADDR}. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unaligned_to_strongly_ordered

TAGS **UNPREDICTABLE** DISPLAY Unaligned Access to Strongly Ordered Memory %{ADDR}. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unknown_sau_rnr

TAGS **UNPREDICTABLE** DISPLAY SAU_RNR was set to an unsupported value.

ArchMsg.Warning.warning_unpredictable_AIRCR_PRIS_and_BFHFNMINs

TAGS **UNPREDICTABLE** DISPLAY The effect of setting both AIRCR.BFHFNMINs and AIRCR.PRIS to 1 is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_AIRCR_VECTCLRACTIVE_when_not_in_debug

TAGS **UNPREDICTABLE** DISPLAY The effect of writing a 1 to AIRCR.VECTCLRACTIVE if the processor is not halted in Debug state is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_AIRCR_VECTRESET_and_SYSRESETREQ

TAGS **UNPREDICTABLE** DISPLAY When the processor is halted in Debug state, if a write to the register writes a 1 to both VECTRESET and SYSRESETREQ, the behavior is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_AIRCR_VECTRESET_when_not_in_debug

TAGS **UNPREDICTABLE** DISPLAY The effect of writing a 1 to AIRCR.VECTRESET if the processor is not halted in Debug state is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_AIRCR_incorrect_VKEY

TAGS **UNPREDICTABLE** DISPLAY The value 0x05FA must be written to AIRCR.VKEY, otherwise the register write is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_EXC_RETURN_Reserved_Bit

EXC_RETURN[23:7] are reserved with the special condition that all bits should be written as one. Values other than all 1s are **UNPREDICTABLE**. TAGS **UNPREDICTABLE**. Fields:

EXC_RETURN unsigned int

Special PC value.

SBO_MASK unsigned int

Which bits Should Be One to not UNPRED.

ArchMsg.Warning.warning_unpredictable_PMEVTYPErn_ELO_VS_reserved_value

TAGS UNPREDICTABLE DISPLAY Unpredictable reserved value of PMEVTYPER<n>_ELO.VS.

ArchMsg.Warning.warning_unpredictable_change_to_DEMCR_MON_STEP_at_insufficient_priority

TAGS UNPREDICTABLE DISPLAY The effect of changing DEMCR.MON_STEP at an execution priority that is lower than the priority of the DebugMonitor exception is **CONSTRAINED UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_change_to_priority_of_active_exception

TAGS UNPREDICTABLE DISPLAY Changing the priority of an active exception is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_clear_lspact

TAGS UNPREDICTABLE DISPLAY **UNPREDICTABLE** state on context switch. Fields:

DRVEXC_TAKEN unsigned int

Derived exception taken, not pended.

REACHED_LAST_STORE unsigned int

Only the last faulted.

ArchMsg.Warning.warning_unpredictable_dbg_exit_unaligned_dlr

TAGS UNPREDICTABLE DISPLAY Debug exit to AArch32 with DLR[0] set to 1. PC[0] can be set to 0 or to DLR[0].

ArchMsg.Warning.warning_unpredictable_exception_catch

TAGS UNPREDICTABLE DISPLAY Generation of an exception catch event is unpredictable.

ArchMsg.Warning.warning_unpredictable_exception_return_inconsistent_state

TAGS UNPREDICTABLE DISPLAY state on exception return is unpredictable.

ArchMsg.Warning.warning_unpredictable_exception_return_instruction

TAGS UNPREDICTABLE DISPLAY exception return instruction is unpredictable.

ArchMsg.Warning.warning_unpredictable_in_debug_state

TAGS UNPREDICTABLE DISPLAY Instruction is **UNPREDICTABLE** when executed in debug state.

ArchMsg.Warning.warning_unpredictable_mair_encoding

TAGS UNPREDICTABLE DISPLAY Memory attribute %{ATTR} has invalid MAIR encoding; this may lead to unpredictable results. Fields:

ATTR unsigned int

Memory attribute.

ArchMsg.Warning.warning_unpredictable_pmu_counter_access

TAGS UNPREDICTABLE DISPLAY Access to PMU counters from non-secure ELO or EL1 is **UNPREDICTABLE** with MDCR_EL2.HPMN set to 0.

ArchMsg.Warning.warning_unpredictable_prioritization_breakpoint_match_vector_catch

TAGS UNPREDICTABLE DISPLAY Two events with the same priority occurred at the same instruction: Address Matching Vector Catch debug event, and Breakpoint debug event. It is constrained unpredictable which is taken.

ArchMsg.Warning.warning_unpredictable_stack_selection

TAGS **UNPREDICTABLE** DISPLAY **UNPREDICTABLE** state on context switch.

ArchMsg.Warning.warning_unpredictable_tsize_out_of_range

TAGS **UNPREDICTABLE** DISPLAY **%{TCR}.T%{TTBR:d}SZ (%{TSIZE:d} bits)** is out of range. Must be between 25 and **%{TSIZE_MAX:d}**. Fields:

TCR enum

Which TCR.

TSIZE signed int

Value of TSize.

TSIZE_MAX signed int

The maximum allowed value.

TTBR unsigned int

Which TTBR region.

ArchMsg.Warning.warning_unpredictable_unaligned_pc_as_base_register

TAGS **UNPREDICTABLE** DISPLAY The PC value must be word-aligned, otherwise the behavior of the instruction is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_unaligned_pop_stack

TAGS **UNPREDICTABLE** DISPLAY **CONSTRAINED UNPREDICTABLE** unaligned stack frame **%{FRAME_ADDR:x}** in exception-return. Fields:

FRAME_ADDR unsigned int

unaligned stack frame address.

ArchMsg.Warning.warning_unpredictable_unaligned_to_non_writeback

TAGS **UNPREDICTABLE** DISPLAY Unaligned Access to Non-Writeback-Cacheable Memory.

ArchMsg.Warning.warning_unpredictable_vmsa_memattrib

TAGS **UNPREDICTABLE** DISPLAY Unpredictable vmsa memory attribute with texcb value programmed to **%{TEXCB}** when tex remap is **%{TEX_REMAP}**. Fields:

TEXCB unsigned int

texcb value.

TEX_REMAP bool

use tex remap.

ArchMsg.Warning.warning_unpredictable_vtcr_t0sz_sl0

TAGS **UNPREDICTABLE** DISPLAY The combination of **VTCR.TOSZ=%{TOSZ:d}** and **VTCR.SL0=%{SL0:d}** is unpredictable. Fields:

SL0 signed int

VTCR.SL0, starting level for stage 2 translation table walks.

TOSZ signed int

VTCR.TOSZ, required input address range.

ArchMsg.Warning.warning_unpredictable_write_DHCSR

TAGS UNPREDICTABLE DISPLAY UNPREDICTABLE set of bit-changes in DHCSR. Fields:

NEW unsigned int

data attempted to be written.

OLD unsigned int

previous value.

ArchMsg.Warning.warning_unpredictable_write_SHCSR

TAGS UNPREDICTABLE DISPLAY UNPREDICTABLE setting a pending bit to 1 for an exception with priority >= execution priority.

ArchMsg.Warning.warning_unsupported_access_to_memory_mapped_register

TAGS UNPREDICTABLE DISPLAY Access to memory mapped register is unsupported.

ArchMsg.Warning.warning_user_jmcr_access

TAGS UNPREDICTABLE DISPLAY User %{WRITE:(write|read)} access to JMCR is unpredictable. Fields:

WRITE unsigned int

access is write.

ArchMsg.Warning.warning_wcr_mask_and_bas

TAGS UNPREDICTABLE DISPLAY DBGWCR_EL1.MASK is non-zero and BAS is not 0xff.

ArchMsg.Warning.warning_wcr_mask_reserved

TAGS UNPREDICTABLE DISPLAY DBGWCR_EL1.MASK is set to a reserved value.

ArchMsg.Warning.warning_wcr_non_configuous_bas

TAGS UNPREDICTABLE DISPLAY DBGWCR_EL1.BAS has non-contiguous set of ones.

ArchMsg.Warning.warning_wvr_masked_bit_not_zero

TAGS UNPREDICTABLE DISPLAY DBGWVRn_EL1.VA contains a bit masked by DBGWCRn_EL1.MASK that is not zero.

ArchMsg.warning_non_architectural_beat_order

DISPLAY Current set of committed beats does not give legal ECI value.

ArchMsg.warning_unpred_system_register_access

DISPLAY Unpredictable %{IS_WRITE:(read|write)} access to system register offset %{OFFSET}. Fields:

IS_WRITE unsigned int

Write Not Read.

OFFSET unsigned int

Register Offset.

BEAT_EXCEPTION

Exception during of one beat of an M. Fields:

BEAT unsigned int

Beat number 0..3.

EXC unsigned int

Internal representation of the excep.

EXCEPTION_PC unsigned int

Address of older in-flight instructi.

INST unsigned int

This instruction as FirstHalfWord:Se.

INST_ADDR unsigned int

Address of faulting instruction inst.

BEAT_START

Start of one beat of an MVE Vector i. Fields:

BEAT unsigned int

Beat number 0..3.

DISASS string

Disassembly of instruction.

ELMT_MASK unsigned int

Element Mask.

EXCEPTION_PC unsigned int

Address of older in-flight instructi.

INST unsigned int

This instruction as FirstHalfWord:Se.

INST_ADDR unsigned int

Address of this instruction.

LTP_MASK unsigned int

Element Mask from Loop Tail Predicat.

VPT_MASK unsigned int

Element Mask from VPR.

BRANCH_MISPREDICT

Simulating branch mispredict. Fields:

PC unsigned int

Origin address (or 0 if unavailable).

BRA_DIR

Direct branches, to immediate address. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISET enum

The instruction set of the branch instruction.

IS_COND bool

Indicates if this is a conditional branch.

IS_HINTED bool

Whether the conditional instruction is hinted.

IS_ISB bool

Whether the instruction is ISB.

IS_LINK bool

Indicates if this is a branch with link.

PC unsigned int

The address of the branch instruction.

TARGET_ISET enum

The instruction set after the branch.

TARGET_PC unsigned int

The address the instruction branches to.

BRA_INDIR

Indirect branches, perhaps to a register. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISET enum

The instruction set of the branch instruction.

IS_COND bool

Indicates if this is a conditional branch.

IS_HINTED bool

Whether the conditional instruction is hinted.

IS_ISB bool

Whether the instruction is ISB.

IS_LINK bool

Indicates if this is a branch with link.

PC unsigned int

The address of the branch instruction.

TARGET_ISET enum

The instruction set after the branch.

TARGET_PC unsigned int

The address the instruction branches to.

BRA_INDIR_CCFAIL

Indirect branches fail, perhaps to a register.

BRA_LOB

Low Overhead branch. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISET enum

The instruction set of the branch instruction.

IS_COND bool

Indicates if this is a conditional branch.

IS_HINTED bool

Whether the conditional instruction is hinted.

IS_ISB bool

Whether the instruction is ISB.

IS_LINK bool

Indicates if this is a branch with link.

PC unsigned int

The address of the branch instruction.

TARGET_ISET enum

The instruction set after the branch.

TARGET_PC unsigned int

The address the instruction branches to.

BRA_RET_CCFAIL

Branch return fail, perhaps to a register.

CACHE_MAINTENANCE_OP

Cache Maintenance Operation. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

DATA unsigned int

Specified MVA or Set/way.

FUNCTION enum

Clean / Invalidate.

SCOPE enum

Affected region.

SIDE enum

Inst / Data.

CCFAIL

Conditional instruction condition check fail. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CCFAIL_UNC

Unconditional instruction condition check fail. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CCPASS

Conditional instruction condition check pass. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CCPASS_UNC

Unconditional instruction condition check pass. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CODE_CACHE_MAINT

Code cache maintenance. Fields:

END_INCL unsigned int

Inclusive end address for VA or PA requests.

NS bool

Non-secure world for PA.

START unsigned int

Start address for VA or PA requests.

TYPE enum

Request type.

COMPILE_BLOCK_END

Last instruction of basic block translated. Fields:

VADDR unsigned int

Address of next instruction after this basic block.

COMPILE_BLOCK_START

First instruction of basic block translated. Fields:

VADDR unsigned int

Address of first instruction in basic block.

COMPILE_INST

ARM instruction compiled. Fields:

DISASS string

Disassembly of the instruction.

ISSET enum

The instruction set of this instruction.

ITSTATE unsigned int

The ITSTATE current for the instruction.

OPCODE unsigned int

The opcode of the instruction.

PC unsigned int

The address of the instruction.

SIZE unsigned int

The size of the instruction in bytes.

CONTEXTIDR

Context ID Register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

NS bool

Secure or nonsecure banked register is accessed.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The new value written.

CONTEXT_SYNC

Called for every context synchronization event. Fields:

ARCH_CSE bool

Architectural Context Synchronization Event.

PC unsigned int

The address of the synchronization instruction.

SYSREG_SYNC bool

System register synchronization event.

CORE_ENDIAN

Core BE8 Big-Endian state changed. Fields:

BE8 bool

Core BE8 Big-Endian state.

CORE_INFO

Static processor attributes. Only triggered by a call to DumpState(). Fields:

ARCH_PROFILE enum

The architecture profile of the core.

CLUSTER_ID unsigned int

The cluster ID of this processor.

CORE_NUM unsigned int

The number of this core in an MP processor.

DETAILED_CORE_FEATURES enum

Detailed core features as a bitfield. Enum values are bitmasks to extract the info from the field.

FPU_VERSION enum

The VFP version implemented by the core.

MEM_ARCH enum

The memory architecture of the core.

MIDR_EL1 unsigned int

MIDR_EL1 value.

NUM_CORES unsigned int

The number of cores in this MP processor.

QUANTUM_SIZE unsigned int

The default quantum size of the core (in ticks).

SECURITY_FEATURES bool

Does the core have security features?.

CORE_LOADS

Processor load accesses. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ACQREL enum

Is this an acquire/release.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DATA unsigned int

The data read or written.

ELEMENT_SIZE unsigned int

Width of each element.

ENDIANNESS enum

Byte ordering of this access.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

LOCK enum

Normal, exclusive or locked access.

MECID unsigned int

Memory Encryption Context ID, if applicable.

MECID2 unsigned int

Memory Encryption Context ID of the second page, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PAGE_SIZE unsigned int

Page size as $\log_2(\text{size in bytes})$.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK, 2=Exclusive Failed, 3=Unsupported.

SIZE unsigned int

Width of the access in bytes. Only required if DATA is not traced.

TRANS bool

Is this a translated access.

VADDR unsigned int

The virtual address of the access.

CORE_REGS

Changes of the core registers R0 to R14. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number, 0 to 14.

OLD_VALUE unsigned int

The old value overwritten.

PHYS_ID enum

The physical register accessed.

VALUE unsigned int

The new value written to the register.

CORE_STALLED_TRANSITION

Detect when a core is prevented from progressing PChannel states. Fields:

CURRENT_STATE enum

Current power mode of core.

REASON string

The reason a transition was stalled, and what must be done to unblock it.

TARGET_STATE enum

Desired power mode of core after transition.

CORE_STORES

Processor store accesses. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ACQREL enum

Is this an acquire/release.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DATA unsigned int

The data read or written.

ELEMENT_SIZE unsigned int

Width of each element.

ENDIANNESS enum

Byte ordering of this access.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

LOCK enum

Normal, exclusive or locked access.

MECID unsigned int

Memory Encryption Context ID, if applicable.

MECID2 unsigned int

Memory Encryption Context ID of the second page, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PAGE_SIZE unsigned int

Page size as $\log_2(\text{size in bytes})$.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK, 2=Exclusive Failed, 3=Unsupported.

SIZE unsigned int

Width of the access in bytes. Only required if DATA is not traced.

TRANS bool

Is this a translated access.

VADDR unsigned int

The virtual address of the access.

CRYPTO_SPEC

Every crypto instruction speculatively executed.

DATA_CACHE_SET_ALLOC_TAGS

Set allocation tags from DC GVA instruction. Fields:

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

NSDESC unsigned int

The security state of the access.

PADDR unsigned int

The physical address of the access.

PAS enum

The physical address space of the page.

RESPONSE enum

0=Aborted, 1=OK.

SIZE unsigned int

The size of memory in bytes.

TAG unsigned int

The value of the tag being set.

TAGGING_TYPE enum

Type of Memory tagging (Physical or Virtual).

TAG_VADDR unsigned int

The virtual address of the Virtual Tag if Virtual tagging is used; otherwise 0.

VADDR unsigned int

Virtual address used for setting allocation tags.

DATA_CACHE_ZERO

Zero and invalidate cache line from DC ZVA instruction. Fields:

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

NSDESC unsigned int

The security state of the access.

PADDR unsigned int

The physical address of the access.

PADDR_TAG unsigned int

If tag checked, the physical address tag for the access.

PAS enum

The physical address space of the page.

RESPONSE enum

0=Aborted, 1=OK.

SIZE unsigned int

The size of zero'd memory in bytes.

TAG_CHECKED bool

Is this access tag checked.

VADDR unsigned int

Virtual address of the zero'd cache line.

DEBUG_EVENT

Hardware debug support event. Fields:

EVENT enum

Description of event.

VALUE unsigned int

data value.

DMI_ALLOCATE

DMI allocated. Fields:

ACCESS_DIRECTION enum

READ / WRITE.

PADDR unsigned int

The physical address of the DMI.

PTR unsigned int

DMI range pointer.

SIDE enum

DSIDE / ISIDE.

SIZE unsigned int

The size of DMI.

VADDR unsigned int

The virtual address of the DMI..

DMI_HIT

DMI hit, access used DMI (only fire for small DMI regions i.e. less than a full page). Fields:

ACCESS_DIRECTION enum

READ / WRITE.

PADDR unsigned int

The physical address of the DMI.

SIDE enum

DSIDE / ISIDE.

SIZE unsigned int

The size of the DMI.

VADDR unsigned int

The virtual address of the DMI..

DMI_REVOKE

DMI revoked. Fields:

ACCESS_DIRECTION enum

READ / WRITE.

PADDR unsigned int

The physical address of the DMI.

SIDE enum

DSIDE / ISIDE.

SIZE unsigned int

The size of DMI.

VADDR unsigned int

The virtual address of the DMI..

DTCM_ACCESS

Trace TCM accesses.

DTCM_ACCESS_RAZ_WI

Trace TCM accesses. Fields:

ADDR unsigned intAddress of the **RAZ/WI** access.**DWT_MATCH**

DWT comparator matches. Fields:

NUM unsigned int

DWT comparator number.

TYPE enum

DWT comparator configuration.

END_COMPILE

Compilation end. Fields:

END_OF_PAGE_COUNT unsigned int

Number of basic blocks exited due to page boundary since START_COMPILE.

FETCHFAIL_COUNT unsigned int

Number of basic blocks exited due to ifetch failure START_COMPILE.

INST_COUNT unsigned int

Number of instructions compiled since START_COMPILE.

NONSEQ_COUNT unsigned int

Number of basic blocks exited due to non-sequential instructions since START_COMPILE.

PAGE_STRADDLE_COUNT unsigned int

Number of basic blocks exited due to unaligned instructions crossing page since START_COMPILE.

EXCEPTION

Exceptions that are taken. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ISET enum

The instruction set of the processor when the exception occurred.

LR unsigned int

The value assigned to the link register.

NS enum

The core's non-secure bit.

PC unsigned int

The location where the exception occurred.

TARGET_ISET enum

The instruction set of the exception handler code.

TARGET_PC unsigned int

The address the exception branches to.

VECTOR enum

The exception vector.

EXCEPTION_END

Every exception completed. Fields:

NS enum

The core's non-secure bit.

EXCEPTION_ENTRY

Event marking the entry of an exception. All pushes of registers to the stack will follow this event. Fields:

PC unsigned int

The location where the exception occurred.

VECTOR enum

The exception that occurred.

EXCEPTION_RAISE

Every exception raised.

EXCEPTION_RETURN

Marks the end of an exception, but on an M core does not indicate a program flow change (branch). In most situations it will be followed by an indirect branch. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISSET enum

The instruction set of the branch instruction.

PC unsigned int

The address of the branch instruction.

EXCEPTION_RETURN_PREBRANCH

Exception return event that must occur before PC and processor state have updated.

EXCEPTION_START

Every exception started. Fields:

NS enum

The core's non-secure bit.

EXCEPTION_VECTOR_FETCH

I-side vector fetch for M-class exception. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

DATA unsigned int

Result of fetch if OK.

RESPONSE enum

0=Aborted, 1=OK.

VADDR unsigned int

Where the vector is fetched from (including VTOR).

FP_STATE

Floating point state. Fields:

FPSR unsigned int

FPSR register value.

MASK unsigned int

The mask indicating updated cumulative bits.

FREQ_CHANGED

Core frequency changed. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The instruction count of this core.

NEW_FREQ signed int

The new frequency of this core (expressed in Hz).

OLD_FREQ signed int

The old frequency of this core (expressed in Hz).

HLT

HLT instruction occurred. Fields:

IMM16 unsigned int

The 16-bit immediate value encoded in HLT instruction.

INFO_EXCEPTION_REASON

Provide information of real cause of exception. Fields:

FaultCause enum

Which bit of which V7M/V8M-mainline FSR/HFSR would have been set.

PC unsigned int

The location where the exception occurred.

PHASE enum

What we are doing with the exception.

REASONS enum

Why we are doing it.

SecurityState enum

Privacy State of exception.

VECTOR enum

The exception that occurred.

INFO_FP_CONTEXT

Inform when ExecuteFPCheck writes CONTROL.FPCA and FPSCR etc. Fields:

WHAT enum

What we are doing.

INFO_STACKING

Provide information of real cause of exception. Fields:

PHASE enum

Are we starting or ending a sequence of stack writes.

REASONS enum

Why we are doing it.

INST

Every instruction executed. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DEBUG_STATE bool

The instruction is executed in debug state.

DISASS string

Disassembly of instruction.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISSET enum

The current instruction set.

ITSTATE unsigned int

The current ITSTATE.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

MODE enum

The mode the core is in.

NS unsigned int

The core's non-secure bit.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

OPCODE unsigned int

The opcode of the instruction.

PADDR unsigned int

The physical address of the instruction.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the instruction.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the page.

PC unsigned int

The address of the instruction.

SECURITY_STATE enum

The core's security state.

SIZE unsigned int

The size of the instruction in bytes.

INST_END

Every instruction completed.

INST_START

Every instruction started. Fields:

DEBUG_STATE bool

The instruction is executed in debug state.

ISSET enum

The current instruction set.

MODE enum

The mode the core is in.

NS enum

The current Secure State.

PC unsigned int

The address of the conditional instruction.

SECURITY_STATE enum

The current Security State.

INST_STRADDLE

Instruction straddles boundary. Fields:

DEBUG_STATE bool

The instruction is executed in debug state.

ISSET enum

The current instruction set.

MODE enum

The mode the core is in.

NS enum

The current Secure State.

PC unsigned int

The address of the conditional instruction.

SECURITY_STATE enum

The current Security State.

ISB

ISB instruction occurred. Fields:

PC unsigned int

The address of the ISB instruction.

ITCM_ACCESS

Trace TCM accesses.

ITCM_ACCESS_RAZ_WI

Trace TCM accesses. Fields:

ADDR unsigned int

Address of the **RAZ/WI** access.

ITM

Instrumentation Trace Macrocell. Fields:

ITM_PACKET_TYPE enum

ITM and DWT packets type.

PACKET_HEADER unsigned int

ITM Packet Header.

PACKET_PAYLOAD unsigned int

ITM Packet Payload.

LOAD_MULTIPLE_REGS_LIST

List of destination registers (not source) of a multiple registers load instruction. Fields:

REG string

Register to add into the list.

LOCAL_MONITOR

Local monitor activity. Fields:

PADDR unsigned int

Local Monitor Address.

State enum

State of the monitor (Open/Exclusive).

LOCKUP_CYCLE

This event is triggered if the core enters or stays in the lock-up state. Fields:

CAUSE enum

What causes the lockup?.

LOCKUP_ENTRY

This event is triggered if the core enters the lock-up state. Fields:

PC unsigned int

The PC when the lockup state is entered.

LOCKUP_EXIT

This event is triggered if the core leaves the lock-up state.

LO_BRANCH_INFO

Low Overhead Branch cache Info. Fields:

BF unsigned int

Value originates from a BF instructi.

CAUSE enum

What caused the update.

END_ADDR unsigned int

Address of the instruction after the.

JUMP_ADDR unsigned int

Jump address.

LF unsigned int

Link/forever.

T16IND unsigned int

T16 indirect branch.

VALID unsigned int

Entry is valid.

LO_BRANCH_INSTR

Low Overhead Branch Instruction. Fields:

EFFECT enum

Did it set LO_BRANCH_INFO and/or branch.

INSTR enum

What sort of instruction (BF/..BFCSEL/LE/WLS/...).

MODE_CHANGE

Mode change. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

MODE **enum**

The new mode.

OLD_MODE **enum**

The old mode.

MPU_TRANS

Address translation information. Fields:

CORE_NUM **unsigned int**

Core number in a multi processor.

EXEC_PERM **enum**

Execution Permission.

IDAU_REG_NUM **enum**

IDAU region-number or code.

INNERCACHE_RA **bool**

Is the inner cache allocate on read.

INNERCACHE_TYPE **enum**

Inner Caching scheme (NC/MB/WA).

INNERCACHE_WA **bool**

Is the inner cache allocate on write.

MEMTYPE **enum**

Memory type.

NSDESC **enum**

Security-attributes of the transaction.

OUTERCACHE_RA **bool**

Is the outer cache allocate on read.

OUTERCACHE_TYPE **enum**

Outer Caching scheme (NC/MB/WA).

OUTERCACHE_WA **bool**

Is the outer cache allocate on write.

PADDR **unsigned int**

Address of the access.

PAGESIZE **unsigned int**

Page size as $\log_2(\text{size})$.

PAS **enum**

PAS accessed is secure or nonsecure memory.

READ_PERM **enum**

Read Permission.

REG_NUM unsigned int

MPU register based region-number 0-15 (else 0xFF=none, 0xBB/0xDD=background/default).

SAU_REG_NUM enum

SAU region-number or code.

SAU_RGN_TYPE enum

Privacy Region Type.

SH enum

Shareability.

SIDE enum

Inst / Data.

WRITE_PERM enum

Write Permission.

PERIODIC

Called for every quantum. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The instruction count of this CPU.

PC unsigned int

The address of the next instruction to be executed on this CPU.

PMU_COUNTER_OVERFLOW

PMU counter overflow. Fields:

COUNTER_GROUP enum

Counter group.

COUNTER_RANGE enum

Counter range.

EVENT_ID unsigned int

Event Identifier, field is valid only for Event counter group.

INDEX unsigned int

Counter index (as selected by PMSELR), field is valid only for Event/Cycle counter groups.

INTERRUPT bool

Is interrupt enabled on overflow for this counter.

PMU_M_EVENT_AUT_RETIRED

PMU Event AUT_RETIRED.

PMU_M_EVENT_BF_CANCEL

PMU Event BF_CANCEL.

PMU_M_EVENT_BF_RETIRED

PMU Event BF_RETIRED.

PMU_M_EVENT_DWT_CMPMATCH0

PMU Event DWT_CMPMATCH0.

PMU_M_EVENT_DWT_CMPMATCH1

PMU Event DWT_CMPMATCH1.

PMU_M_EVENT_DWT_CMPMATCH2

PMU Event DWT_CMPMATCH2.

PMU_M_EVENT_DWT_CMPMATCH3

PMU Event DWT_CMPMATCH3.

PMU_M_EVENT_DWT_CMPMATCH4

PMU Event DWT_CMPMATCH4.

PMU_M_EVENT_DWT_CMPMATCH5

PMU Event DWT_CMPMATCH5.

PMU_M_EVENT_DWT_CMPMATCH6

PMU Event DWT_CMPMATCH6.

PMU_M_EVENT_DWT_CMPMATCH7

PMU Event DWT_CMPMATCH7.

PMU_M_EVENT_INST_SPEC

PMU Event INST_SPEC.

PMU_M_EVENT_LE_CANCEL

PMU Event LE_CANCEL.

PMU_M_EVENT_LE_RETIRED

PMU Event LE_RETIRED.

PMU_M_EVENT_MVE_FP_HP_RETIRED

PMU Event MVE_FP_HP_RETIRED.

PMU_M_EVENT_MVE_FP_HP_SPEC

PMU Event MVE_FP_HP_SPEC.

PMU_M_EVENT_MVE_FP_MAC_RETIRED

PMU Event MVE_FP_MAC_RETIRED.

PMU_M_EVENT_MVE_FP_MAC_SPEC

PMU Event MVE_FP_MAC_SPEC.

PMU_M_EVENT_MVE_FP_RETIRED

PMU Event MVE_FP_RETIRED.

PMU_M_EVENT_MVE_FP_SPEC

PMU Event MVE_FP_SPEC.

PMU_M_EVENT_MVE_FP_SP_RETIRE

PMU Event MVE_FP_SP_RETIRE.

PMU_M_EVENT_MVE_FP_SP_SPEC

PMU Event MVE_FP_SP_SPEC.

PMU_M_EVENT_MVE_INST_RETIRE

PMU Event MVE_INST_RETIRE.

PMU_M_EVENT_MVE_INST_SPEC

PMU Event MVE_INST_SPEC.

PMU_M_EVENT_MVE_INT_MAC_RETIRE

PMU Event MVE_INT_MAC_RETIRE.

PMU_M_EVENT_MVE_INT_MAC_SPEC

PMU Event MVE_INT_MAC_SPEC.

PMU_M_EVENT_MVE_INT_RETIRE

PMU Event MVE_INT_RETIRE.

PMU_M_EVENT_MVE_INT_SPEC

PMU Event MVE_INT_SPEC.

PMU_M_EVENT_MVE_LDST_MULTI_RETIRE

PMU Event MVE_LDST_MULTI_RETIRE.

PMU_M_EVENT_MVE_LDST_MULTI_SPEC

PMU Event MVE_LDST_MULTI_SPEC.

PMU_M_EVENT_MVE_LDST_RETIRE

PMU Event MVE_LDST_RETIRE.

PMU_M_EVENT_MVE_LDST_SPEC

PMU Event MVE_LDST_SPEC.

PMU_M_EVENT_MVE_LD_MULTI_RETIRE

PMU Event MVE_LD_MULTI_RETIRE.

PMU_M_EVENT_MVE_LD_MULTI_SPEC

PMU Event MVE_LD_MULTI_SPEC.

PMU_M_EVENT_MVE_LD_RETIRE

PMU Event MVE_LD_RETIRE.

PMU_M_EVENT_MVE_LD_SPEC

PMU Event MVE_LD_SPEC.

PMU_M_EVENT_MVE_PRED

PMU Event MVE_PRED.

PMU_M_EVENT_MVE_ST_MULTI_RETIRE

PMU Event MVE_ST_MULTI_RETIRE.

PMU_M_EVENT_MVE_ST_MULTI_SPEC

PMU Event MVE_ST_MULTI_SPEC.

PMU_M_EVENT_MVE_ST_RETIRE

PMU Event MVE_ST_RETIRE.

PMU_M_EVENT_MVE_ST_SPEC

PMU Event MVE_ST_SPEC.

PMU_M_EVENT_MVE_VREDUCE_FP_RETIRE

PMU Event MVE_VREDUCE_FP_RETIRE.

PMU_M_EVENT_MVE_VREDUCE_FP_SPEC

PMU Event MVE_VREDUCE_FP_SPEC.

PMU_M_EVENT_MVE_VREDUCE_INT_RETIRE

PMU Event MVE_VREDUCE_INT_RETIRE.

PMU_M_EVENT_MVE_VREDUCE_INT_SPEC

PMU Event MVE_VREDUCE_INT_SPEC.

PMU_M_EVENT_MVE_VREDUCE_RETIRE

PMU Event MVE_VREDUCE_RETIRE.

PMU_M_EVENT_MVE_VREDUCE_SPEC

PMU Event MVE_VREDUCE_SPEC.

PMU_M_EVENT_PAC_RETIRE

PMU Event PAC_RETIRE.

PMU_M_EVENT_SE_CALL_NS

PMU Event SE_CALL_NS.

PMU_M_EVENT_SE_CALL_S

PMU Event SE_CALL_S.

POTINDEX_UPDATE

Value of POT index for current pc changed. Fields:

NEW_POTINDEX unsigned int

New POTIndex.

OLD_POTINDEX unsigned int

Old POTIndex.

PPB_READ_ACCESS

Trace reads from the core to memory mapped registers in the PPB (private peripheral bus) address range. Fields:

ACCESS_SIZE unsigned int

Logarithm of the access size: 0=byte, 1=halfword, 2=word, 3=doubleword, ...

BURST_LEN unsigned int

The number of data transfers (beats) in this burst.

DATA unsigned int

The data transferred.

INNER_ATTRIBUTES unsigned int

The inner attributes: bit0=write allocate, bit1=read allocate, bit2=cachable, bit3=bufferable, bit4=shareable.

INST bool

Is an instruction fetch.

LOCK enum

Normal / Locked / Exclusive.

MANAGER_ID unsigned int

The AXI manager ID.

NS bool

Is a nonsecure access.

OUTER_ATTRIBUTES unsigned int

The outer attributes: bit0=write allocate, bit1=read allocate, bit2=cachable, bit3=bufferable, bit4=shareable.

PADDR unsigned int

Physical address of access.

PRIV bool

Is privileged access.

RESPONSE enum

Whether the transaction was successful, or an error occurred.

USER_FLAGS unsigned int

Core specific additional signals.

PPB_WRITE_ACCESS

Trace writes from the core to memory mapped registers in the PPB (private peripheral bus) address range. Fields:

ACCESS_SIZE unsigned int

Logarithm of the access size: 0=byte, 1=halfword, 2=word, 3=doubleword, ...

BURST_LEN unsigned int

The number of data transfers (beats) in this burst.

DATA unsigned int

The data transferred.

INNER_ATTRIBUTES unsigned int

The inner attributes: bit0=write allocate, bit1=read allocate, bit2=cachable, bit3=bufferable, bit4=shareable.

INST bool

Is an instruction fetch.

LOCK enum

Normal / Locked / Exclusive.

MANAGER_ID unsigned int

The AXI manager ID.

NS bool

Is a nonsecure access.

OUTER_ATTRIBUTES unsigned int

The outer attributes: bit0=write allocate, bit1=read allocate, bit2=cachable, bit3=bufferable, bit4=shareable.

PADDR unsigned int

Physical address of access.

PRIV bool

Is privileged access.

RESPONSE enum

Whether the transaction was successful, or an error occurred.

USER_FLAGS unsigned int

Core specific additional signals.

PRELOAD_DATA

Data preload from PLD instruction. Fields:

VADDR unsigned int

Virtual address of the data that should be preloaded.

PRELOAD_INST

Instruction preload from PLI instruction. Fields:

VADDR unsigned int

Virtual address of the instruction that should be preloaded.

PRE_CORE_LOAD

Trace just before a core load. Fields:

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

PRE_CORE_STORE

Trace just before a core store. Fields:

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

PSTATE_PACTIVE_OPMODE_TRANSITION

Trace pactive and Opmode mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Operating mode after the transition.

OLD_STATE enum

Operating mode before the transition.

PSTATE_PACTIVE_OPMODE_TRANSITION_DENIED

Trace the denied pactive and Opmode mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Operating mode after the transition.

OLD_STATE enum

Operating mode before the transition.

REASON string

Reason for the transition being denied.

PSTATE_PACTIVE_POWER_TRANSITION

Trace pactive and power mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Power mode after the transition.

OLD_STATE enum

Power mode before the transition.

PSTATE_PACTIVE_POWER_TRANSITION_DENIED

Trace the denied pactive and power mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Power mode after the transition.

OLD_STATE enum

Power mode before the transition.

REASON string

Reason for the transition being denied.

PSTATE_PACTIVE_THREAD_OPMODE_TRANSITION

Trace pactive and Opmode mode transitions for threads on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Operating mode after the transition.

OLD_STATE enum

Operating mode before the transition.

QUANTUM_START

Called at the start of every quantum.

RUN_STATE

Run state transition. Fields:

INST_COUNT unsigned int

Ticks count at point of transition.

NEW enum

New run state.

OLD enum

Old run state.

SEMIHOSTING_CALL

Call of a semihost function occurred. Fields:

PC unsigned int

The program counter after the semihosting call.

SEMIHOSTING_PRECALL

About to call semihost. Fields:

EL2 bool

Use EL2 translation regime.

NS bool

Is Non-Secure.

REG_WIDTH unsigned int

The current register width in bytes.

SIGNAL

External signal state change. Fields:

SIGNAL enum

Signal that changed.

STATE bool

Signal asserted state.

START_COMPILE

Compilation started. Fields:

VADDR unsigned int

Instruction where compilation begins.

STG_MULTIPLE_MICRO_OPS_ADDRESS_RANGE

Address range for MTE stores with multiple uops. Fields:

ADDR_END unsigned int

Final address.

ADDR_START unsigned int

Starting address.

STORE_MULTIPLE_REGS_ADDRESS_RANGE

Address range for stores with multiple registers. Fields:

ADDR_END unsigned int

Final address.

ADDR_START unsigned int

Starting address.

SUBOPTIMAL_LDST_RETIRED

suboptimal load/store. Fields:

ACCESS_SIZE unsigned int

Log2 of the access width in bytes used for alignment checking (1: half-word, 2: word ...).

OPTIMAL_ALIGN_SELECTOR unsigned int

Selector to decide the optimal alignment size (1: access width, 2: 4-bytes, 3: 8-bytes, ..., 12: 4-kbytes).

VADDR unsigned int

The virtual address of the access.

SYNC

Called for every synchronization, that is every quantum break. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The instruction count of this CPU.

LOCAL_QUANTUM unsigned int

The local quantum of this CPU.

LOCAL_TIME unsigned int

The local time of this CPU.

SYNC_REASON enum

Reason for sync.

SYSCALL

System call instruction executed. Fields:

IMM unsigned int

Immediate value of the system call instruction.

TYPE enum

System call type.

VADDR unsigned int

Instruction that caused the system call.

SYSREG_READ

Trace reads from system registers. Fields:

REG_NAME enum

Register number.

VALUE unsigned int

The value read.

SYSREG_UPDATE128

Triggers when the system updates a register. Fields:

REG enum

Register number.

UNKNOWN unsigned int

Bits of the register which became unknown.

VALUE unsigned int

Value written to the register.

SYSREG_UPDATE32

Triggers when the system updates a register. Fields:

REG enum

Register number.

UNKNOWN unsigned int

Bits of the register which became unknown.

VALUE unsigned int

Value written to the register.

SYSREG_UPDATE64

Triggers when the system updates a register. Fields:

REG enum

Register number.

UNKNOWN unsigned int

Bits of the register which became unknown.

VALUE unsigned int

Value written to the register.

SYSREG_WRITE

Trace write to system registers. Fields:

REG_ACCESS enum

Register access status.

REG_NAME enum

Register number.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

Value written to the register.

UNALIGNED_LDST_RETIRED

Processor unaligned load/store. Fields:

ACCESS_SIZE unsigned int

Log2 of the access width in bytes used for alignment checking (1=>half-word, 2->word ...).

NUMBER_OF_BEATS unsigned int

Number of accesses (beats) in this burst (Total size in bytes is ACCESS_SIZE * NUMBER_OF_BEATS).

VADDR unsigned int

The virtual address of the access.

VFP_D_REGS

VFP/NEON D 64 bit register write. Fields:

ALIASING enum

Type of register aliasing used.

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a write to the corresponding bit in the register. For example, 0x1 would indicate a write to VALUE[0].

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

VFP_D_REGS_READ

VFP/NEON D 64 bit register read. Fields:

ID unsigned int

The register number.

VALUE unsigned int

The current value of the register.

VFP_Q_REGS

VFP/NEON Q 128 bit register write. Fields:

ALIASING enum

Type of register aliasing used.

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a write to the corresponding byte in the register. For example, 0x1 would indicate a write to VALUE[7:0].

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

VFP_Q_REGS_READ

VFP/NEON Q 128 bit register read. Fields:

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a read to the corresponding byte in the register. For example, 0x1 would indicate a read to VALUE[7:0].

VALUE unsigned int

The current value of the register.

VFP_SYS_REGS

Writes to the VFP/NEON units system registers. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ID enum

Which VFP system register is written.

MASK unsigned int

The mask indicating updated cumulative bits.

OLD_VALUE unsigned int

The register's old value overwritten.

VALUE unsigned int

The new value written to the VFP system register.

VFP_S_REGS

VFP/NEON S 32 bit register write. Fields:

ALIASING enum

Type of register aliasing used.

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a write to the corresponding bit in the register. For example, 0x1 would indicate a write to VALUE[0].

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

VFP_S_REGS_READ

VFP/NEON S 32 bit register read. Fields:

ID unsigned int

The register number.

VALUE unsigned int

The current value of the register.

WAYPOINT

Signals end of basic block execution. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ISSET enum

Origin instruction set.

IS_COND bool

Indicates if this is a conditional waypoint.

PC unsigned int

Origin address (or 0 if unavailable).

TAKEN bool

Indicates if this waypoint was taken.

TARGET unsigned int

Destination address.

TARGET_ISET enum

Destination instruction set.

WFE_END

WFE ended. Fields:

INST_COUNT unsigned int

Ticks count when leaving WFE.

WFE_EVENT_REGISTER

WFE event register status: set/clear, reason. Fields:

INST_COUNT unsigned int

Ticks count.

REASON enum

Reason for set/clear. Only REASON==0 (Cleared by WFE) clears the bit, all other reasons set it.

WFE_IGNORED

WFE ignored. Fields:

EVENT bool

This WFE was ignored because the event register was set.

INST_COUNT unsigned int

Ticks count when ignoring WFE.

TRAPPED bool

This WFE was trapped.

WFE_START

WFE entered. Fields:

INST_COUNT unsigned int

Ticks count when entering WFE.

WFI_END

WFI ended. Fields:

INST_COUNT unsigned int

Ticks count when leaving WFI.

WFI_IGNORED

WFI ignored. Fields:

DISABLED bool

This WFI was ignored because WFI is disabled.

INST_COUNT unsigned int

Ticks count when ignoring WFI.

REASON enum

specifies reason why WFI trace was ignored.

TRAPPED bool

This WFI was trapped.

WFI_START

WFI entered. Fields:

INST_COUNT unsigned int

Ticks count when entering WFI.

WFI_WAKEUP

WFI wakeup. Fields:

INST_COUNT unsigned int

Ticks count when WFI wakeup occurred.

REASON enum

Reason for wakeup.

xPSR

Changes to the xPSR register. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

OLD_VALUE unsigned int

The old xPSR value.

UNKNOWN unsigned int

Bits within the register that have unknown value.

VALUE unsigned int

The new xPSR value.

2.47 ARM_Cortex-M7

This section describes the trace sources.

ASYNC_MEMORY_FAULT

Context ID Register write. Fields:

FAULT unsigned int

Fault status present in ESR.ISS encoding for Serror in ARCH64 and present in DFSR encoding for external abort format in ARCH32.

PADDR unsigned int

Physical Address (or 0 if unavailable).

ATOMIC_END_ACCESS

Bus trace access for atomics. Fields:

ACCESS_FAIL bool

Memory access failed.

ADDR unsigned int

The virtual address of the access.

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

COMPARE_VALUE unsigned int

Compare value for CAS.

LOAD_VALUE unsigned int

Loaded values.

NSDESC unsigned int

The security state of the access.

OPERAND_VALUE unsigned int

Operation's operand.

OPERATION enum

Operation type.

PADDR unsigned int

The physical address of the access.

PRIV bool

Is this a privileged access?.

ATOMIC_SLAVE_ACCESS

Information on the atomic operation at slave side. Fields:

ACCESS_RESULT unsigned int

Result of atomic access.

COMPARE_VALUE unsigned int

Value to be compared with memory by CAS.

LOAD_VALUE unsigned int

Value to be loaded from memory as a result of atomic operation.

MANAGER unsigned int

ID of manager which initiated the transaction.

NS bool

Transaction is Non-Secure.

OPERAND_VALUE unsigned int

Incoming operand value provided by core.

OPERATION enum

Type of atomic operation.

PADDR unsigned int

Full physical address of TCM.

STORE_VALUE unsigned int

Value to be stored to memory as a result of atomic operation.

ATOMIC_START_ACCESS

Bus trace access for atomics. Fields:

ADDR unsigned int

The virtual address of the access.

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

COMPARE_VALUE unsigned int

Compare value for CAS.

NSDESC unsigned int

The security state of the access.

OPERAND_VALUE unsigned int

Operation's operand.

OPERATION enum

Operation type.

PADDR unsigned int

The physical address of the access.

PRIV bool

Is this a privileged access?.

ArchMsg.Error.exit_code

DISPLAY exit code: %{EXITCODE} component kind: %{KIND} reason: %{REASON}. Fields:

EXITCODE signed int

Exit code to be returned.

KIND string

Component kind that invoked the exit code trace.

REASON string

Reason why the exit code trace was invoked (optional).

ArchMsg.Warning.Unpredictably Indexed PM Event Register

TAGS **UNPREDICTABLE** DISPLAY Event %{IsDirect:(Selector value|Register)} %{SEL} is >= %{N}.
Fields:

IsDirect unsigned int

Direct Access rather than indirect.

N unsigned int

Number of Accessible Registers.

SEL unsigned int

Selector.

ArchMsg.Warning.cache_contents_unknown

A cache has been activated without being invalidated and may contain **UNKNOWN** content.
DISPLAY %{SIDE:(D|I)}-Cache was enabled in %{REGIME:(SECURE|NON_SECURE|HYP)} regime but was not invalidated since power-on: cache lines could contain **UNKNOWN** content
%{SIDE:} (Note: D-side caches must invalidate SECURE lines even if the cache is only used by NON_SECURE code in order to ensure that unexpected dirty lines tagged SECURE are not naturally evicted). Fields:

REGIME unsigned int

Current translation regime.

SIDE unsigned int

0 d-side cache or unified cache, 1 i-side.

ArchMsg.Warning.dap_csw_bad_size

A write to CM3DAP CSW has an invalid size field. DISPLAY Write %{DATA} to CM3DAP CSW has invalid size field. Fields:

DATA unsigned int

bits[2:0] are size.

ArchMsg.Warning.dcimvac_matches_watchpoint

TAGS **UNPREDICTABLE** DISPLAY Watchpoints matching an AArch32 DCIMVA are implementation defined.

ArchMsg.Warning.decode_fieldmismatch

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with different fields %{FIELD1} and %{FIELD2}. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_fracbitsnegative

TAGS **UNPREDICTABLE** DISPLAY VCVT Instruction is unpredictable with a negative number of fraction bits.

ArchMsg.Warning.decode_initblock

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable in an IT block.

ArchMsg.Warning.decode_initblocknotlast

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable in an IT block when not the last.

ArchMsg.Warning.decode_invalid_condition

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable when the condition is not AL.

ArchMsg.Warning.decode_invalidfieldcombination

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with the values in %{FIELD1} and %{FIELD2}. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_invalidvalue

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with the value in %{FIELD}. Fields:

FIELD enum

Field name.

ArchMsg.Warning.decode_mem_hint_unallocated

TAGS **UNPREDICTABLE** DISPLAY There is no memory hint allocated to this bit pattern.

ArchMsg.Warning.decode_registermatch

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with %{FIELD1} and %{FIELD2} as the same register. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_registermismatch

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with %{FIELD1} and %{FIELD2} as different registers. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_registeroutofrange

TAGS UNPREDICTABLE DISPLAY Floating-point instruction is unpredictable indexing beyond the end of the register bank.

ArchMsg.Warning.decode_sbzsbo

TAGS UNPREDICTABLE DISPLAY Reserved bits in the instruction are not canonical.

ArchMsg.Warning.decode_transactiontoobig

TAGS UNPREDICTABLE DISPLAY Attempt to perform a memory transaction of more than 128 bytes.

ArchMsg.Warning.decode_unpred_other

TAGS UNPREDICTABLE DISPLAY Instruction is unpredictable.

ArchMsg.Warning.decode_unpreduseofpc

TAGS UNPREDICTABLE DISPLAY Instruction is unpredictable when %{FIELD} is PC. Fields:

FIELD enum

Field name.

ArchMsg.Warning.decode_unpreduseofr13

TAGS UNPREDICTABLE DISPLAY Instruction is unpredictable when %{FIELD} is R13. Fields:

FIELD enum

Field name.

ArchMsg.Warning.decode_unpreduseofr13inreglist

TAGS UNPREDICTABLE DISPLAY Instruction is unpredictable with R13 in register-list.

ArchMsg.Warning.decode_useofd16orhigher

TAGS UNPREDICTABLE DISPLAY Attempt to use register D16 or higher where it is not allowed.

ArchMsg.Warning.decode_writebackandbaseinlist

TAGS UNPREDICTABLE DISPLAY Instruction is unpredictable with write back when the base register is in the transfer list.

ArchMsg.Warning.decode_zeroregistersinlist

TAGS UNPREDICTABLE DISPLAY Instruction is unpredictable when no registers are to be transferred.

ArchMsg.Warning.recursive_branch

An instruction is performing a branch that targets the same instruction. If this was intended then a WFI might be more effective DISPLAY recursive branch detected at PC=%{PC}.

PRIMARY KEY PC. Fields:

PC unsigned int

Address of instruction causing the branch.

ArchMsg.Warning.recursive_exception

An instruction has generated an exception that targets the same instruction. DISPLAY recursive %{TYPE} exception detected at PC=%{PC} (CPSR=%{CPSR}). PRIMARY KEY PC. Fields:

CPSR unsigned int

Processor state.

PC unsigned int

Address of instruction causing the exception.

TYPE enum

Type of exception.

ArchMsg.Warning.reentrant_vector_catch

TAGS **UNPREDICTABLE** DISPLAY Vector catch debug event on Prefetch or Data abort vector %{ADDR}. Fields:

ADDR unsigned int

vector address.

ArchMsg.Warning.reserved_it_state

TAGS **UNPREDICTABLE** DISPLAY Execution with reserved IT state %{ITSTATE}. Fields:

ITSTATE unsigned int

reserved IT state.

ArchMsg.Warning.secure_vector_fetch_from_nonsecure

DISPLAY Secure vector table is being fetched from Non-secure memory.

ArchMsg.Warning.sg_in_it_block

Arm recommends that software does not place SG (security gate instruction) inside an IT instruction block.

ArchMsg.Warning.unpred_FAR_on_LRCPC3_non_atomic_mem_access

TAGS **UNPREDICTABLE** Unpredictable FAR/FSTATUS on LRCPC3 pair instructions resulting in two separate memory accessesDISPLAY Unpredictable FAR on LRCPC3 non atomic memory access at %{ADDR1} with FSTATUS %{FSTATUS1} or %{ADDR2} with FSTATUS %{FSTATUS2}. Fields:

ADDR1 unsigned int

Possible FAR address 1.

ADDR2 unsigned int

Possible FAR address 2.

FSTATUS1 unsigned int

Possible fault status value 1.

FSTATUS2 unsigned int

Possible fault status value 2.

ArchMsg.Warning.unpred_async_tag_check_fail_on_sve_speculative_load_do_update_ffr

Unpredictable set to FALSE of the FFR element associated to SVE speculative load with an async tag check fail TAGS **UNPREDICTABLE**.

ArchMsg.Warning.unpredictable_a32_breakpoint

TAGS **UNPREDICTABLE** DISPLAY Unpredictable breakpoint on a A32 instruction at %{ADDR1} Offset %{OFFSET} BAS %{BAS} for a %{IS_ADDRESS_MISMATCH_BREAKPOINT:(Mismatch|Match)} Breakpoint number %{N:d} Breakpoint address %{ADDR2}. Fields:

ADDR1 unsigned int

Address.

ADDR2 unsigned int

Breakpoint address.

BAS unsigned int

BAS field.

IS_ADDRESS_MISMATCH_BREAKPOINT bool

Is a Mismatch breakpoint, not a Match breakpoint.

N unsigned int

Breakpoint number.

OFFSET unsigned int

Always has a value of zero.

ArchMsg.Warning.unpredictable_banked_register_access

TAGS **UNPREDICTABLE** DISPLAY Access to banked register (R=%{R:d} SYSm=%{SYSM:2x}) is unpredictable. Fields:

R unsigned int

R bit.

SYSM unsigned int

SYSm.

ArchMsg.Warning.unpredictable_ccfail_undef

TAGS **UNPREDICTABLE** DISPLAY An instruction %{OPCODE} at VA=%{VADDR} failed its condition codes check but would otherwise have caused a non-data-dependent trap or undefined exception. Behaviour is implementation-defined choice of exception or nop. Fields:

OPCODE unsigned int

opcode of the instruction.

VADDR unsigned int

Virtual address of the instruction being executed.

ArchMsg.Warning.unpredictable_ici_bits

TAGS **UNPREDICTABLE** DISPLAY ICI bits do not apply or are architecturally unpredictable for instruction.

ArchMsg.Warning.unpredictable_t32_breakpoint

TAGS **UNPREDICTABLE** DISPLAY Unpredictable breakpoint on a T32 instruction at %{ADDR1} Offset %{OFFSET} BAS %{BAS} for a %{IS_ADDRESS_MISMATCH_BREAKPOINT:(Mismatch|Match)} Breakpoint number %{N:d} Breakpoint address %{ADDR2}. Fields:

ADDR1 unsigned int

Address.

ADDR2 unsigned int

Breakpoint address.

BAS unsigned int

BAS field.

IS_ADDRESS_MISMATCH_BREAKPOINT bool

Is a Mismatch breakpoint, not a Match breakpoint.

N unsigned int

Breakpoint number.

OFFSET unsigned int

Indicates which halfword of a 32-bit Thumb instruction the breakpoint is set on. 0 for the first halfword or if the instruction is 16 bits wide, or 1 for the second halfword.

ArchMsg.Warning.unpredictable_t32_vector_catch

TAGS **UNPREDICTABLE** DISPLAY Unpredictable Vector catch exception on a T32 instruction at %{ADDR} Offset %{OFFSET}. Fields:

ADDR unsigned int

Address.

OFFSET unsigned int

Indicates which halfword of a 32-bit Thumb instruction the vector catch exception is set on. 0 for the first halfword or if the instruction is 16 bits wide, or 1 for the second halfword.

ArchMsg.Warning.unpredictable_watchpoint_far

TAGS **UNPREDICTABLE** DISPLAY The value for FAR/EDWAR for hitting this watchpoint may be anywhere in the range %{LOWER_BOUND} to %{UPPER_BOUND}. Fields:

LOWER_BOUND unsigned int

the lowest address accessed by the instruction that triggered the watchpoint.

UPPER_BOUND unsigned int

the highest watchpointed address accessed by that instruction.

ArchMsg.Warning.warning_AdvSIMDExpandImmUnexpectedZero

TAGS **UNPREDICTABLE** DISPLAY AdvSIMDExpandImm may treat this immediate value as **UNPREDICTABLE**.

ArchMsg.Warning.warning_ConditionalSMC

TAGS **UNPREDICTABLE** DISPLAY SMC instruction has **UNPREDICTABLE** effects when conditional and when combined with traps.

ArchMsg.Warning.warning_access_crosses_page_boundary

TAGS **UNPREDICTABLE** DISPLAY Access crosses page boundary.

ArchMsg.Warning.warning_access_crosses_page_boundary_has_fault

TAGS **UNPREDICTABLE** DISPLAY Access that crosses page boundary has fault(s).

ArchMsg.Warning.warning_access_crosses_page_boundary_has_fault_on_both_pages

TAGS **UNPREDICTABLE** DISPLAY Access that crosses page boundary has a fault on each page.
Fields:

ADDR1 unsigned int

FAR address on first page.

ADDR2 unsigned int

FAR address on second page.

FSTATUS1 unsigned int

Fault status on first page.

FSTATUS2 unsigned int

Fault status on second page.

ArchMsg.Warning.warning_access_crosses_page_boundary_spanning_different_memory

TAGS **UNPREDICTABLE** DISPLAY Access at address %{ADDR} crosses page from MEM_TYPE %{MEMTYPE_PAGE1} to %{MEMTYPE_PAGE2}. Fields:

ADDR unsigned int

address of access crossing page.

MEMTYPE_PAGE1 signed int

First Page's memory type (0-Normal, 1-Non-Normal).

MEMTYPE_PAGE2 signed int

Second Page's memory type (0-Normal, 1-Non-Normal).

ArchMsg.Warning.warning_access_wraps_around_memory

TAGS **UNPREDICTABLE** DISPLAY Access wraps around memory in %{WIDTH:d}bit mode.
Fields:

WIDTH unsigned int

Address width.

ArchMsg.Warning.warning_bcr_linking_status

TAGS **UNPREDICTABLE** DISPLAY DBGBCR.{BT2,BT} combination gives unpredictable behaviour.

ArchMsg.Warning.warning_bcr_mask_reserved

TAGS **UNPREDICTABLE** DISPLAY DBGBCR_ELn.MASK is set to a reserved value.

ArchMsg.Warning.warning_bvr_masked_bit_not_zero

TAGS **UNPREDICTABLE** DISPLAY DBGBVRn_EL1.VA contains a bit masked by DBGBCRn_EL1.MASK that is not zero.

ArchMsg.Warning.warning_ccsidr_unimplemented_level

TAGS **UNPREDICTABLE** DISPLAY CCSIDR read while CSSELR %{CSSELR:x} points to an unimplemented cache level. Fields:

CSSELR unsigned int

current effective value of CSSELR.

ArchMsg.Warning.warning_change_to_ns_when_tge_set

TAGS **UNPREDICTABLE** DISPLAY Attempting to change NS to 1 when HCR.TGE = 1.

ArchMsg.Warning.warning_contiguous_bit_check_abort

TAGS **UNPREDICTABLE** DISPLAY An abort occurred whilst attempting to check TLB entry is contiguous at %{ENTRY_ADDR}. Fields:

ENTRY_ADDR unsigned int

address of conflicting TLB entry.

ArchMsg.Warning.warning_contiguous_bit_error

TAGS **UNPREDICTABLE** DISPLAY TLB entry at %{ENTRY_ADDR} was not contiguous with entry at %{CONTIG_ADDR} Contents are %{ENTRY_DATA} but expected %{CONTIG_DATA}. Fields:

CONTIG_ADDR unsigned int

address of the TLB first read, with which this entry is expected to be contiguous.

CONTIG_DATA unsigned int

expected contents based on the entry at CONTIG_ADDR.

ENTRY_ADDR unsigned int

address of conflicting TLB entry.

ENTRY_DATA unsigned int

contents of conflicting TLB entry.

ArchMsg.Warning.warning_contiguous_bit_error_D128

TAGS **UNPREDICTABLE** DISPLAY TLB entry at %{ENTRY_ADDR} was not contiguous with entry at %{CONTIG_ADDR} Contents are %{ENTRY_DATA_HI} %{ENTRY_DATA_LO} but expected %{CONTIG_DATA}. Fields:

CONTIG_ADDR unsigned int

address of the TLB first read, with which this entry is expected to be contiguous.

CONTIG_DATA unsigned int

expected contents based on the entry at CONTIG_ADDR.

ENTRY_ADDR unsigned int

address of conflicting TLB entry.

ENTRY_DATA_HI unsigned int

Higher bit contents of conflicting TLB entry.

ENTRY_DATA_LO unsigned int

Lower bit contents of conflicting TLB entry.

ArchMsg.Warning.warning_cp10_cp11_mismatch

TAGS **UNPREDICTABLE** DISPLAY Unpredictable configuration of CP10 and CP11 controls at EL **%{EL:d}**. Fields:

EL unsigned int

EL owning the control.

ArchMsg.Warning.warning_cp10_cp11_reserved_value

TAGS **UNPREDICTABLE** DISPLAY Unpredictable configuration of CPACR CP10 (**%{CP10}**) or CP11 (**%{CP11}**). Fields:

CP10 unsigned int

CP10 access.

CP11 unsigned int

CP11 access.

ArchMsg.Warning.warning_csselr_level_out_of_range

TAGS **UNPREDICTABLE** DISPLAY CSSELR.Level out of range, written value **%{LEVEL}** with only **%{IMPLEMENTED:d}** levels implemented. Fields:

IMPLEMENTED unsigned int

number of cache levels implemented.

LEVEL unsigned int

written value of CSSELR.Level.

TYPE enum

The type of cache selected by CSSELR.InD and TnD.

ArchMsg.Warning.warning_debug_flow_control_bits_not_obeyed

TAGS **UNPREDICTABLE** DISPLAY Flow control bits for **%{REGISTER}** were not obeyed, causing **%{ERROR}**. Fields:

ERROR enum

type of error resulting.

REGISTER enum

register under consideration.

ArchMsg.Warning.warning_debug_register_access_during_reset

TAGS **UNPREDICTABLE** DISPLAY Unpredictable **%{IS_WRITE:(read|write)}** access to debug register offset **%{OFFSET}** during reset. Fields:

IS_WRITE unsigned int

Write Not Read.

OFFSET unsigned int

Register Offset.

ArchMsg.Warning.warning_decode_cps_inconsistent_fields

TAGS **UNPREDICTABLE** DISPLAY The mode setting fields in the CPS instruction are inconsistent.

ArchMsg.Warning.warning_decode_invalid_state

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable in current exception-level/security state.

ArchMsg.Warning.warning_default_cacheable_mmu_on

TAGS **UNPREDICTABLE** DISPLAY Default cacheable is enabled (HCR.DC = 1) while MMU is enabled.

ArchMsg.Warning.warning_default_cacheable_vmmu_off

TAGS **UNPREDICTABLE** DISPLAY Default cacheable is enabled (HCR.DC = 1) while HCR.VM = 0.

ArchMsg.Warning.warning_deprecated_wvr_bit_2_set

TAGS **UNPREDICTABLE** DISPLAY DBGWVR%{N:d}_EL1=%{WVR:h} has bit 2 set, deprecated in ARMv8. Fields:

N unsigned int

index.

WVR unsigned int

DBGWVR.

ArchMsg.Warning.warning_eret_while_software_step_active_pending

Missing ISB between setting MDSCR_EL1.SS and ERET TAGS **UNPREDICTABLE**.

ArchMsg.Warning.warning_exclusive_to_non_normal

TAGS **UNPREDICTABLE** DISPLAY Exclusive Access to Non-Normal Memory %{ADDR}. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_exclusive_to_non_writeback

TAGS **UNPREDICTABLE** DISPLAY Exclusive Access to Non-Writeback-Cacheable Memory %{ADDR}. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_execute_from_device_memory

TAGS **UNPREDICTABLE** DISPLAY An attempt was made to execute from Device memory.

ArchMsg.Warning.warning_illegal_cpsr_mode

TAGS **UNPREDICTABLE** DISPLAY Writing an illegal or unimplement mode (%{NEW_MODE:h}) to CPSR was unpredictable. Fields:

NEW_MODE unsigned int

New value of CPSR.M.

ArchMsg.Warning.warning_illegal_srs_mode

TAGS **UNPREDICTABLE** DISPLAY Illegal or **UNPREDICTABLE** mode (%{MODE:h}) used for SRS instruction. Fields:

MODE unsigned int

mode for Banked SP.

ArchMsg.Warning.warning_implementation_defined_read_debug_register_in_SCS

Read Access to Debug registers from the processor is IMPLEMENTATION DEFINED TAGS IMP_DEF. Fields:

PPB_OFFSET unsigned int

debug register offset.

ArchMsg.Warning.warning_implementation_defined_sequential_security_transitions_supported

The behavior of sequential instruction fetches that cross from non-secure to secure memory and contain an SG instruction is **CONSTRAINED UNPREDICTABLE** TAGS **UNPREDICTABLE**.

ArchMsg.Warning.warning_implementation_defined_stack_limit_check_supported

It is **IMPLEMENTATION DEFINED** whether stack pointer limit checking is performed for this instruction TAGS IMP_DEF. Fields:

ADDRESS unsigned int

Address of instruction.

OPCODE unsigned int

instruction opcode.

ArchMsg.Warning.warning_implementation_defined_taggednonsharedmemorytype

TAGS IMP_DEF DISPLAY Memory attribute has Tagged, non shared memory type; it is **IMPLEMENTATION DEFINED** whether the memory location is treated as Tagged or non-Tagged.

ArchMsg.Warning.warning_implementation_defined_write_debug_register_in_SCS

Write Access to Debug registers from the processor is IMPLEMENTATION DEFINED TAGS IMP_DEF. Fields:

DATA unsigned int

data attempted to be written.

PPB_OFFSET unsigned int

debug register offset.

ArchMsg.Warning.warning_invalid_tcr_granule

TAGS **UNPREDICTABLE** DISPLAY TCR.TG%{TG_ID:(0|1)} bits have been set to a granule size %{REQUEST:d}K not implemented - using %{SUBSTITUTE:d}K. Fields:

REQUEST signed int

page size requested (or 0 for reserved).

SUBSTITUTE signed int

best guess available page size.

TG_ID bool

bits TG0 or TG1.

ArchMsg.Warning.warning_load_multiple_user_registers_from_user_mode

TAGS UNPREDICTABLE DISPLAY An LDM(user registers) instruction executed from user mode.

ArchMsg.Warning.warning_mte_in_mem_attr_not_supported

TAGS UNPREDICTABLE DISPLAY Memory attribute %{ATTR} has Tagged memory type but Armv8.5-MTE is not implemented. Fields:

ATTR unsigned int

Memory attribute.

ArchMsg.Warning.warning_reserved_breakpoint_state_match

TAGS UNPREDICTABLE DISPLAY %{IS_BREAKPOINT:(Watchpoint|Breakpoint)} programmed with a reserved combination of HMC, SSC and %{IS_BREAKPOINT:(PAC|PMC)}. Fields:

IS_BREAKPOINT unsigned int

Is a breakpoint, not a watchpoint.

ArchMsg.Warning.warning_shareability

TAGS UNPREDICTABLE DISPLAY Unpredictable: combination of the 1st and 2nd stages of translation is Normal Inner Non-Cacheable, Outer Non-Cacheable.

ArchMsg.Warning.warning_software_step_set_while_enabled

TAGS UNPREDICTABLE DISPLAY MDSCR_EL1.SS set to 1 while software step debug exceptions are enabled.

ArchMsg.Warning.warning_thumb_instruction_wraps_around_memory

TAGS UNPREDICTABLE DISPLAY Thumb instruction wraps around memory.

ArchMsg.Warning.warning_ttbr_sbz_bits_are_not_zero

TAGS UNPREDICTABLE DISPLAY %{TTBR}[%{MSB:d};0] should be zero, but are not. Fields:

MSB unsigned int

x-1.

TTBR enum

which TTBR register.

ArchMsg.Warning.warning_unaligned_address_dbgdtrrx_write

TAGS UNPREDICTABLE DISPLAY Unaligned address %{ADDR} in X0/R0 when DBGDTRRX written in memory access mode. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unaligned_address_dbgdtrtx_read

TAGS **UNPREDICTABLE** DISPLAY Unaligned address %{ADDR} in XO/RO when DBGDTRTX read in memory access mode. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unaligned_to_device

TAGS **UNPREDICTABLE** DISPLAY Unaligned Access to Device Memory %{ADDR}. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unaligned_to_strongly_ordered

TAGS **UNPREDICTABLE** DISPLAY Unaligned Access to Strongly Ordered Memory %{ADDR}. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unknown_sau_rnr

TAGS **UNPREDICTABLE** DISPLAY SAU_RNR was set to an unsupported value.

ArchMsg.Warning.warning_unpredictable_AIRCR_PRIS_and_BFHFNMINs

TAGS **UNPREDICTABLE** DISPLAY The effect of setting both AIRCR.BFHFNMINs and AIRCR.PRIS to 1 is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_AIRCR_VECTCLRACTIVE_when_not_in_debug

TAGS **UNPREDICTABLE** DISPLAY The effect of writing a 1 to AIRCR.VECTCLRACTIVE if the processor is not halted in Debug state is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_AIRCR_VECTRESET_and_SYSRESETREQ

TAGS **UNPREDICTABLE** DISPLAY When the processor is halted in Debug state, if a write to the register writes a 1 to both VECTRESET and SYSRESETREQ, the behavior is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_AIRCR_VECTRESET_when_not_in_debug

TAGS **UNPREDICTABLE** DISPLAY The effect of writing a 1 to AIRCR.VECTRESET if the processor is not halted in Debug state is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_AIRCR_incorrect_VKEY

TAGS **UNPREDICTABLE** DISPLAY The value 0x05FA must be written to AIRCR.VKEY, otherwise the register write is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_EXC_RETURN_Reserved_Bit

EXC_RETURN[23:7] are reserved with the special condition that all bits should be written as one. Values other than all 1s are **UNPREDICTABLE**. TAGS **UNPREDICTABLE**. Fields:

EXC_RETURN unsigned int

Special PC value.

SBO_MASK unsigned int

Which bits Should Be One to not UNPRED.

ArchMsg.Warning.warning_unpredictable_PMEVTYPErn_ELO_VS_reserved_value

TAGS UNPREDICTABLE DISPLAY Unpredictable reserved value of PMEVTYPER<n>_ELO.VS.

ArchMsg.Warning.warning_unpredictable_change_to_DEMCR_MON_STEP_at_insufficient_priority

TAGS UNPREDICTABLE DISPLAY The effect of changing DEMCR.MON_STEP at an execution priority that is lower than the priority of the DebugMonitor exception is **CONSTRAINED UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_change_to_priority_of_active_exception

TAGS UNPREDICTABLE DISPLAY Changing the priority of an active exception is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_clear_lspact

TAGS UNPREDICTABLE DISPLAY **UNPREDICTABLE** state on context switch. Fields:

DRVEXC_TAKEN unsigned int

Derived exception taken, not pended.

REACHED_LAST_STORE unsigned int

Only the last faulted.

ArchMsg.Warning.warning_unpredictable_dbg_exit_unaligned_dlr

TAGS UNPREDICTABLE DISPLAY Debug exit to AArch32 with DLR[0] set to 1. PC[0] can be set to 0 or to DLR[0].

ArchMsg.Warning.warning_unpredictable_exception_catch

TAGS UNPREDICTABLE DISPLAY Generation of an exception catch event is unpredictable.

ArchMsg.Warning.warning_unpredictable_exception_return_inconsistent_state

TAGS UNPREDICTABLE DISPLAY state on exception return is unpredictable.

ArchMsg.Warning.warning_unpredictable_exception_return_instruction

TAGS UNPREDICTABLE DISPLAY exception return instruction is unpredictable.

ArchMsg.Warning.warning_unpredictable_in_debug_state

TAGS UNPREDICTABLE DISPLAY Instruction is **UNPREDICTABLE** when executed in debug state.

ArchMsg.Warning.warning_unpredictable_mair_encoding

TAGS UNPREDICTABLE DISPLAY Memory attribute %{ATTR} has invalid MAIR encoding; this may lead to unpredictable results. Fields:

ATTR unsigned int

Memory attribute.

ArchMsg.Warning.warning_unpredictable_pmu_counter_access

TAGS UNPREDICTABLE DISPLAY Access to PMU counters from non-secure EL0 or EL1 is **UNPREDICTABLE** with MDCR_EL2.HPMN set to 0.

ArchMsg.Warning.warning_unpredictable_prioritization_breakpoint_match_vector_catch

TAGS UNPREDICTABLE DISPLAY Two events with the same priority occurred at the same instruction: Address Matching Vector Catch debug event, and Breakpoint debug event. It is constrained unpredictable which is taken.

ArchMsg.Warning.warning_unpredictable_stack_selection

TAGS **UNPREDICTABLE** DISPLAY **UNPREDICTABLE** state on context switch.

ArchMsg.Warning.warning_unpredictable_tsize_out_of_range

TAGS **UNPREDICTABLE** DISPLAY **%{TCR}.T%{TTBR:d}SZ (%{TSIZE:d} bits)** is out of range. Must be between 25 and **%{TSIZE_MAX:d}**. Fields:

TCR enum

Which TCR.

TSIZE signed int

Value of TSize.

TSIZE_MAX signed int

The maximum allowed value.

TTBR unsigned int

Which TTBR region.

ArchMsg.Warning.warning_unpredictable_unaligned_pc_as_base_register

TAGS **UNPREDICTABLE** DISPLAY The PC value must be word-aligned, otherwise the behavior of the instruction is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_unaligned_pop_stack

TAGS **UNPREDICTABLE** DISPLAY **CONSTRAINED UNPREDICTABLE** unaligned stack frame **%{FRAME_ADDR:x}** in exception-return. Fields:

FRAME_ADDR unsigned int

unaligned stack frame address.

ArchMsg.Warning.warning_unpredictable_unaligned_to_non_writeback

TAGS **UNPREDICTABLE** DISPLAY Unaligned Access to Non-Writeback-Cacheable Memory.

ArchMsg.Warning.warning_unpredictable_vmsa_memattrib

TAGS **UNPREDICTABLE** DISPLAY Unpredictable vmsa memory attribute with texcb value programmed to **%{TEXCB}** when tex remap is **%{TEX_REMAP}**. Fields:

TEXCB unsigned int

texcb value.

TEX_REMAP bool

use tex remap.

ArchMsg.Warning.warning_unpredictable_vtcr_t0sz_sl0

TAGS **UNPREDICTABLE** DISPLAY The combination of **VTCR.TOSZ=%{TOSZ:d}** and **VTCR.SL0=%{SL0:d}** is unpredictable. Fields:

SL0 signed int

VTCR.SL0, starting level for stage 2 translation table walks.

TOSZ signed int

VTCR.TOSZ, required input address range.

ArchMsg.Warning.warning_unpredictable_write_DHCSR

TAGS UNPREDICTABLE DISPLAY UNPREDICTABLE set of bit-changes in DHCSR. Fields:

NEW unsigned int

data attempted to be written.

OLD unsigned int

previous value.

ArchMsg.Warning.warning_unpredictable_write_SHCSR

TAGS UNPREDICTABLE DISPLAY UNPREDICTABLE setting a pending bit to 1 for an exception with priority >= execution priority.

ArchMsg.Warning.warning_unsupported_access_to_memory_mapped_register

TAGS UNPREDICTABLE DISPLAY Access to memory mapped register is unsupported.

ArchMsg.Warning.warning_user_jmcr_access

TAGS UNPREDICTABLE DISPLAY User %{WRITE:(write|read)} access to JMCR is unpredictable. Fields:

WRITE unsigned int

access is write.

ArchMsg.Warning.warning_wcr_mask_and_bas

TAGS UNPREDICTABLE DISPLAY DBGWCR_EL1.MASK is non-zero and BAS is not 0xff.

ArchMsg.Warning.warning_wcr_mask_reserved

TAGS UNPREDICTABLE DISPLAY DBGWCR_ELn.MASK is set to a reserved value.

ArchMsg.Warning.warning_wcr_non_configuous_bas

TAGS UNPREDICTABLE DISPLAY DBGWCR_EL1.BAS has non-contiguous set of ones.

ArchMsg.Warning.warning_wvr_masked_bit_not_zero

TAGS UNPREDICTABLE DISPLAY DBGWVRn_EL1.VA contains a bit masked by DBGWCRn_EL1.MASK that is not zero.

ArchMsg.warning_unpred_system_register_access

DISPLAY Unpredictable %{IS_WRITE:(read|write)} access to system register offset %{OFFSET}. Fields:

IS_WRITE unsigned int

Write Not Read.

OFFSET unsigned int

Register Offset.

BRANCH_MISPREDICT

Simulating branch mispredict. Fields:

PC unsigned int

Origin address (or 0 if unavailable).

BRA_DIR

Direct branches, to immediate address. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISSET enum

The instruction set of the branch instruction.

IS_COND bool

Indicates if this is a conditional branch.

IS_HINTED bool

Whether the conditional instruction is hinted.

IS_ISB bool

Whether the instruction is ISB.

IS_LINK bool

Indicates if this is a branch with link.

PC unsigned int

The address of the branch instruction.

TARGET_ISET enum

The instruction set after the branch.

TARGET_PC unsigned int

The address the instruction branches to.

BRA_INDIR

Indirect branches, perhaps to a register. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISSET enum

The instruction set of the branch instruction.

IS_COND bool

Indicates if this is a conditional branch.

IS_HINTED bool

Whether the conditional instruction is hinted.

IS_ISB bool

Whether the instruction is ISB.

IS_LINK bool

Indicates if this is a branch with link.

PC unsigned int

The address of the branch instruction.

TARGET_ISET enum

The instruction set after the branch.

TARGET_PC unsigned int

The address the instruction branches to.

BRA_INDIR_CCFAIL

Indirect branches fail, perhaps to a register.

BRA_RET_CCFAIL

Branch return fail, perhaps to a register.

CACHE_MAINTENANCE_OP

Cache Maintenance Operation. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

DATA unsigned int

Specified MVA or Set/way.

FUNCTION enum

Clean / Invalidate.

SCOPE enum

Affected region.

SIDE enum

Inst / Data.

CCFAIL

Conditional instruction condition check fail. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CCFAIL_UNC

Unconditional instruction condition check fail. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CCPASS

Conditional instruction condition check pass. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CCPASS_UNC

Unconditional instruction condition check pass. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CODE_CACHE_MAINT

Code cache maintenance. Fields:

END_INCL unsigned int

Inclusive end address for VA or PA requests.

NS bool

Non-secure world for PA.

START unsigned int

Start address for VA or PA requests.

TYPE enum

Request type.

COMPILE_BLOCK_END

Last instruction of basic block translated. Fields:

VADDR unsigned int

Address of next instruction after this basic block.

COMPILE_BLOCK_START

First instruction of basic block translated. Fields:

VADDR unsigned int

Address of first instruction in basic block.

COMPILE_INST

ARM instruction compiled. Fields:

DISASS string

Disassembly of the instruction.

ISSET enum

The instruction set of this instruction.

ITSTATE unsigned int

The ITSTATE current for the instruction.

OPCODE unsigned int

The opcode of the instruction.

PC unsigned int

The address of the instruction.

SIZE unsigned int

The size of the instruction in bytes.

CONTEXTIDR

Context ID Register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

NS bool

Secure or nonsecure banked register is accessed.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The new value written.

CONTEXT_SYNC

Called for every context synchronization event. Fields:

ARCH_CSE bool

Architectural Context Synchronization Event.

PC unsigned int

The address of the synchronization instruction.

SYSREG_SYNC bool

System register synchronization event.

CORE_ENDIAN

Core BE8 Big-Endian state changed. Fields:

BE8 bool

Core BE8 Big-Endian state.

CORE_INFO

Static processor attributes. Only triggered by a call to DumpState(). Fields:

ARCH_PROFILE enum

The architecture profile of the core.

CLUSTER_ID unsigned int

The cluster ID of this processor.

CORE_NUM unsigned int

The number of this core in an MP processor.

DETAILED_CORE_FEATURES enum

Detailed core features as a bitfield. Enum values are bitmasks to extract the info from the field.

FPU_VERSION enum

The VFP version implemented by the core.

MEM_ARCH enum

The memory architecture of the core.

MIDR_EL1 unsigned int

MIDR_EL1 value.

NUM_CORES unsigned int

The number of cores in this MP processor.

QUANTUM_SIZE unsigned int

The default quantum size of the core (in ticks).

SECURITY_FEATURES bool

Does the core have security features?.

CORE_LOADS

Processor load accesses. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ACQREL enum

Is this an acquire/release.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DATA unsigned int

The data read or written.

ELEMENT_SIZE unsigned int

Width of each element.

ENDIANNESS enum

Byte ordering of this access.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

LOCK enum

Normal, exclusive or locked access.

MECID unsigned int

Memory Encryption Context ID, if applicable.

MECID2 unsigned int

Memory Encryption Context ID of the second page, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PAGE_SIZE unsigned int

Page size as $\log_2(\text{size in bytes})$.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK, 2=Exclusive Failed, 3=Unsupported.

SIZE unsigned int

Width of the access in bytes. Only required if DATA is not traced.

TRANS bool

Is this a translated access.

VADDR unsigned int

The virtual address of the access.

CORE_REGS

Changes of the core registers R0 to R14. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number, 0 to 14.

OLD_VALUE unsigned int

The old value overwritten.

PHYS_ID enum

The physical register accessed.

VALUE unsigned int

The new value written to the register.

CORE_STALLED_TRANSITION

Detect when a core is prevented from progressing PChannel states. Fields:

CURRENT_STATE enum

Current power mode of core.

REASON string

The reason a transition was stalled, and what must be done to unblock it.

TARGET_STATE enum

Desired power mode of core after transition.

CORE_STORES

Processor store accesses. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ACQREL enum

Is this an acquire/release.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DATA unsigned int

The data read or written.

ELEMENT_SIZE unsigned int

Width of each element.

ENDIANNESS enum

Byte ordering of this access.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

LOCK enum

Normal, exclusive or locked access.

MECID unsigned int

Memory Encryption Context ID, if applicable.

MECID2 unsigned int

Memory Encryption Context ID of the second page, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PAGE_SIZE unsigned int

Page size as $\log_2(\text{size in bytes})$.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK, 2=Exclusive Failed, 3=Unsupported.

SIZE unsigned int

Width of the access in bytes. Only required if DATA is not traced.

TRANS bool

Is this a translated access.

VADDR unsigned int

The virtual address of the access.

CRYPTO_SPEC

Every crypto instruction speculatively executed.

DATA_CACHE_SET_ALLOC_TAGS

Set allocation tags from DC GVA instruction. Fields:

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

NSDESC unsigned int

The security state of the access.

PADDR unsigned int

The physical address of the access.

PAS enum

The physical address space of the page.

RESPONSE enum

0=Aborted, 1=OK.

SIZE unsigned int

The size of memory in bytes.

TAG unsigned int

The value of the tag being set.

TAGGING_TYPE enum

Type of Memory tagging (Physical or Virtual).

TAG_VADDR unsigned int

The virtual address of the Virtual Tag if Virtual tagging is used; otherwise 0.

VADDR unsigned int

Virtual address used for setting allocation tags.

DATA_CACHE_ZERO

Zero and invalidate cache line from DC ZVA instruction. Fields:

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

NSDESC unsigned int

The security state of the access.

PADDR unsigned int

The physical address of the access.

PADDR_TAG unsigned int

If tag checked, the physical address tag for the access.

PAS enum

The physical address space of the page.

RESPONSE enum

0=Aborted, 1=OK.

SIZE unsigned int

The size of zero'd memory in bytes.

TAG_CHECKED bool

Is this access tag checked.

VADDR unsigned int

Virtual address of the zero'd cache line.

DEBUG_EVENT

Hardware debug support event. Fields:

EVENT enum

Description of event.

VALUE unsigned int

data value.

DMI_ALLOCATE

DMI allocated. Fields:

ACCESS_DIRECTION enum

READ / WRITE.

PADDR unsigned int

The physical address of the DMI.

PTR unsigned int

DMI range pointer.

SIDE enum

DSIDE / ISIDE.

SIZE unsigned int

The size of DMI.

VADDR unsigned int

The virtual address of the DMI..

DMI_HIT

DMI hit, access used DMI (only fire for small DMI regions i.e. less than a full page). Fields:

ACCESS_DIRECTION enum

READ / WRITE.

PADDR unsigned int

The physical address of the DMI.

SIDE enum

DSIDE / ISIDE.

SIZE unsigned int

The size of the DMI.

VADDR unsigned int

The virtual address of the DMI..

DMI_REVOKE

DMI revoked. Fields:

ACCESS_DIRECTION enum

READ / WRITE.

PADDR unsigned int

The physical address of the DMI.

SIDE enum

DSIDE / ISIDE.

SIZE unsigned int

The size of DMI.

VADDR unsigned int

The virtual address of the DMI..

DTCM_ACCESS

Trace TCM accesses.

DTCM_ACCESS_RAZ_WI

Trace TCM accesses. Fields:

ADDR unsigned int

Address of the **RAZ/WI** access.

DWT_MATCH

DWT comparator matches. Fields:

NUM unsigned int

DWT comparator number.

TYPE enum

DWT comparator configuration.

END_COMPILE

Compilation end. Fields:

END_OF_PAGE_COUNT unsigned int

Number of basic blocks exited due to page boundary since **START_COMPILE**.

FETCHFAIL_COUNT unsigned int

Number of basic blocks exited due to ifetch failure **START_COMPILE**.

INST_COUNT unsigned int

Number of instructions compiled since **START_COMPILE**.

NONSEQ_COUNT unsigned int

Number of basic blocks exited due to non-sequential instructions since **START_COMPILE**.

PAGE_STRADDLE_COUNT unsigned int

Number of basic blocks exited due to unaligned instructions crossing page since **START_COMPILE**.

EXCEPTION

Exceptions that are taken. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ISSET enum

The instruction set of the processor when the exception occurred.

LR unsigned int

The value assigned to the link register.

PC unsigned int

The location where the exception occurred.

TARGET_ISET enum

The instruction set of the exception handler code.

TARGET_PC unsigned int

The address the exception branches to.

VECTOR enum

The exception vector.

EXCEPTION_END

Every exception completed.

EXCEPTION_ENTRY

Event marking the entry of an exception. All pushes of registers to the stack will follow this event. Fields:

PC unsigned int

The location where the exception occurred.

VECTOR enum

The exception that occurred.

EXCEPTION_RAISE

Every exception raised.

EXCEPTION_RETURN

Marks the end of an exception, but on an M core does not indicate a program flow change (branch). In most situations it will be followed by an indirect branch. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISET enum

The instruction set of the branch instruction.

PC unsigned int

The address of the branch instruction.

EXCEPTION_RETURN_PREBRANCH

Exception return event that must occur before PC and processor state have updated.

EXCEPTION_START

Every exception started.

EXCEPTION_VECTOR_FETCH

I-side vector fetch for M-class exception. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

DATA unsigned int

Result of fetch if OK.

RESPONSE enum

0=Aborted, 1=OK.

VADDR unsigned int

Where the vector is fetched from (including VTOR).

FP_STATE

Floating point state. Fields:

FPSR unsigned int

FPSR register value.

MASK unsigned int

The mask indicating updated cumulative bits.

FREQ_CHANGED

Core frequency changed. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The instruction count of this core.

NEW_FREQ signed int

The new frequency of this core (expressed in Hz).

OLD_FREQ signed int

The old frequency of this core (expressed in Hz).

HLT

HLT instruction occurred. Fields:

IMM16 unsigned int

The 16-bit immediate value encoded in HLT instruction.

INFO_EXCEPTION_REASON

Provide information of real cause of exception. Fields:

FaultCause enum

Which bit of which V7M/V8M-mainline FSR/HFSR would have been set.

PC unsigned int

The location where the exception occurred.

PHASE enum

What we are doing with the exception.

REASONS enum

Why we are doing it.

VECTOR enum

The exception that occurred.

INFO_FP_CONTEXT

Inform when ExecuteFPCheck writes CONTROL.FPCA and FPSCR etc. Fields:

WHAT enum

What we are doing.

INFO_STACKING

Provide information of real cause of exception. Fields:

PHASE enum

Are we starting or ending a sequence of stack writes.

REASONS enum

Why we are doing it.

INST

Every instruction executed. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DEBUG_STATE bool

The instruction is executed in debug state.

DISASS string

Disassembly of instruction.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISSET enum

The current instruction set.

ITSTATE unsigned int

The current ITSTATE.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

MODE enum

The mode the core is in.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

OPCODE unsigned int

The opcode of the instruction.

PADDR unsigned int

The physical address of the instruction.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the instruction.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the page.

PC unsigned int

The address of the instruction.

SECURITY_STATE enum

The core's security state.

SIZE unsigned int

The size of the instruction in bytes.

INST_END

Every instruction completed.

INST_START

Every instruction started. Fields:

DEBUG_STATE bool

The instruction is executed in debug state.

ISET enum

The current instruction set.

MODE enum

The mode the core is in.

NS enum

The current Secure State.

PC unsigned int

The address of the conditional instruction.

SECURITY_STATE enum

The current Security State.

INST_STRADDLE

Instruction straddles boundary. Fields:

DEBUG_STATE bool

The instruction is executed in debug state.

ISET enum

The current instruction set.

MODE enum

The mode the core is in.

NS enum

The current Secure State.

PC unsigned int

The address of the conditional instruction.

SECURITY_STATE enum

The current Security State.

ISB

ISB instruction occurred. Fields:

PC unsigned int

The address of the ISB instruction.

ITCM_ACCESS

Trace TCM accesses.

ITCM_ACCESS_RAZ_WI

Trace TCM accesses. Fields:

ADDR unsigned int

Address of the **RAZ/WI** access.

ITM

Instrumentation Trace Macrocell. Fields:

ITM_PACKET_TYPE enum

ITM and DWT packets type.

PACKET_HEADER unsigned int

ITM Packet Header.

PACKET_PAYLOAD unsigned int

ITM Packet Payload.

LOAD_MULTIPLE_REGS_LIST

List of destination registers (not source) of a multiple registers load instruction. Fields:

REG string

Register to add into the list.

LOCAL_MONITOR

Local monitor activity. Fields:

PADDR unsigned int

Local Monitor Address.

State enum

State of the monitor (Open/Exclusive).

LOCKUP_CYCLE

This event is triggered if the core enters or stays in the lock-up state. Fields:

CAUSE enum

What causes the lockup?.

LOCKUP_ENTRY

This event is triggered if the core enters the lock-up state. Fields:

PC unsigned int

The PC when the lockup state is entered.

LOCKUP_EXIT

This event is triggered if the core leaves the lock-up state.

MODE_CHANGE

Mode change. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

MODE enum

The new mode.

OLD_MODE enum

The old mode.

MPU_TRANS

Address translation information. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

EXEC_PERM enum

Execution Permission.

INNERCACHE_RA bool

Is the inner cache allocate on read.

INNERCACHE_TYPE enum

Inner Caching scheme (NC/MB/WA).

INNERCACHE_WA bool

Is the inner cache allocate on write.

MEMTYPE enum

Memory type.

OUTERCACHE_RA bool

Is the outer cache allocate on read.

OUTERCACHE_TYPE enum

Outer Caching scheme (NC/MB/WA).

OUTERCACHE_WA bool

Is the outer cache allocate on write.

PADDR unsigned int

Address of the access.

PAGESIZE unsigned int

Page size as $\log_2(\text{size})$.

READ_PERM enum

Read Permission.

REG_NUM unsigned int

MPU register based region-number 0-15 (else 0xFF=none, 0xBB/0xDD=background/default).

SH enum

Shareability.

SIDE enum

Inst / Data.

WRITE_PERM enum

Write Permission.

PERIODIC

Called for every quantum. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The instruction count of this CPU.

PC unsigned int

The address of the next instruction to be executed on this CPU.

POTINDEX_UPDATE

Value of POT index for current pc changed. Fields:

NEW_POTINDEX unsigned int

New POTIndex.

OLD_POTINDEX unsigned int

Old POTIndex.

PPB_READ_ACCESS

Trace reads from the core to memory mapped registers in the PPB (private peripheral bus) address range. Fields:

ACCESS_SIZE unsigned int

Logarithm of the access size: 0=byte, 1=halfword, 2=word, 3=doubleword, ...

BURST_LEN unsigned int

The number of data transfers (beats) in this burst.

DATA unsigned int

The data transferred.

INNER_ATTRIBUTES unsigned int

The inner attributes: bit0=write allocate, bit1=read allocate, bit2=cachable, bit3=bufferable, bit4=shareable.

INST bool

Is an instruction fetch.

LOCK enum

Normal / Locked / Exclusive.

MANAGER_ID unsigned int

The AXI manager ID.

NS bool

Is a nonsecure access.

OUTER_ATTRIBUTES unsigned int

The outer attributes: bit0=write allocate, bit1=read allocate, bit2=cachable, bit3=bufferable, bit4=shareable.

PADDR unsigned int

Physical address of access.

PRIV bool

Is privileged access.

RESPONSE enum

Whether the transaction was successful, or an error occurred.

USER_FLAGS unsigned int

Core specific additional signals.

PPB_WRITE_ACCESS

Trace writes from the core to memory mapped registers in the PPB (private peripheral bus) address range. Fields:

ACCESS_SIZE unsigned int

Logarithm of the access size: 0=byte, 1=halfword, 2=word, 3=doubleword, ...

BURST_LEN unsigned int

The number of data transfers (beats) in this burst.

DATA unsigned int

The data transferred.

INNER_ATTRIBUTES unsigned int

The inner attributes: bit0=write allocate, bit1=read allocate, bit2=cachable, bit3=bufferable, bit4=shareable.

INST bool

Is an instruction fetch.

LOCK enum

Normal / Locked / Exclusive.

MANAGER_ID unsigned int

The AXI manager ID.

NS bool

Is a nonsecure access.

OUTER_ATTRIBUTES unsigned int

The outer attributes: bit0=write allocate, bit1=read allocate, bit2=cachable, bit3=bufferable, bit4=shareable.

PADDR unsigned int

Physical address of access.

PRIV bool

Is privileged access.

RESPONSE enum

Whether the transaction was successful, or an error occurred.

USER_FLAGS unsigned int

Core specific additional signals.

PRELOAD_DATA

Data preload from PLD instruction. Fields:

VADDR unsigned int

Virtual address of the data that should be preloaded.

PRELOAD_INST

Instruction preload from PLI instruction. Fields:

VADDR unsigned int

Virtual address of the instruction that should be preloaded.

PRE_CORE_LOAD

Trace just before a core load. Fields:

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

PRE_CORE_STORE

Trace just before a core store. Fields:

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

PSTATE_PACTIVE_OPMODE_TRANSITION

Trace pactive and Opmode mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Operating mode after the transition.

OLD_STATE enum

Operating mode before the transition.

PSTATE_PACTIVE_OPMODE_TRANSITION_DENIED

Trace the denied pactive and Opmode mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Operating mode after the transition.

OLD_STATE enum

Operating mode before the transition.

REASON string

Reason for the transition being denied.

PSTATE_PACTIVE_POWER_TRANSITION

Trace pactive and power mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Power mode after the transition.

OLD_STATE enum

Power mode before the transition.

PSTATE_PACTIVE_POWER_TRANSITION_DENIED

Trace the denied pactive and power mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Power mode after the transition.

OLD_STATE enum

Power mode before the transition.

REASON string

Reason for the transition being denied.

PSTATE_PACTIVE_THREAD_OPMODE_TRANSITION

Trace pactive and Opmode mode transitions for threads on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Operating mode after the transition.

OLD_STATE enum

Operating mode before the transition.

QUANTUM_START

Called at the start of every quantum.

RUN_STATE

Run state transition. Fields:

INST_COUNT unsigned int

Ticks count at point of transition.

NEW enum

New run state.

OLD enum

Old run state.

SEMIHOSTING_CALL

Call of a semihost function occurred. Fields:

PC unsigned int

The program counter after the semihosting call.

SEMIHOSTING_PRECALL

About to call semihost. Fields:

EL2 bool

Use EL2 translation regime.

NS bool

Is Non-Secure.

REG_WIDTH unsigned int

The current register width in bytes.

SIGNAL

External signal state change. Fields:

SIGNAL enum

Signal that changed.

STATE bool

Signal asserted state.

START_COMPILE

Compilation started. Fields:

VADDR unsigned int

Instruction where compilation begins.

STG_MULTIPLE_MICRO_OPS_ADDRESS_RANGE

Address range for MTE stores with multiple uops. Fields:

ADDR_END unsigned int

Final address.

ADDR_START unsigned int

Starting address.

STORE_MULTIPLE_REGS_ADDRESS_RANGE

Address range for stores with multiple registers. Fields:

ADDR_END unsigned int

Final address.

ADDR_START unsigned int

Starting address.

SUBOPTIMAL_LDST_RETIRED

suboptimal load/store. Fields:

ACCESS_SIZE unsigned int

Log2 of the access width in bytes used for alignment checking (1: half-word, 2: word ...).

OPTIMAL_ALIGN_SELECTOR unsigned int

Selector to decide the optimal alignment size (1: access width, 2: 4-bytes, 3: 8-bytes, ..., 12: 4-kbytes).

VADDR unsigned int

The virtual address of the access.

SYNC

Called for every synchronization, that is every quantum break. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The instruction count of this CPU.

LOCAL_QUANTUM unsigned int

The local quantum of this CPU.

LOCAL_TIME unsigned int

The local time of this CPU.

SYNC_REASON enum

Reason for sync.

SYSCALL

System call instruction executed. Fields:

IMM unsigned int

Immediate value of the system call instruction.

TYPE enum

System call type.

VADDR unsigned int

Instruction that caused the system call.

SYSREG_READ

Trace reads from system registers. Fields:

REG_NAME enum

Register number.

VALUE unsigned int

The value read.

SYSREG_UPDATE128

Triggers when the system updates a register. Fields:

REG enum

Register number.

UNKNOWN unsigned int

Bits of the register which became unknown.

VALUE unsigned int

Value written to the register.

SYSREG_UPDATE32

Triggers when the system updates a register. Fields:

REG enum

Register number.

UNKNOWN unsigned int

Bits of the register which became unknown.

VALUE unsigned int

Value written to the register.

SYSREG_UPDATE64

Triggers when the system updates a register. Fields:

REG enum

Register number.

UNKNOWN unsigned int

Bits of the register which became unknown.

VALUE unsigned int

Value written to the register.

SYSREG_WRITE

Trace write to system registers. Fields:

REG_ACCESS enum

Register access status.

REG_NAME enum

Register number.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

Value written to the register.

UNALIGNED_LDST_RETIRED

Processor unaligned load/store. Fields:

ACCESS_SIZE unsigned int

Log2 of the access width in bytes used for alignment checking (1=>half-word, 2->word ...).

NUMBER_OF_BEATS unsigned int

Number of accesses (beats) in this burst (Total size in bytes is ACCESS_SIZE * NUMBER_OF_BEATS).

VADDR unsigned int

The virtual address of the access.

VFP_D_REGS

VFP/NEON D 64 bit register write. Fields:

ALIASING enum

Type of register aliasing used.

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a write to the corresponding bit in the register. For example, 0x1 would indicate a write to VALUE[0].

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

VFP_D_REGS_READ

VFP/NEON D 64 bit register read. Fields:

ID unsigned int

The register number.

VALUE unsigned int

The current value of the register.

VFP_Q_REGS

VFP/NEON Q 128 bit register write. Fields:

ALIASING enum

Type of register aliasing used.

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a write to the corresponding byte in the register. For example, 0x1 would indicate a write to VALUE[7:0].

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

VFP_Q_REGS_READ

VFP/NEON Q 128 bit register read. Fields:

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a read to the corresponding byte in the register. For example, 0x1 would indicate a read to VALUE[7:0].

VALUE unsigned int

The current value of the register.

VFP_SYS_REGS

Writes to the VFP/NEON units system registers. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ID enum

Which VFP system register is written.

MASK unsigned int

The mask indicating updated cumulative bits.

OLD_VALUE unsigned int

The register's old value overwritten.

VALUE unsigned int

The new value written to the VFP system register.

VFP_S_REGS

VFP/NEON S 32 bit register write. Fields:

ALIASING enum

Type of register aliasing used.

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a write to the corresponding bit in the register. For example, 0x1 would indicate a write to VALUE[0].

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

VFP_S_REGS_READ

VFP/NEON S 32 bit register read. Fields:

ID unsigned int

The register number.

VALUE unsigned int

The current value of the register.

WAYPOINT

Signals end of basic block execution. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ISET enum

Origin instruction set.

IS_COND bool

Indicates if this is a conditional waypoint.

PC unsigned int

Origin address (or 0 if unavailable).

TAKEN bool

Indicates if this waypoint was taken.

TARGET unsigned int

Destination address.

TARGET_ISET enum

Destination instruction set.

WFE_END

WFE ended. Fields:

INST_COUNT unsigned int

Ticks count when leaving WFE.

WFE_EVENT_REGISTER

WFE event register status: set/clear, reason. Fields:

INST_COUNT unsigned int

Ticks count.

REASON enum

Reason for set/clear. Only REASON==0 (Cleared by WFE) clears the bit, all other reasons set it.

WFE_IGNORED

WFE ignored. Fields:

EVENT bool

This WFE was ignored because the event register was set.

INST_COUNT unsigned int

Ticks count when ignoring WFE.

TRAPPED bool

This WFE was trapped.

WFE_START

WFE entered. Fields:

INST_COUNT unsigned int

Ticks count when entering WFE.

WFI_END

WFI ended. Fields:

INST_COUNT unsigned int

Ticks count when leaving WFI.

WFI_IGNORED

WFI ignored. Fields:

DISABLED bool

This WFI was ignored because WFI is disabled.

INST_COUNT unsigned int

Ticks count when ignoring WFI.

REASON enum

specifies reason why WFI trace was ignored.

TRAPPED bool

This WFI was trapped.

WFI_START

WFI entered. Fields:

INST_COUNT unsigned int

Ticks count when entering WFI.

WFI_WAKEUP

WFI wakeup. Fields:

INST_COUNT unsigned int

Ticks count when WFI wakeup occurred.

REASON enum

Reason for wakeup.

XPSR

Changes to the xPSR register. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

OLD_VALUE unsigned int

The old xPSR value.

UNKNOWN unsigned int

Bits within the register that have unknown value.

VALUE unsigned int

The new xPSR value.

2.48 ARM_Cortex-M85

This section describes the trace sources.

ASYNC_MEMORY_FAULT

Context ID Register write. Fields:

FAULT unsigned int

Fault status present in ESR.ISS encoding for Error in ARCH64 and present in DFSR encoding for external abort format in ARCH32.

PADDR unsigned int

Physical Address (or 0 if unavailable).

ATOMIC_END_ACCESS

Bus trace access for atomics. Fields:

ACCESS_FAIL bool

Memory access failed.

ADDR unsigned int

The virtual address of the access.

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

COMPARE_VALUE unsigned int

Compare value for CAS.

LOAD_VALUE unsigned int

Loaded values.

NSDESC unsigned int

The security state of the access.

OPERAND_VALUE unsigned int

Operation's operand.

OPERATION enum

Operation type.

PADDR unsigned int

The physical address of the access.

PRIV bool

Is this a privileged access?.

ATOMIC_SLAVE_ACCESS

Information on the atomic operation at slave side. Fields:

ACCESS_RESULT unsigned int

Result of atomic access.

COMPARE_VALUE unsigned int

Value to be compared with memory by CAS.

LOAD_VALUE unsigned int

Value to be loaded from memory as a result of atomic operation.

MANAGER unsigned int

ID of manager which initiated the transaction.

NS bool

Transaction is Non-Secure.

OPERAND_VALUE unsigned int

Incoming operand value provided by core.

OPERATION enum

Type of atomic operation.

PADDR unsigned int

Full physical address of TCM.

STORE_VALUE unsigned int

Value to be stored to memory as a result of atomic operation.

ATOMIC_START_ACCESS

Bus trace access for atomics. Fields:

ADDR unsigned int

The virtual address of the access.

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

COMPARE_VALUE unsigned int

Compare value for CAS.

NSDESC unsigned int

The security state of the access.

OPERAND_VALUE unsigned int

Operation's operand.

OPERATION enum

Operation type.

PADDR unsigned int

The physical address of the access.

PRIV bool

Is this a privileged access?.

ArchMsg.Error.exit_code

DISPLAY exit code: %{EXITCODE} component kind: %{KIND} reason: %{REASON}. Fields:

EXITCODE signed int

Exit code to be returned.

KIND string

Component kind that invoked the exit code trace.

REASON string

Reason why the exit code trace was invoked (optional).

ArchMsg.Warning.Unpredictably Indexed PM Event RegisterTAGS **UNPREDICTABLE** DISPLAY Event %{IsDirect:(Selector value|Register)} %{SEL} is >= %{N}.

Fields:

IsDirect unsigned int

Direct Access rather than indirect.

N unsigned int

Number of Accessible Registers.

SEL unsigned int

Selector.

ArchMsg.Warning.cache_contents_unknown

A cache has been activated without being invalidated and may contain **UNKNOWN** content. DISPLAY %{SIDE:(D|I)}-Cache was enabled in %{REGIME:(SECURE|NON_SECURE|HYP)} regime but was not invalidated since power-on: cache lines could contain **UNKNOWN** content %{SIDE:(Note: D-side caches must invalidate SECURE lines even if the cache is only used by NON_SECURE code in order to ensure that unexpected dirty lines tagged SECURE are not naturally evicted)}. Fields:

REGIME unsigned int

Current translation regime.

SIDE unsigned int

0 d-side cache or unified cache, 1 i-side.

ArchMsg.Warning.dap_csw_bad_size

A write to CM3DAP CSW has an invalid size field. DISPLAY Write %{DATA} to CM3DAP CSW has invalid size field. Fields:

DATA unsigned int

bits[2:0] are size.

ArchMsg.Warning.dcimvac_matches_watchpoint

TAGS UNPREDICTABLE DISPLAY Watchpoints matching an AArch32 DCIMVA are implementation defined.

ArchMsg.Warning.decode_fieldmismatch

TAGS UNPREDICTABLE DISPLAY Instruction is unpredictable with different fields %{FIELD1} and %{FIELD2}. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_frachitsnegative

TAGS UNPREDICTABLE DISPLAY VCVT Instruction is unpredictable with a negative number of fraction bits.

ArchMsg.Warning.decode_initblock

TAGS UNPREDICTABLE DISPLAY Instruction is unpredictable in an IT block.

ArchMsg.Warning.decode_initblocknotlast

TAGS UNPREDICTABLE DISPLAY Instruction is unpredictable in an IT block when not the last.

ArchMsg.Warning.decode_invalid_condition

TAGS UNPREDICTABLE DISPLAY Instruction is unpredictable when the condition is not AL.

ArchMsg.Warning.decode_invalidfieldcombination

TAGS UNPREDICTABLE DISPLAY Instruction is unpredictable with the values in %{FIELD1} and %{FIELD2}. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_invalidvalue

TAGS UNPREDICTABLE DISPLAY Instruction is unpredictable with the value in %{FIELD}. Fields:

FIELD enum

Field name.

ArchMsg.Warning.decode_mem_hint_unallocated

TAGS UNPREDICTABLE DISPLAY There is no memory hint allocated to this bit pattern.

ArchMsg.Warning.decode_registermatch

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with %{FIELD1} and %{FIELD2} as the same register. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_registermismatch

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with %{FIELD1} and %{FIELD2} as different registers. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_registeroutofrange

TAGS **UNPREDICTABLE** DISPLAY Floating-point instruction is unpredictable indexing beyond the end of the register bank.

ArchMsg.Warning.decode_sbzsbo

TAGS **UNPREDICTABLE** DISPLAY Reserved bits in the instruction are not canonical.

ArchMsg.Warning.decode_transactiontoobig

TAGS **UNPREDICTABLE** DISPLAY Attempt to perform a memory transaction of more than 128 bytes.

ArchMsg.Warning.decode_unpred_other

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable.

ArchMsg.Warning.decode_unpreduseofpc

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable when %{FIELD} is PC. Fields:

FIELD enum

Field name.

ArchMsg.Warning.decode_unpreduseofr13

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable when %{FIELD} is R13. Fields:

FIELD enum

Field name.

ArchMsg.Warning.decode_unpreduseofr13inreglist

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with R13 in register-list.

ArchMsg.Warning.decode_useofd16orhigher

TAGS **UNPREDICTABLE** DISPLAY Attempt to use register D16 or higher where it is not allowed.

ArchMsg.Warning.decode_writebackandbaseinlist

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with write back when the base register is in the transfer list.

ArchMsg.Warning.decode_zeroregistersinlist

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable when no registers are to be transferred.

ArchMsg.Warning.extension_unpred

Probably unpred in IT state or decod. Fields:

PC unsigned int

Address of this instruction.

ArchMsg.Warning.msr_zeroes_ltpsize

May be pre-MVE setup code.

ArchMsg.Warning.pmu_and_dwt

DISPLAY Trying to access a %{IS_PMU:(DWT|PMU)} register while %{IS_PMU:(PMU|DWT)} is also active. Fields:

IS_PMU bool

Whether trying to access the PMU or the DWT.

ArchMsg.Warning.recursive_branch

An instruction is performing a branch that targets the same instruction. If this was intended then a WFI might be more effective DISPLAY recursive branch detected at PC=%{PC}. PRIMARY KEY PC. Fields:

PC unsigned int

Address of instruction causing the branch.

ArchMsg.Warning.recursive_exception

An instruction has generated an exception that targets the same instruction. DISPLAY recursive %{TYPE} exception detected at PC=%{PC} (CPSR=%{CPSR}). PRIMARY KEY PC. Fields:

CPSR unsigned int

Processor state.

PC unsigned int

Address of instruction causing the exception.

TYPE enum

Type of exception.

ArchMsg.Warning.reentrant_vector_catch

TAGS **UNPREDICTABLE** DISPLAY Vector catch debug event on Prefetch or Data abort vector %{ADDR}. Fields:

ADDR unsigned int

vector address.

ArchMsg.Warning.reserved_it_state

TAGS **UNPREDICTABLE** DISPLAY Execution with reserved IT state %{ITSTATE}. Fields:

ITSTATE unsigned int
reserved IT state.

ArchMsg.Warning.secure_vector_fetch_from_nonsecure

DISPLAY Secure vector table is being fetched from Non-secure memory.

ArchMsg.Warning.sg_in_it_block

Arm recommends that software does not place SG (security gate instruction) inside an IT instruction block.

ArchMsg.Warning.unpred_FAR_on_LRCPC3_non_atomic_mem_access

TAGS **UNPREDICTABLE** Unpredictable FAR/FSTATUS on LRCPC3 pair instructions resulting in two separate memory accesses
DISPLAY Unpredictable FAR on LRCPC3 non atomic memory access at %{ADDR1} with FSTATUS %{FSTATUS1} or %{ADDR2} with FSTATUS %{FSTATUS2}. Fields:

ADDR1 unsigned int
Possible FAR address 1.

ADDR2 unsigned int
Possible FAR address 2.

FSTATUS1 unsigned int
Possible fault status value 1.

FSTATUS2 unsigned int
Possible fault status value 2.

ArchMsg.Warning.unpred_async_tag_check_fail_on_sve_speculative_load_do_update_ffr

Unpredictable set to FALSE of the FFR element associated to SVE speculative load with an async tag check fail TAGS **UNPREDICTABLE**.

ArchMsg.Warning.unpredictable_a32_breakpoint

TAGS **UNPREDICTABLE** DISPLAY Unpredictable breakpoint on a A32 instruction at %{ADDR1} Offset %{OFFSET} BAS %{BAS} for a %{IS_ADDRESS_MISMATCH_BREAKPOINT:(Mismatch|Match)} Breakpoint number %{N:d} Breakpoint address %{ADDR2}. Fields:

ADDR1 unsigned int
Address.

ADDR2 unsigned int
Breakpoint address.

BAS unsigned int
BAS field.

IS_ADDRESS_MISMATCH_BREAKPOINT bool
Is a Mismatch breakpoint, not a Match breakpoint.

N unsigned int

Breakpoint number.

OFFSET unsigned int

Always has a value of zero.

ArchMsg.Warning.unpredictable_banked_register_access

TAGS **UNPREDICTABLE** DISPLAY Access to banked register (R=%{R:d} SYSm=%{SYSm:2x}) is unpredictable. Fields:

R unsigned int

R bit.

SYSm unsigned int

SYSm.

ArchMsg.Warning.unpredictable_ccfail_undef

TAGS **UNPREDICTABLE** DISPLAY An instruction %{OPCODE} at VA=%{VADDR} failed its condition codes check but would otherwise have caused a non-data-dependent trap or undefined exception. Behaviour is implementation-defined choice of exception or nop. Fields:

OPCODE unsigned int

opcode of the instruction.

VADDR unsigned int

Virtual address of the instruction being executed.

ArchMsg.Warning.unpredictable_ici_bits

TAGS **UNPREDICTABLE** DISPLAY ICI bits do not apply or are architecturally unpredictable for instruction.

ArchMsg.Warning.unpredictable_t32_breakpoint

TAGS **UNPREDICTABLE** DISPLAY Unpredictable breakpoint on a T32 instruction at %{ADDR1} Offset %{OFFSET} BAS %{BAS} for a %{IS_ADDRESS_MISMATCH_BREAKPOINT:(Mismatch|Match)} Breakpoint number %{N:d} Breakpoint address %{ADDR2}. Fields:

ADDR1 unsigned int

Address.

ADDR2 unsigned int

Breakpoint address.

BAS unsigned int

BAS field.

IS_ADDRESS_MISMATCH_BREAKPOINT bool

Is a Mismatch breakpoint, not a Match breakpoint.

N unsigned int

Breakpoint number.

OFFSET unsigned int

Indicates which halfword of a 32-bit Thumb instruction the breakpoint is set on. 0 for the first halfword or if the instruction is 16 bits wide, or 1 for the second halfword.

ArchMsg.Warning.unpredictable_t32_vector_catch

TAGS **UNPREDICTABLE** DISPLAY Unpredictable Vector catch exception on a T32 instruction at **%{ADDR}** Offset **%{OFFSET}**. Fields:

ADDR unsigned int

Address.

OFFSET unsigned int

Indicates which halfword of a 32-bit Thumb instruction the vector catch exception is set on. 0 for the first halfword or if the instruction is 16 bits wide, or 1 for the second halfword.

ArchMsg.Warning.unpredictable_watchpoint_far

TAGS **UNPREDICTABLE** DISPLAY The value for FAR/EDWAR for hitting this watchpoint may be anywhere in the range **%{LOWER_BOUND}** to **%{UPPER_BOUND}**. Fields:

LOWER_BOUND unsigned int

the lowest address accessed by the instruction that triggered the watchpoint.

UPPER_BOUND unsigned int

the highest watchpointed address accessed by that instruction.

ArchMsg.Warning.warning_AdvSIMDExpandImmUnexpectedZero

TAGS **UNPREDICTABLE** DISPLAY AdvSIMDExpandImm may treat this immediate value as **UNPREDICTABLE**.

ArchMsg.Warning.warning_ConditionalSMC

TAGS **UNPREDICTABLE** DISPLAY SMC instruction has **UNPREDICTABLE** effects when conditional and when combined with traps.

ArchMsg.Warning.warning_access_crosses_page_boundary

TAGS **UNPREDICTABLE** DISPLAY Access crosses page boundary.

ArchMsg.Warning.warning_access_crosses_page_boundary_has_fault

TAGS **UNPREDICTABLE** DISPLAY Access that crosses page boundary has fault(s).

ArchMsg.Warning.warning_access_crosses_page_boundary_has_fault_on_both_pages

TAGS **UNPREDICTABLE** DISPLAY Access that crosses page boundary has a fault on each page. Fields:

ADDR1 unsigned int

FAR address on first page.

ADDR2 unsigned int

FAR address on second page.

FSTATUS1 unsigned int

Fault status on first page.

FSTATUS2 unsigned int

Fault status on second page.

ArchMsg.Warning.warning_access_crosses_page_boundary_spanning_different_memory

TAGS **UNPREDICTABLE** DISPLAY Access at address %{ADDR} crosses page from MEM_TYPE %{MEMTYPE_PAGE1} to %{MEMTYPE_PAGE2}. Fields:

ADDR unsigned int

address of access crossing page.

MEMTYPE_PAGE1 signed int

First Page's memory type (0-Normal, 1-Non-Normal).

MEMTYPE_PAGE2 signed int

Second Page's memory type (0-Normal, 1-Non-Normal).

ArchMsg.Warning.warning_access_wraps_around_memory

TAGS **UNPREDICTABLE** DISPLAY Access wraps around memory in %{WIDTH:d}bit mode. Fields:

WIDTH unsigned int

Address width.

ArchMsg.Warning.warning_bcr_linking_status

TAGS **UNPREDICTABLE** DISPLAY DBGBCR.{BT2,BT} combination gives unpredictable behaviour.

ArchMsg.Warning.warning_bcr_mask_reserved

TAGS **UNPREDICTABLE** DISPLAY DBGBCR_ELn.MASK is set to a reserved value.

ArchMsg.Warning.warning_bvr_masked_bit_not_zero

TAGS **UNPREDICTABLE** DISPLAY DBGBVRn_EL1.VA contains a bit masked by DBGBCRn_EL1.MASK that is not zero.

ArchMsg.Warning.warning_ccsidr_unimplemented_level

TAGS **UNPREDICTABLE** DISPLAY CCSIDR read while CSSELR %{CSSELR:x} points to an unimplemented cache level. Fields:

CSSELR unsigned int

current effective value of CSSELR.

ArchMsg.Warning.warning_change_to_ns_when_tge_set

TAGS **UNPREDICTABLE** DISPLAY Attempting to change NS to 1 when HCR.TGE = 1.

ArchMsg.Warning.warning_contiguous_bit_check_abort

TAGS **UNPREDICTABLE** DISPLAY An abort occurred whilst attempting to check TLB entry is contiguous at %{ENTRY_ADDR}. Fields:

ENTRY_ADDR unsigned int

address of conflicting TLB entry.

ArchMsg.Warning.warning_contiguous_bit_error

TAGS **UNPREDICTABLE** DISPLAY TLB entry at %{ENTRY_ADDR} was not contiguous with entry at %{CONTIG_ADDR} Contents are %{ENTRY_DATA} but expected %{CONTIG_DATA}. Fields:

CONTIG_ADDR unsigned int

address of the TLB first read, with which this entry is expected to be contiguous.

CONTIG_DATA unsigned int

expected contents based on the entry at CONTIG_ADDR.

ENTRY_ADDR unsigned int

address of conflicting TLB entry.

ENTRY_DATA unsigned int

contents of conflicting TLB entry.

ArchMsg.Warning.warning_contiguous_bit_error_D128

TAGS **UNPREDICTABLE** DISPLAY TLB entry at %{ENTRY_ADDR} was not contiguous with entry at %{CONTIG_ADDR} Contents are %{ENTRY_DATA_HI} %{ENTRY_DATA_LO} but expected %{CONTIG_DATA}. Fields:

CONTIG_ADDR unsigned int

address of the TLB first read, with which this entry is expected to be contiguous.

CONTIG_DATA unsigned int

expected contents based on the entry at CONTIG_ADDR.

ENTRY_ADDR unsigned int

address of conflicting TLB entry.

ENTRY_DATA_HI unsigned int

Higher bit contents of conflicting TLB entry.

ENTRY_DATA_LO unsigned int

Lower bit contents of conflicting TLB entry.

ArchMsg.Warning.warning_cp10_cp11_mismatch

TAGS **UNPREDICTABLE** DISPLAY Unpredictable configuration of CP10 and CP11 controls at EL %{EL:d}. Fields:

EL unsigned int

EL owning the control.

ArchMsg.Warning.warning_cp10_cp11_reserved_value

TAGS **UNPREDICTABLE** DISPLAY Unpredictable configuration of CPACR CP10 (%{CP10}) or CP11 (%{CP11}). Fields:

CP10 unsigned int

CP10 access.

CP11 unsigned int

CP11 access.

ArchMsg.Warning.warning_csselr_level_out_of_range

TAGS **UNPREDICTABLE** DISPLAY CSSELR.Level out of range, written value %{LEVEL} with only %{IMPLEMENTED:d} levels implemented. Fields:

IMPLEMENTED unsigned int

number of cache levels implemented.

LEVEL unsigned int

written value of CSSELR.Level.

TYPE enum

The type of cache selected by CSSELR.InD and TnD.

ArchMsg.Warning.warning_debug_flow_control_bits_not_obeyed

TAGS **UNPREDICTABLE** DISPLAY Flow control bits for %{REGISTER} were not obeyed, causing %{ERROR}. Fields:

ERROR enum

type of error resulting.

REGISTER enum

register under consideration.

ArchMsg.Warning.warning_debug_register_access_during_reset

TAGS **UNPREDICTABLE** DISPLAY Unpredictable %{IS_WRITE:(read|write)} access to debug register offset %{OFFSET} during reset. Fields:

IS_WRITE unsigned int

Write Not Read.

OFFSET unsigned int

Register Offset.

ArchMsg.Warning.warning_decode_cps_inconsistent_fields

TAGS **UNPREDICTABLE** DISPLAY The mode setting fields in the CPS instruction are inconsistent.

ArchMsg.Warning.warning_decode_invalid_state

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable in current exception-level/security state.

ArchMsg.Warning.warning_default_cacheable_mmu_on

TAGS **UNPREDICTABLE** DISPLAY Default cacheable is enabled (HCR.DC = 1) while MMU is enabled.

ArchMsg.Warning.warning_default_cacheable_vmmu_off

TAGS **UNPREDICTABLE** DISPLAY Default cacheable is enabled (HCR.DC = 1) while HCR.VM = 0.

ArchMsg.Warning.warning_deprecated_wvr_bit_2_set

TAGS **UNPREDICTABLE** DISPLAY DBGWVR%{N:d}_EL1=%{WVR:h} has bit 2 set, deprecated in ARMv8. Fields:

N unsigned int
index.

WVR unsigned int
DBGWVR.

ArchMsg.Warning.warning_eret_while_software_step_active_pending

Missing ISB between setting MDSCR_EL1.SS and ERET TAGS UNPREDICTABLE.

ArchMsg.Warning.warning_exclusive_to_non_normal

TAGS UNPREDICTABLE DISPLAY Exclusive Access to Non-Normal Memory %{ADDR}. Fields:

ADDR unsigned int
Address.

ArchMsg.Warning.warning_exclusive_to_non_writeback

TAGS UNPREDICTABLE DISPLAY Exclusive Access to Non-Writeback-Cacheable Memory %{ADDR}. Fields:

ADDR unsigned int
Address.

ArchMsg.Warning.warning_execute_from_device_memory

TAGS UNPREDICTABLE DISPLAY An attempt was made to execute from Device memory.

ArchMsg.Warning.warning_illegal_cpsr_mode

TAGS UNPREDICTABLE DISPLAY Writing an illegal or unimplement mode (%{NEW_MODE:h}) to CPSR was unpredictable. Fields:

NEW_MODE unsigned int
New value of CPSR.M.

ArchMsg.Warning.warning_illegal_srs_mode

TAGS UNPREDICTABLE DISPLAY Illegal or UNPREDICTABLE mode (%{MODE:h}) used for SRS instruction. Fields:

MODE unsigned int
mode for Banked SP.

ArchMsg.Warning.warning_implementation_defined_read_debug_register_in_SCS

Read Access to Debug registers from the processor is IMPLEMENTATION DEFINED TAGS IMP_DEF. Fields:

PPB_OFFSET unsigned int
debug register offset.

ArchMsg.Warning.warning_implementation_defined_sequential_security_transitions_supported

The behavior of sequential instruction fetches that cross from non-secure to secure memory and contain an SG instruction is CONSTRAINED UNPREDICTABLE TAGS UNPREDICTABLE.

ArchMsg.Warning.warning_implementation_defined_stack_limit_check_supported

It is **IMPLEMENTATION DEFINED** whether stack pointer limit checking is performed for this instruction TAGS IMP_DEF. Fields:

ADDRESS unsigned int

Address of instruction.

OPCODE unsigned int

instruction opcode.

ArchMsg.Warning.warning_implementation_defined_taggednonsharedmemorytype

TAGS IMP_DEF DISPLAY Memory attribute has Tagged, non shared memory type; it is **IMPLEMENTATION DEFINED** whether the memory location is treated as Tagged or non-Tagged.

ArchMsg.Warning.warning_implementation_defined_write_debug_register_in_SCS

Write Access to Debug registers from the processor is **IMPLEMENTATION DEFINED** TAGS IMP_DEF. Fields:

DATA unsigned int

data attempted to be written.

PPB_OFFSET unsigned int

debug register offset.

ArchMsg.Warning.warning_invalid_tcr_granule

TAGS **UNPREDICTABLE** DISPLAY TCR.TG{%TG_ID:(0|1)} bits have been set to a granule size {%REQUEST:d}K not implemented - using {%SUBSTITUTE:d}K. Fields:

REQUEST signed int

page size requested (or 0 for reserved).

SUBSTITUTE signed int

best guess available page size.

TG_ID bool

bits TG0 or TG1.

ArchMsg.Warning.warning_load_multiple_user_registers_from_user_mode

TAGS **UNPREDICTABLE** DISPLAY An LDM(user registers) instruction executed from user mode.

ArchMsg.Warning.warning_mte_in_mem_attr_not_supported

TAGS **UNPREDICTABLE** DISPLAY Memory attribute {%ATTR} has Tagged memory type but Armv8.5-MTE is not implemented. Fields:

ATTR unsigned int

Memory attribute.

ArchMsg.Warning.warning_reserved_breakpoint_state_match

TAGS **UNPREDICTABLE** DISPLAY {%IS_BREAKPOINT:(Watchpoint|Breakpoint)} programmed with a reserved combination of HMC, SSC and {%IS_BREAKPOINT:(PAC|PMC)}. Fields:

IS_BREAKPOINT unsigned int

Is a breakpoint, not a watchpoint.

ArchMsg.Warning.warning_shareability

TAGS **UNPREDICTABLE** DISPLAY Unpredictable: combination of the 1st and 2nd stages of translation is Normal Inner Non-Cacheable, Outer Non-Cacheable.

ArchMsg.Warning.warning_software_step_set_while_enabled

TAGS **UNPREDICTABLE** DISPLAY MDSCR_EL1.SS set to 1 while software step debug exceptions are enabled.

ArchMsg.Warning.warning_thumb_instruction_wraps_around_memory

TAGS **UNPREDICTABLE** DISPLAY Thumb instruction wraps around memory.

ArchMsg.Warning.warning_ttbr_sbz_bits_are_not_zero

TAGS **UNPREDICTABLE** DISPLAY %{TTBR}[%{MSB:d}:0] should be zero, but are not. Fields:

MSB unsigned int

x-1.

TTBR enum

which TTBR register.

ArchMsg.Warning.warning_unaligned_address_dbgdtrrx_write

TAGS **UNPREDICTABLE** DISPLAY Unaligned address %{ADDR} in X0/R0 when DBGDTRRX written in memory access mode. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unaligned_address_dbgdtrtx_read

TAGS **UNPREDICTABLE** DISPLAY Unaligned address %{ADDR} in X0/R0 when DBGDTRTX read in memory access mode. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unaligned_to_device

TAGS **UNPREDICTABLE** DISPLAY Unaligned Access to Device Memory %{ADDR}. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unaligned_to_strongly_ordered

TAGS **UNPREDICTABLE** DISPLAY Unaligned Access to Strongly Ordered Memory %{ADDR}. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unknown_sau_rnr

TAGS UNPREDICTABLE DISPLAY SAU_RNR was set to an unsupported value.

ArchMsg.Warning.warning_unpredictable_AIRCR_PRIS_and_BFHFNMINs

TAGS UNPREDICTABLE DISPLAY The effect of setting both AIRCR.BFHFNMINs and AIRCR.PRIS to 1 is UNPREDICTABLE.

ArchMsg.Warning.warning_unpredictable_AIRCR_VECTCLRACTIVE_when_not_in_debug

TAGS UNPREDICTABLE DISPLAY The effect of writing a 1 to AIRCR.VECTCLRACTIVE if the processor is not halted in Debug state is UNPREDICTABLE.

ArchMsg.Warning.warning_unpredictable_AIRCR_VECTRESET_and_SYSRESETREQ

TAGS UNPREDICTABLE DISPLAY When the processor is halted in Debug state, if a write to the register writes a 1 to both VECTRESET and SYSRESETREQ, the behavior is UNPREDICTABLE.

ArchMsg.Warning.warning_unpredictable_AIRCR_VECTRESET_when_not_in_debug

TAGS UNPREDICTABLE DISPLAY The effect of writing a 1 to AIRCR.VECTRESET if the processor is not halted in Debug state is UNPREDICTABLE.

ArchMsg.Warning.warning_unpredictable_AIRCR_incorrect_VKEY

TAGS UNPREDICTABLE DISPLAY The value 0x05FA must be written to AIRCR.VKEY, otherwise the register write is UNPREDICTABLE.

ArchMsg.Warning.warning_unpredictable_EXC_RETURN_Reserved_Bit

EXC_RETURN[23:7] are reserved with the special condition that all bits should be written as one. Values other than all 1s are UNPREDICTABLE. TAGS UNPREDICTABLE. Fields:

EXC_RETURN unsigned int

Special PC value.

SBO_MASK unsigned int

Which bits Should Be One to not UNPRED.

ArchMsg.Warning.warning_unpredictable_PMEVTYPEPn_ELO_VS_reserved_value

TAGS UNPREDICTABLE DISPLAY Unpredictable reserved value of PMEVTYPEn<n>_ELO.VS.

ArchMsg.Warning.warning_unpredictable_change_to_DEMCR_MON_STEP_at_insufficient_priority

TAGS UNPREDICTABLE DISPLAY The effect of changing DEMCR.MON_STEP at an execution priority that is lower than the priority of the DebugMonitor exception is **CONSTRAINED** UNPREDICTABLE.

ArchMsg.Warning.warning_unpredictable_change_to_priority_of_active_exception

TAGS UNPREDICTABLE DISPLAY Changing the priority of an active exception is UNPREDICTABLE.

ArchMsg.Warning.warning_unpredictable_clear_lspact

TAGS UNPREDICTABLE DISPLAY UNPREDICTABLE state on context switch. Fields:

DRVEXC_TAKEN unsigned int

Derived exception taken, not pended.

REACHED_LAST_STORE unsigned int

Only the last faulted.

ArchMsg.Warning.warning_unpredictable_dbg_exit_unaligned_dlr

TAGS **UNPREDICTABLE** DISPLAY Debug exit to AArch32 with DLR[0] set to 1. PC[0] can be set to 0 or to DLR[0].

ArchMsg.Warning.warning_unpredictable_exception_catch

TAGS **UNPREDICTABLE** DISPLAY Generation of an exception catch event is unpredictable.

ArchMsg.Warning.warning_unpredictable_exception_return_inconsistent_state

TAGS **UNPREDICTABLE** DISPLAY state on exception return is unpredictable.

ArchMsg.Warning.warning_unpredictable_exception_return_instruction

TAGS **UNPREDICTABLE** DISPLAY exception return instruction is unpredictable.

ArchMsg.Warning.warning_unpredictable_in_debug_state

TAGS **UNPREDICTABLE** DISPLAY Instruction is **UNPREDICTABLE** when executed in debug state.

ArchMsg.Warning.warning_unpredictable_mair_encoding

TAGS **UNPREDICTABLE** DISPLAY Memory attribute %{ATTR} has invalid MAIR encoding; this may lead to unpredictable results. Fields:

ATTR unsigned int

Memory attribute.

ArchMsg.Warning.warning_unpredictable_pmu_counter_access

TAGS **UNPREDICTABLE** DISPLAY Access to PMU counters from non-secure EL0 or EL1 is **UNPREDICTABLE** with MDCR_EL2.HPMN set to 0.

ArchMsg.Warning.warning_unpredictable_prioritization_breakpoint_match_vector_catch

TAGS **UNPREDICTABLE** DISPLAY Two events with the same priority occurred at the same instruction: Address Matching Vector Catch debug event, and Breakpoint debug event. It is constrained unpredictable which is taken.

ArchMsg.Warning.warning_unpredictable_stack_selection

TAGS **UNPREDICTABLE** DISPLAY **UNPREDICTABLE** state on context switch.

ArchMsg.Warning.warning_unpredictable_tsize_out_of_range

TAGS **UNPREDICTABLE** DISPLAY %{TCR}.T%{TTBR:d}SZ (%{TSIZE:d} bits) is out of range. Must be between 25 and %{TSIZE_MAX:d}. Fields:

TCR enum

Which TCR.

TSIZE signed int

Value of TSize.

TSIZE_MAX signed int

The maximum allowed value.

TTBR unsigned int

Which TTBR region.

ArchMsg.Warning.warning_unpredictable_unaligned_pc_as_base_register

TAGS **UNPREDICTABLE** DISPLAY The PC value must be word-aligned, otherwise the behavior of the instruction is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_unaligned_pop_stack

TAGS **UNPREDICTABLE** DISPLAY **CONSTRAINED UNPREDICTABLE** unaligned stack frame %{FRAME_ADDR:x} in exception-return. Fields:

FRAME_ADDR unsigned int

unaligned stack frame address.

ArchMsg.Warning.warning_unpredictable_unaligned_to_non_writeback

TAGS **UNPREDICTABLE** DISPLAY Unaligned Access to Non-Writeback-Cacheable Memory.

ArchMsg.Warning.warning_unpredictable_vmsa_memattrib

TAGS **UNPREDICTABLE** DISPLAY Unpredictable vmsa memory attribute with texcb value programmed to %{TEXCB} when tex remap is %{TEX_REMAP}. Fields:

TEXCB unsigned int

texcb value.

TEX_REMAP bool

use tex remap.

ArchMsg.Warning.warning_unpredictable_vtcr_t0sz_sl0

TAGS **UNPREDICTABLE** DISPLAY The combination of VTCR.T0SZ=%{T0SZ:d} and VTCR.SL0=%{SL0:d} is unpredictable. Fields:

SL0 signed int

VTCR.SL0, starting level for stage 2 translation table walks.

T0SZ signed int

VTCR.T0SZ, required input address range.

ArchMsg.Warning.warning_unpredictable_write_DHCSR

TAGS **UNPREDICTABLE** DISPLAY **UNPREDICTABLE** set of bit-changes in DHCSR. Fields:

NEW unsigned int

data attempted to be written.

OLD unsigned int

previous value.

ArchMsg.Warning.warning_unpredictable_write_SHCSR

TAGS **UNPREDICTABLE** DISPLAY **UNPREDICTABLE** setting a pending bit to 1 for an exception with priority >= execution priority.

ArchMsg.Warning.warning_unsupported_access_to_memory_mapped_register

TAGS **UNPREDICTABLE** DISPLAY Access to memory mapped register is unsupported.

ArchMsg.Warning.warning_user_jmcr_access

TAGS UNPREDICTABLE DISPLAY User %{WRITE:(write|read)} access to JMCR is unpredictable.
Fields:

WRITE unsigned int
access is write.

ArchMsg.Warning.warning_wcr_mask_and_bas

TAGS UNPREDICTABLE DISPLAY DBGWCR_EL1.MASK is non-zero and BAS is not 0xff.

ArchMsg.Warning.warning_wcr_mask_reserved

TAGS UNPREDICTABLE DISPLAY DBGWCR_ELn.MASK is set to a reserved value.

ArchMsg.Warning.warning_wcr_non_configuous_bas

TAGS UNPREDICTABLE DISPLAY DBGWCR_EL1.BAS has non-contiguous set of ones.

ArchMsg.Warning.warning_wvr_masked_bit_not_zero

TAGS UNPREDICTABLE DISPLAY DBGWVRn_EL1.VA contains a bit masked by
DBGWCRn_EL1.MASK that is not zero.

ArchMsg.warning_non_architectural_beat_order

DISPLAY Current set of committed beats does not give legal ECI value.

ArchMsg.warning_unpred_system_register_access

DISPLAY Unpredictable %{IS_WRITE:(read|write)} access to system register offset
%{OFFSET}. Fields:

IS_WRITE unsigned int
Write Not Read.

OFFSET unsigned int
Register Offset.

BEAT_EXCEPTION

Exception during of one beat of an M. Fields:

BEAT unsigned int
Beat number 0..3.

EXC unsigned int
Internal representation of the excep.

EXCEPTION_PC unsigned int
Address of older in-flight instructi.

INST unsigned int
This instruction as FirstHalfWord:Se.

INST_ADDR unsigned int
Address of faulting instruction inst.

BEAT_START

Start of one beat of an MVE Vector i. Fields:

BEAT unsigned int

Beat number 0..3.

DISASS string

Disassembly of instruction.

ELMT_MASK unsigned int

Element Mask.

EXCEPTION_PC unsigned int

Address of older in-flight instructi.

INST unsigned int

This instruction as FirstHalfWord:Se.

INST_ADDR unsigned int

Address of this instruction.

LTP_MASK unsigned int

Element Mask from Loop Tail Predicat.

VPT_MASK unsigned int

Element Mask from VPR.

BRANCH_MISPREDICT

Simulating branch mispredict. Fields:

PC unsigned int

Origin address (or 0 if unavailable).

BRA_DIR

Direct branches, to immediate address. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISSET enum

The instruction set of the branch instruction.

IS_COND bool

Indicates if this is a conditional branch.

IS_HINTED bool

Whether the conditional instruction is hinted.

IS_ISB bool

Whether the instruction is ISB.

IS_LINK bool

Indicates if this is a branch with link.

PC unsigned int

The address of the branch instruction.

TARGET_ISET enum

The instruction set after the branch.

TARGET_PC unsigned int

The address the instruction branches to.

BRA_INDIR

Indirect branches, perhaps to a register. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISET enum

The instruction set of the branch instruction.

IS_COND bool

Indicates if this is a conditional branch.

IS_HINTED bool

Whether the conditional instruction is hinted.

IS_ISB bool

Whether the instruction is ISB.

IS_LINK bool

Indicates if this is a branch with link.

PC unsigned int

The address of the branch instruction.

TARGET_ISET enum

The instruction set after the branch.

TARGET_PC unsigned int

The address the instruction branches to.

BRA_INDIR_CCFAIL

Indirect branches fail, perhaps to a register.

BRA_LOB

Low Overhead branch. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISSET enum

The instruction set of the branch instruction.

IS_COND bool

Indicates if this is a conditional branch.

IS_HINTED bool

Whether the conditional instruction is hinted.

IS_ISB bool

Whether the instruction is ISB.

IS_LINK bool

Indicates if this is a branch with link.

PC unsigned int

The address of the branch instruction.

TARGET_ISET enum

The instruction set after the branch.

TARGET_PC unsigned int

The address the instruction branches to.

BRA_RET_CCFAIL

Branch return fail, perhaps to a register.

CACHE_MAINTENANCE_OP

Cache Maintenance Operation. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

DATA unsigned int

Specified MVA or Set/way.

FUNCTION enum

Clean / Invalidate.

SCOPE enum

Affected region.

SIDE enum

Inst / Data.

CCFAIL

Conditional instruction condition check fail. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CCFAIL_UNC

Unconditional instruction condition check fail. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CCPASS

Conditional instruction condition check pass. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CCPASS_UNC

Unconditional instruction condition check pass. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CODE_CACHE_MAINT

Code cache maintenance. Fields:

END_INCL unsigned int

Inclusive end address for VA or PA requests.

NS bool

Non-secure world for PA.

START unsigned int

Start address for VA or PA requests.

TYPE enum

Request type.

COMPILE_BLOCK_END

Last instruction of basic block translated. Fields:

VADDR unsigned int

Address of next instruction after this basic block.

COMPILE_BLOCK_START

First instruction of basic block translated. Fields:

VADDR unsigned int

Address of first instruction in basic block.

COMPILE_INST

ARM instruction compiled. Fields:

DISASS string

Disassembly of the instruction.

ISSET enum

The instruction set of this instruction.

ITSTATE unsigned int

The ITSTATE current for the instruction.

OPCODE unsigned int

The opcode of the instruction.

PC unsigned int

The address of the instruction.

SIZE unsigned int

The size of the instruction in bytes.

CONTEXTIDR

Context ID Register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

NS bool

Secure or nonsecure banked register is accessed.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The new value written.

CONTEXT_SYNC

Called for every context synchronization event. Fields:

ARCH_CSE bool

Architectural Context Synchronization Event.

PC unsigned int

The address of the synchronization instruction.

SYSREG_SYNC bool

System register synchronization event.

CORE_ENDIAN

Core BE8 Big-Endian state changed. Fields:

BE8 bool

Core BE8 Big-Endian state.

CORE_INFO

Static processor attributes. Only triggered by a call to DumpState(). Fields:

ARCH_PROFILE enum

The architecture profile of the core.

CLUSTER_ID unsigned int

The cluster ID of this processor.

CORE_NUM unsigned int

The number of this core in an MP processor.

DETAILED_CORE_FEATURES enum

Detailed core features as a bitfield. Enum values are bitmasks to extract the info from the field.

FPU_VERSION enum

The VFP version implemented by the core.

MEM_ARCH enum

The memory architecture of the core.

MIDR_EL1 unsigned int

MIDR_EL1 value.

NUM_CORES unsigned int

The number of cores in this MP processor.

QUANTUM_SIZE unsigned int

The default quantum size of the core (in ticks).

SECURITY_FEATURES bool

Does the core have security features?.

CORE_LOADS

Processor load accesses. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ACQREL enum

Is this an acquire/release.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DATA unsigned int

The data read or written.

ELEMENT_SIZE unsigned int

Width of each element.

ENDIANNESS enum

Byte ordering of this access.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

LOCK enum

Normal, exclusive or locked access.

MECID unsigned int

Memory Encryption Context ID, if applicable.

MECID2 unsigned int

Memory Encryption Context ID of the second page, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PAGE_SIZE unsigned int

Page size as $\log_2(\text{size in bytes})$.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK, 2=Exclusive Failed, 3=Unsupported.

SIZE unsigned int

Width of the access in bytes. Only required if DATA is not traced.

TRANS bool

Is this a translated access.

VADDR unsigned int

The virtual address of the access.

CORE_REGS

Changes of the core registers R0 to R14. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number, 0 to 14.

OLD_VALUE unsigned int

The old value overwritten.

PHYS_ID enum

The physical register accessed.

VALUE unsigned int

The new value written to the register.

CORE_STALLED_TRANSITION

Detect when a core is prevented from progressing PChannel states. Fields:

CURRENT_STATE enum

Current power mode of core.

REASON string

The reason a transition was stalled, and what must be done to unblock it.

TARGET_STATE enum

Desired power mode of core after transition.

CORE_STORES

Processor store accesses. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ACQREL enum

Is this an acquire/release.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DATA unsigned int

The data read or written.

ELEMENT_SIZE unsigned int

Width of each element.

ENDIANNESS enum

Byte ordering of this access.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

LOCK enum

Normal, exclusive or locked access.

MECID unsigned int

Memory Encryption Context ID, if applicable.

MECID2 unsigned int

Memory Encryption Context ID of the second page, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PAGE_SIZE unsigned int

Page size as $\log_2(\text{size in bytes})$.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK, 2=Exclusive Failed, 3=Unsupported.

SIZE unsigned int

Width of the access in bytes. Only required if DATA is not traced.

TRANS bool

Is this a translated access.

VADDR unsigned int

The virtual address of the access.

CRYPTO_SPEC

Every crypto instruction speculatively executed.

DATA_CACHE_SET_ALLOC_TAGS

Set allocation tags from DC GVA instruction. Fields:

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

NSDESC unsigned int

The security state of the access.

PADDR unsigned int

The physical address of the access.

PAS enum

The physical address space of the page.

RESPONSE enum

0=Aborted, 1=OK.

SIZE unsigned int

The size of memory in bytes.

TAG unsigned int

The value of the tag being set.

TAGGING_TYPE enum

Type of Memory tagging (Physical or Virtual).

TAG_VADDR unsigned int

The virtual address of the Virtual Tag if Virtual tagging is used; otherwise 0.

VADDR unsigned int

Virtual address used for setting allocation tags.

DATA_CACHE_ZERO

Zero and invalidate cache line from DC ZVA instruction. Fields:

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

NSDESC unsigned int

The security state of the access.

PADDR unsigned int

The physical address of the access.

PADDR_TAG unsigned int

If tag checked, the physical address tag for the access.

PAS enum

The physical address space of the page.

RESPONSE enum

0=Aborted, 1=OK.

SIZE unsigned int

The size of zero'd memory in bytes.

TAG_CHECKED bool

Is this access tag checked.

VADDR unsigned int

Virtual address of the zero'd cache line.

DEBUG_EVENT

Hardware debug support event. Fields:

EVENT enum

Description of event.

VALUE unsigned int

data value.

DMI_ALLOCATE

DMI allocated. Fields:

ACCESS_DIRECTION enum

READ / WRITE.

PADDR unsigned int

The physical address of the DMI.

PTR unsigned int

DMI range pointer.

SIDE enum

DSIDE / ISIDE.

SIZE unsigned int

The size of DMI.

VADDR unsigned int

The virtual address of the DMI..

DMI_HIT

DMI hit, access used DMI (only fire for small DMI regions i.e. less than a full page). Fields:

ACCESS_DIRECTION enum

READ / WRITE.

PADDR unsigned int

The physical address of the DMI.

SIDE enum

DSIDE / ISIDE.

SIZE unsigned int

The size of the DMI.

VADDR unsigned int

The virtual address of the DMI..

DMI_REVOKE

DMI revoked. Fields:

ACCESS_DIRECTION enum

READ / WRITE.

PADDR unsigned int

The physical address of the DMI.

SIDE enum

DSIDE / ISIDE.

SIZE unsigned int

The size of DMI.

VADDR unsigned int

The virtual address of the DMI..

DTCM_ACCESS

Trace TCM accesses.

DTCM_ACCESS_RAZ_WI

Trace TCM accesses. Fields:

ADDR unsigned int

Address of the **RAZ/WI** access.

DWT_MATCH

DWT comparator matches. Fields:

NUM unsigned int

DWT comparator number.

TYPE enum

DWT comparator configuration.

END_COMPILE

Compilation end. Fields:

END_OF_PAGE_COUNT unsigned int

Number of basic blocks exited due to page boundary since **START_COMPILE**.

FETCHFAIL_COUNT unsigned int

Number of basic blocks exited due to ifetch failure **START_COMPILE**.

INST_COUNT unsigned int

Number of instructions compiled since **START_COMPILE**.

NONSEQ_COUNT unsigned int

Number of basic blocks exited due to non-sequential instructions since **START_COMPILE**.

PAGE_STRADDLE_COUNT unsigned int

Number of basic blocks exited due to unaligned instructions crossing page since **START_COMPILE**.

EXCEPTION

Exceptions that are taken. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ISSET enum

The instruction set of the processor when the exception occurred.

LR unsigned int

The value assigned to the link register.

NS enum

The core's non-secure bit.

PC unsigned int

The location where the exception occurred.

TARGET_ISSET enum

The instruction set of the exception handler code.

TARGET_PC unsigned int

The address the exception branches to.

VECTOR enum

The exception vector.

EXCEPTION_END

Every exception completed. Fields:

NS enum

The core's non-secure bit.

EXCEPTION_ENTRY

Event marking the entry of an exception. All pushes of registers to the stack will follow this event. Fields:

PC unsigned int

The location where the exception occurred.

VECTOR enum

The exception that occurred.

EXCEPTION_RAISE

Every exception raised.

EXCEPTION_RETURN

Marks the end of an exception, but on an M core does not indicate a program flow change (branch). In most situations it will be followed by an indirect branch. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISSET enum

The instruction set of the branch instruction.

PC unsigned int

The address of the branch instruction.

EXCEPTION_RETURN_PREBRANCH

Exception return event that must occur before PC and processor state have updated.

EXCEPTION_START

Every exception started. Fields:

NS enum

The core's non-secure bit.

EXCEPTION_VECTOR_FETCH

I-side vector fetch for M-class exception. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

DATA unsigned int

Result of fetch if OK.

RESPONSE enum

0=Aborted, 1=OK.

VADDR unsigned int

Where the vector is fetched from (including VTOR).

FP_STATE

Floating point state. Fields:

FPSR unsigned int

FPSR register value.

MASK unsigned int

The mask indicating updated cumulative bits.

FREQ_CHANGED

Core frequency changed. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The instruction count of this core.

NEW_FREQ signed int

The new frequency of this core (expressed in Hz).

OLD_FREQ signed int

The old frequency of this core (expressed in Hz).

HLT

HLT instruction occurred. Fields:

IMM16 unsigned int

The 16-bit immediate value encoded in HLT instruction.

INFO_EXCEPTION_REASON

Provide information of real cause of exception. Fields:

FaultCause enum

Which bit of which V7M/V8M-mainline FSR/HFSR would have been set.

PC unsigned int

The location where the exception occurred.

PHASE enum

What we are doing with the exception.

REASONS enum

Why we are doing it.

SecurityState enum

Privacy State of exception.

VECTOR enum

The exception that occurred.

INFO_FP_CONTEXT

Inform when ExecuteFPCheck writes CONTROL.FPCA and FPSCR etc. Fields:

WHAT enum

What we are doing.

INFO_STACKING

Provide information of real cause of exception. Fields:

PHASE enum

Are we starting or ending a sequence of stack writes.

REASONS enum

Why we are doing it.

INST

Every instruction executed. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DEBUG_STATE bool

The instruction is executed in debug state.

DISASS string

Disassembly of instruction.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISET enum

The current instruction set.

ITSTATE unsigned int

The current ITSTATE.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

MODE enum

The mode the core is in.

NS unsigned int

The core's non-secure bit.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

OPCODE unsigned int

The opcode of the instruction.

PADDR unsigned int

The physical address of the instruction.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the instruction.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the page.

PC unsigned int

The address of the instruction.

SECURITY_STATE enum

The core's security state.

SIZE unsigned int

The size of the instruction in bytes.

INST_END

Every instruction completed.

INST_START

Every instruction started. Fields:

DEBUG_STATE bool

The instruction is executed in debug state.

ISET enum

The current instruction set.

MODE enum

The mode the core is in.

NS enum

The current Secure State.

PC unsigned int

The address of the conditional instruction.

SECURITY_STATE enum

The current Security State.

INST_STRADDLE

Instruction straddles boundary. Fields:

DEBUG_STATE bool

The instruction is executed in debug state.

ISET enum

The current instruction set.

MODE enum

The mode the core is in.

NS enum

The current Secure State.

PC unsigned int

The address of the conditional instruction.

SECURITY_STATE enum

The current Security State.

ISB

ISB instruction occurred. Fields:

PC unsigned int

The address of the ISB instruction.

ITCM_ACCESS

Trace TCM accesses.

ITCM_ACCESS_RAZ_WI

Trace TCM accesses. Fields:

ADDR unsigned int

Address of the **RAZ/WI** access.

ITM

Instrumentation Trace Macrocell. Fields:

ITM_PACKET_TYPE enum

ITM and DWT packets type.

PACKET_HEADER unsigned int

ITM Packet Header.

PACKET_PAYLOAD unsigned int

ITM Packet Payload.

LOAD_MULTIPLE_REGS_LIST

List of destination registers (not source) of a multiple registers load instruction. Fields:

REG string

Register to add into the list.

LOCAL_MONITOR

Local monitor activity. Fields:

PADDR unsigned int

Local Monitor Address.

State enum

State of the monitor (Open/Exclusive).

LOCKUP_CYCLE

This event is triggered if the core enters or stays in the lock-up state. Fields:

CAUSE enum

What causes the lockup?.

LOCKUP_ENTRY

This event is triggered if the core enters the lock-up state. Fields:

PC unsigned int

The PC when the lockup state is entered.

LOCKUP_EXIT

This event is triggered if the core leaves the lock-up state.

LO_BRANCH_INSTR

Low Overhead Branch Instruction. Fields:

EFFECT enum

Did it set LO_BRANCH_INFO and/or branch.

INSTR enum

What sort of instruction (BF/..BFCSEL/LE/WLS/...).

MODE_CHANGE

Mode change. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

MODE enum

The new mode.

OLD_MODE enum

The old mode.

MPU_TRANS

Address translation information. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

EXEC_PERM enum

Execution Permission.

IDAU_REG_NUM enum

IDAU region-number or code.

INNERCACHE_RA bool

Is the inner cache allocate on read.

INNERCACHE_TYPE enum

Inner Caching scheme (NC/MB/WA).

INNERCACHE_WA bool

Is the inner cache allocate on write.

MEMTYPE enum

Memory type.

NSDESC enum

Security-attributes of the transaction.

OUTERCACHE_RA bool

Is the outer cache allocate on read.

OUTERCACHE_TYPE enum

Outer Caching scheme (NC/MB/WA).

OUTERCACHE_WA bool

Is the outer cache allocate on write.

PADDR unsigned int

Address of the access.

PAGESIZE unsigned int

Page size as log2(size).

PAS enum

PAS accessed is secure or nonsecure memory.

READ_PERM enum

Read Permission.

REG_NUM unsigned int

MPU register based region-number 0-15 (else 0xFF=none, 0xBB/0xDD=background/default).

SAU_REG_NUM enum

SAU region-number or code.

SAU_RGN_TYPE enum

Privacy Region Type.

SH enum

Shareability.

SIDE enum

Inst / Data.

WRITE_PERM enum

Write Permission.

PERIODIC

Called for every quantum. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The instruction count of this CPU.

PC unsigned int

The address of the next instruction to be executed on this CPU.

PMU_COUNTER_OVERFLOW

PMU counter overflow. Fields:

COUNTER_GROUP enum

Counter group.

COUNTER_RANGE enum

Counter range.

EVENT_ID unsigned int

Event Identifier, field is valid only for Event counter group.

INDEX unsigned int

Counter index (as selected by PMSELR), field is valid only for Event/Cycle counter groups.

INTERRUPT bool

Is interrupt enabled on overflow for this counter.

PMU_M_EVENT_AUT_RETIRED

PMU Event AUT_RETIRED.

PMU_M_EVENT_BF_CANCEL

PMU Event BF_CANCEL.

PMU_M_EVENT_BF_RETIRED

PMU Event BF_RETIRED.

PMU_M_EVENT_DWT_CMPMATCH0

PMU Event DWT_CMPMATCH0.

PMU_M_EVENT_DWT_CMPMATCH1

PMU Event DWT_CMPMATCH1.

PMU_M_EVENT_DWT_CMPMATCH2

PMU Event DWT_CMPMATCH2.

PMU_M_EVENT_DWT_CMPMATCH3

PMU Event DWT_CMPMATCH3.

PMU_M_EVENT_DWT_CMPMATCH4

PMU Event DWT_CMPMATCH4.

PMU_M_EVENT_DWT_CMPMATCH5

PMU Event DWT_CMPMATCH5.

PMU_M_EVENT_DWT_CMPMATCH6

PMU Event DWT_CMPMATCH6.

PMU_M_EVENT_DWT_CMPMATCH7

PMU Event DWT_CMPMATCH7.

PMU_M_EVENT_INST_SPEC

PMU Event INST_SPEC.

PMU_M_EVENT_LE_CANCEL

PMU Event LE_CANCEL.

PMU_M_EVENT_LE_RETIRED

PMU Event LE_RETIRED.

PMU_M_EVENT_MVE_FP_HP_RETIRED

PMU Event MVE_FP_HP_RETIRED.

PMU_M_EVENT_MVE_FP_HP_SPEC

PMU Event MVE_FP_HP_SPEC.

PMU_M_EVENT_MVE_FP_MAC_RETIRED

PMU Event MVE_FP_MAC_RETIRED.

PMU_M_EVENT_MVE_FP_MAC_SPEC

PMU Event MVE_FP_MAC_SPEC.

PMU_M_EVENT_MVE_FP_RETIRE

PMU Event MVE_FP_RETIRE.

PMU_M_EVENT_MVE_FP_SPEC

PMU Event MVE_FP_SPEC.

PMU_M_EVENT_MVE_FP_SP_RETIRE

PMU Event MVE_FP_SP_RETIRE.

PMU_M_EVENT_MVE_FP_SP_SPEC

PMU Event MVE_FP_SP_SPEC.

PMU_M_EVENT_MVE_INST_RETIRE

PMU Event MVE_INST_RETIRE.

PMU_M_EVENT_MVE_INST_SPEC

PMU Event MVE_INST_SPEC.

PMU_M_EVENT_MVE_INT_MAC_RETIRE

PMU Event MVE_INT_MAC_RETIRE.

PMU_M_EVENT_MVE_INT_MAC_SPEC

PMU Event MVE_INT_MAC_SPEC.

PMU_M_EVENT_MVE_INT_RETIRE

PMU Event MVE_INT_RETIRE.

PMU_M_EVENT_MVE_INT_SPEC

PMU Event MVE_INT_SPEC.

PMU_M_EVENT_MVE_LDST_MULTI_RETIRE

PMU Event MVE_LDST_MULTI_RETIRE.

PMU_M_EVENT_MVE_LDST_MULTI_SPEC

PMU Event MVE_LDST_MULTI_SPEC.

PMU_M_EVENT_MVE_LDST_RETIRE

PMU Event MVE_LDST_RETIRE.

PMU_M_EVENT_MVE_LDST_SPEC

PMU Event MVE_LDST_SPEC.

PMU_M_EVENT_MVE_LD_MULTI_RETIRE

PMU Event MVE_LD_MULTI_RETIRE.

PMU_M_EVENT_MVE_LD_MULTI_SPEC

PMU Event MVE_LD_MULTI_SPEC.

PMU_M_EVENT_MVE_LD_RETIRE

PMU Event MVE_LD_RETIRE.

PMU_M_EVENT_MVE_LD_SPEC

PMU Event MVE_LD_SPEC.

PMU_M_EVENT_MVE_PRED

PMU Event MVE_PRED.

PMU_M_EVENT_MVE_ST_MULTI_RETIRE

PMU Event MVE_ST_MULTI_RETIRE.

PMU_M_EVENT_MVE_ST_MULTI_SPEC

PMU Event MVE_ST_MULTI_SPEC.

PMU_M_EVENT_MVE_ST_RETIRE

PMU Event MVE_ST_RETIRE.

PMU_M_EVENT_MVE_ST_SPEC

PMU Event MVE_ST_SPEC.

PMU_M_EVENT_MVE_VREDUCE_FP_RETIRE

PMU Event MVE_VREDUCE_FP_RETIRE.

PMU_M_EVENT_MVE_VREDUCE_FP_SPEC

PMU Event MVE_VREDUCE_FP_SPEC.

PMU_M_EVENT_MVE_VREDUCE_INT_RETIRE

PMU Event MVE_VREDUCE_INT_RETIRE.

PMU_M_EVENT_MVE_VREDUCE_INT_SPEC

PMU Event MVE_VREDUCE_INT_SPEC.

PMU_M_EVENT_MVE_VREDUCE_RETIRE

PMU Event MVE_VREDUCE_RETIRE.

PMU_M_EVENT_MVE_VREDUCE_SPEC

PMU Event MVE_VREDUCE_SPEC.

PMU_M_EVENT_PAC_RETIRE

PMU Event PAC_RETIRE.

PMU_M_EVENT_SE_CALL_NS

PMU Event SE_CALL_NS.

PMU_M_EVENT_SE_CALL_S

PMU Event SE_CALL_S.

POTINDEX_UPDATE

Value of POT index for current pc changed. Fields:

NEW_POTINDEX unsigned int

New POTIndex.

OLD_POTINDEX unsigned int

Old POTIndex.

PPB_READ_ACCESS

Trace reads from the core to memory mapped registers in the PPB (private peripheral bus) address range. Fields:

ACCESS_SIZE unsigned int

Logarithm of the access size: 0=byte, 1=halfword, 2=word, 3=doubleword, ...

BURST_LEN unsigned int

The number of data transfers (beats) in this burst.

DATA unsigned int

The data transferred.

INNER_ATTRIBUTES unsigned int

The inner attributes: bit0=write allocate, bit1=read allocate, bit2=cachable, bit3=bufferable, bit4=shareable.

INST bool

Is an instruction fetch.

LOCK enum

Normal / Locked / Exclusive.

MANAGER_ID unsigned int

The AXI manager ID.

NS bool

Is a nonsecure access.

OUTER_ATTRIBUTES unsigned int

The outer attributes: bit0=write allocate, bit1=read allocate, bit2=cachable, bit3=bufferable, bit4=shareable.

PADDR unsigned int

Physical address of access.

PRIV bool

Is privileged access.

RESPONSE enum

Whether the transaction was successful, or an error occurred.

USER_FLAGS unsigned int

Core specific additional signals.

PPB_WRITE_ACCESS

Trace writes from the core to memory mapped registers in the PPB (private peripheral bus) address range. Fields:

ACCESS_SIZE unsigned int

Logarithm of the access size: 0=byte, 1=halfword, 2=word, 3=doubleword, ...

BURST_LEN unsigned int

The number of data transfers (beats) in this burst.

DATA unsigned int

The data transferred.

INNER_ATTRIBUTES unsigned int

The inner attributes: bit0=write allocate, bit1=read allocate, bit2=cachable, bit3=bufferable, bit4=shareable.

INST bool

Is an instruction fetch.

LOCK enum

Normal / Locked / Exclusive.

MANAGER_ID unsigned int

The AXI manager ID.

NS bool

Is a nonsecure access.

OUTER_ATTRIBUTES unsigned int

The outer attributes: bit0=write allocate, bit1=read allocate, bit2=cachable, bit3=bufferable, bit4=shareable.

PADDR unsigned int

Physical address of access.

PRIV bool

Is privileged access.

RESPONSE enum

Whether the transaction was successful, or an error occurred.

USER_FLAGS unsigned int

Core specific additional signals.

PRELOAD_DATA

Data preload from PLD instruction. Fields:

VADDR unsigned int

Virtual address of the data that should be preloaded.

PRELOAD_INST

Instruction preload from PLI instruction. Fields:

VADDR unsigned int

Virtual address of the instruction that should be preloaded.

PRE_CORE_LOAD

Trace just before a core load. Fields:

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

PRE_CORE_STORE

Trace just before a core store. Fields:

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

PSTATE_PACTIVE_OPMODE_TRANSITION

Trace pactive and Opmode mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Operating mode after the transition.

OLD_STATE enum

Operating mode before the transition.

PSTATE_PACTIVE_OPMODE_TRANSITION_DENIED

Trace the denied pactive and Opmode mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Operating mode after the transition.

OLD_STATE enum

Operating mode before the transition.

REASON string

Reason for the transition being denied.

PSTATE_PACTIVE_POWER_TRANSITION

Trace pactive and power mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Power mode after the transition.

OLD_STATE enum

Power mode before the transition.

PSTATE_PACTIVE_POWER_TRANSITION_DENIED

Trace the denied pactive and power mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Power mode after the transition.

OLD_STATE enum

Power mode before the transition.

REASON string

Reason for the transition being denied.

PSTATE_PACTIVE_THREAD_OPMODE_TRANSITION

Trace pactive and Opmode mode transitions for threads on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Operating mode after the transition.

OLD_STATE enum

Operating mode before the transition.

QUANTUM_START

Called at the start of every quantum.

RUN_STATE

Run state transition. Fields:

INST_COUNT unsigned int

Ticks count at point of transition.

NEW enum

New run state.

OLD enum

Old run state.

SEMIHOSTING_CALL

Call of a semihost function occurred. Fields:

PC unsigned int

The program counter after the semihosting call.

SEMIHOSTING_PRECALL

About to call semihost. Fields:

EL2 bool

Use EL2 translation regime.

NS bool

Is Non-Secure.

REG_WIDTH unsigned int

The current register width in bytes.

SIGNAL

External signal state change. Fields:

SIGNAL enum

Signal that changed.

STATE bool

Signal asserted state.

START_COMPILE

Compilation started. Fields:

VADDR unsigned int

Instruction where compilation begins.

STG_MULTIPLE_MICRO_OPS_ADDRESS_RANGE

Address range for MTE stores with multiple uops. Fields:

ADDR_END unsigned int

Final address.

ADDR_START unsigned int

Starting address.

STORE_MULTIPLE_REGS_ADDRESS_RANGE

Address range for stores with multiple registers. Fields:

ADDR_END unsigned int

Final address.

ADDR_START unsigned int

Starting address.

SUBOPTIMAL_LDST_RETIRED

suboptimal load/store. Fields:

ACCESS_SIZE unsigned int

Log2 of the access width in bytes used for alignment checking (1: half-word, 2: word ...).

OPTIMAL_ALIGN_SELECTOR unsigned int

Selector to decide the optimal alignment size (1: access width, 2: 4-bytes, 3: 8-bytes, ..., 12: 4-kbytes).

VADDR unsigned int

The virtual address of the access.

SYNC

Called for every synchronization, that is every quantum break. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The instruction count of this CPU.

LOCAL_QUANTUM unsigned int

The local quantum of this CPU.

LOCAL_TIME unsigned int

The local time of this CPU.

SYNC_REASON enum

Reason for sync.

SYSCALL

System call instruction executed. Fields:

IMM unsigned int

Immediate value of the system call instruction.

TYPE enum

System call type.

VADDR unsigned int

Instruction that caused the system call.

SYSREG_READ

Trace reads from system registers. Fields:

REG_NAME enum

Register number.

VALUE unsigned int

The value read.

SYSREG_UPDATE128

Triggers when the system updates a register. Fields:

REG enum

Register number.

UNKNOWN unsigned int

Bits of the register which became unknown.

VALUE unsigned int

Value written to the register.

SYSREG_UPDATE32

Triggers when the system updates a register. Fields:

REG enum

Register number.

UNKNOWN unsigned int

Bits of the register which became unknown.

VALUE unsigned int

Value written to the register.

SYSREG_UPDATE64

Triggers when the system updates a register. Fields:

REG enum

Register number.

UNKNOWN unsigned int

Bits of the register which became unknown.

VALUE unsigned int

Value written to the register.

SYSREG_WRITE

Trace write to system registers. Fields:

REG_ACCESS enum

Register access status.

REG_NAME enum

Register number.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

Value written to the register.

UNALIGNED_LDST_RETIRED

Processor unaligned load/store. Fields:

ACCESS_SIZE unsigned int

Log2 of the access width in bytes used for alignment checking (1=>half-word, 2->word ...).

NUMBER_OF_BEATS unsigned int

Number of accesses (beats) in this burst (Total size in bytes is ACCESS_SIZE * NUMBER_OF_BEATS).

VADDR unsigned int

The virtual address of the access.

VFP_D_REGS

VFP/NEON D 64 bit register write. Fields:

ALIASING enum

Type of register aliasing used.

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a write to the corresponding bit in the register. For example, 0x1 would indicate a write to VALUE[0].

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

VFP_D_REGS_READ

VFP/NEON D 64 bit register read. Fields:

ID unsigned int

The register number.

VALUE unsigned int

The current value of the register.

VFP_Q_REGS

VFP/NEON Q 128 bit register write. Fields:

ALIASING enum

Type of register aliasing used.

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a write to the corresponding byte in the register. For example, 0x1 would indicate a write to VALUE[7:0].

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

VFP_Q_REGS_READ

VFP/NEON Q 128 bit register read. Fields:

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a read to the corresponding byte in the register. For example, 0x1 would indicate a read to VALUE[7:0].

VALUE unsigned int

The current value of the register.

VFP_SYS_REGS

Writes to the VFP/NEON units system registers. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ID enum

Which VFP system register is written.

MASK unsigned int

The mask indicating updated cumulative bits.

OLD_VALUE unsigned int

The register's old value overwritten.

VALUE unsigned int

The new value written to the VFP system register.

VFP_S_REGS

VFP/NEON S 32 bit register write. Fields:

ALIASING enum

Type of register aliasing used.

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a write to the corresponding bit in the register. For example, 0x1 would indicate a write to VALUE[0].

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

VFP_S_REGS_READ

VFP/NEON S 32 bit register read. Fields:

ID unsigned int

The register number.

VALUE unsigned int

The current value of the register.

WAYPOINT

Signals end of basic block execution. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ISET enum

Origin instruction set.

IS_COND bool

Indicates if this is a conditional waypoint.

PC unsigned int

Origin address (or 0 if unavailable).

TAKEN bool

Indicates if this waypoint was taken.

TARGET unsigned int

Destination address.

TARGET_ISET enum

Destination instruction set.

WFE_END

WFE ended. Fields:

INST_COUNT unsigned int

Ticks count when leaving WFE.

WFE_EVENT_REGISTER

WFE event register status: set/clear, reason. Fields:

INST_COUNT unsigned int

Ticks count.

REASON enum

Reason for set/clear. Only REASON==0 (Cleared by WFE) clears the bit, all other reasons set it.

WFE_IGNORED

WFE ignored. Fields:

EVENT bool

This WFE was ignored because the event register was set.

INST_COUNT unsigned int

Ticks count when ignoring WFE.

TRAPPED bool

This WFE was trapped.

WFE_START

WFE entered. Fields:

INST_COUNT unsigned int

Ticks count when entering WFE.

WFI_END

WFI ended. Fields:

INST_COUNT unsigned int

Ticks count when leaving WFI.

WFI_IGNORED

WFI ignored. Fields:

DISABLED bool

This WFI was ignored because WFI is disabled.

INST_COUNT unsigned int

Ticks count when ignoring WFI.

REASON enum

specifies reason why WFI trace was ignored.

TRAPPED bool

This WFI was trapped.

WFI_START

WFI entered. Fields:

INST_COUNT unsigned int

Ticks count when entering WFI.

WFI_WAKEUP

WFI wakeup. Fields:

INST_COUNT unsigned int

Ticks count when WFI wakeup occurred.

REASON enum

Reason for wakeup.

XPSR

Changes to the xPSR register. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

OLD_VALUE unsigned int

The old xPSR value.

UNKNOWN unsigned int

Bits within the register that have unknown value.

VALUE unsigned int

The new xPSR value.

2.49 ARM_Cortex-R4

This section describes the trace sources.

ASYNC_MEMORY_FAULT

Context ID Register write. Fields:

FAULT unsigned int

Fault status present in ESR.ISS encoding for Serror in ARCH64 and present in DFSR encoding for external abort format in ARCH32.

PADDR unsigned int

Physical Address (or 0 if unavailable).

ATOMIC_END_ACCESS

Bus trace access for atomics. Fields:

ACCESS_FAIL bool

Memory access failed.

ADDR unsigned int

The virtual address of the access.

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

COMPARE_VALUE unsigned int

Compare value for CAS.

LOAD_VALUE unsigned int

Loaded values.

NSDESC unsigned int

The security state of the access.

OPERAND_VALUE unsigned int

Operation's operand.

OPERATION enum

Operation type.

PADDR unsigned int

The physical address of the access.

PRIV bool

Is this a privileged access?.

ATOMIC_SLAVE_ACCESS

Information on the atomic operation at slave side. Fields:

ACCESS_RESULT unsigned int

Result of atomic access.

COMPARE_VALUE unsigned int

Value to be compared with memory by CAS.

LOAD_VALUE unsigned int

Value to be loaded from memory as a result of atomic operation.

MANAGER unsigned int

ID of manager which initiated the transaction.

NS bool

Transaction is Non-Secure.

OPERAND_VALUE unsigned int

Incoming operand value provided by core.

OPERATION enum

Type of atomic operation.

PADDR unsigned int

Full physical address of TCM.

STORE_VALUE unsigned int

Value to be stored to memory as a result of atomic operation.

ATOMIC_START_ACCESS

Bus trace access for atomics. Fields:

ADDR unsigned int

The virtual address of the access.

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

COMPARE_VALUE unsigned int

Compare value for CAS.

NSDESC unsigned int

The security state of the access.

OPERAND_VALUE unsigned int

Operation's operand.

OPERATION enum

Operation type.

PADDR unsigned int

The physical address of the access.

PRIV bool

Is this a privileged access?.

ArchMsg.Error.exit_code

DISPLAY exit code: %{EXITCODE} component kind: %{KIND} reason: %{REASON}. Fields:

EXITCODE signed int

Exit code to be returned.

KIND string

Component kind that invoked the exit code trace.

REASON string

Reason why the exit code trace was invoked (optional).

ArchMsg.Warning.ReportedRevisionRegisterRead

DISPLAY Reported revision and patch do not match that of the cpu. Fields:

ORIGINAL_VALUE unsigned int

The value the register would have based on the modelled revision.

REGISTER string

The name of the register that has ben read.

REPORTED_VALUE unsigned int

The value the register has been overridden to.

ArchMsg.Warning.cache_contents_unknown

A cache has been activated without being invalidated and may contain **UNKNOWN** content. DISPLAY %{SIDE:(D|I)}-Cache was enabled in %{REGIME:(SECURE|NON_SECURE|HYP)} regime but was not invalidated since power-on: cache lines could contain **UNKNOWN** content %{SIDE:(Note: D-side caches must invalidate SECURE lines even if the cache is only used by NON_SECURE code in order to ensure that unexpected dirty lines tagged SECURE are not naturally evicted)}. Fields:

REGIME unsigned int

Current translation regime.

SIDE unsigned int

0 d-side cache or unified cache, 1 i-side.

ArchMsg.Warning.recursive_branch

An instruction is performing a branch that targets the same instruction. If this was intended then a WFI might be more effective DISPLAY recursive branch detected at PC=%{PC}. PRIMARY KEY PC. Fields:

PC unsigned int

Address of instruction causing the branch.

ArchMsg.Warning.recursive_exception

An instruction has generated an exception that targets the same instruction. DISPLAY recursive %{TYPE} exception detected at PC=%{PC} (CPSR=%{CPSR}). PRIMARY KEY PC. Fields:

CPSR unsigned int

Processor state.

PC unsigned int

Address of instruction causing the exception.

TYPE enum

Type of exception.

ArchMsg.Warning.secure_vector_fetch_from_nonsecure

DISPLAY Secure vector table is being fetched from Non-secure memory.

BRANCH_MISPREDICT

Simulating branch mispredict. Fields:

PC unsigned int

Origin address (or 0 if unavailable).

BRA_DIR

Direct branches, to immediate address. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISSET enum

The instruction set of the branch instruction.

IS_COND bool

Indicates if this is a conditional branch.

IS_HINTED bool

Whether the conditional instruction is hinted.

IS_ISB bool

Whether the instruction is ISB.

IS_LINK bool

Indicates if this is a branch with link.

PC unsigned int

The address of the branch instruction.

TARGET_ISET enum

The instruction set after the branch.

TARGET_PC unsigned int

The address the instruction branches to.

BRA_INDIR

Indirect branches, perhaps to a register. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISET enum

The instruction set of the branch instruction.

IS_COND bool

Indicates if this is a conditional branch.

IS_HINTED bool

Whether the conditional instruction is hinted.

IS_ISB bool

Whether the instruction is ISB.

IS_LINK bool

Indicates if this is a branch with link.

PC unsigned int

The address of the branch instruction.

TARGET_ISET enum

The instruction set after the branch.

TARGET_PC unsigned int

The address the instruction branches to.

BRA_INDIR_CCFAIL

Indirect branches fail, perhaps to a register.

BRA_RET_CCFAIL

Branch return fail, perhaps to a register.

CACHE_MAINTENANCE_OP

Cache Maintenance Operation. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

DATA unsigned int

Specified MVA or Set/way.

FUNCTION enum

Clean / Invalidate.

SCOPE enum

Affected region.

SIDE enum

Inst / Data.

CCFAIL

Conditional instruction condition check fail. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CCFAIL_UNC

Unconditional instruction condition check fail. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CCPASS

Conditional instruction condition check pass. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CCPASS_UNC

Unconditional instruction condition check pass. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CODE_CACHE_MAINT

Code cache maintenance. Fields:

END_INCL unsigned int

Inclusive end address for VA or PA requests.

NS bool

Non-secure world for PA.

START unsigned int

Start address for VA or PA requests.

TYPE enum

Request type.

COMPILE_BLOCK_END

Last instruction of basic block translated. Fields:

VADDR unsigned int

Address of next instruction after this basic block.

COMPILE_BLOCK_START

First instruction of basic block translated. Fields:

VADDR unsigned int

Address of first instruction in basic block.

COMPILE_INST

ARM instruction compiled. Fields:

DISASS string

Disassembly of the instruction.

ISSET enum

The instruction set of this instruction.

ITSTATE unsigned int

The ITSTATE current for the instruction.

OPCODE unsigned int

The opcode of the instruction.

PC unsigned int

The address of the instruction.

SIZE unsigned int

The size of the instruction in bytes.

CONTEXTIDR

Context ID Register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

NS bool

Secure or nonsecure banked register is accessed.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The new value written.

CONTEXT_SYNC

Called for every context synchronization event. Fields:

ARCH_CSE bool

Architectural Context Synchronization Event.

PC unsigned int

The address of the synchronization instruction.

SYSREG_SYNC bool

System register synchronization event.

CORE_ENDIAN

Core BE8 Big-Endian state changed. Fields:

BE8 bool

Core BE8 Big-Endian state.

CORE_INFO

Static processor attributes. Only triggered by a call to DumpState(). Fields:

ARCH_PROFILE enum

The architecture profile of the core.

CLUSTER_ID unsigned int

The cluster ID of this processor.

CORE_NUM unsigned int

The number of this core in an MP processor.

DETAILED_CORE_FEATURES enum

Detailed core features as a bitfield. Enum values are bitmasks to extract the info from the field.

FPU_VERSION enum

The VFP version implemented by the core.

MEM_ARCH enum

The memory architecture of the core.

MIDR_EL1 unsigned int

MIDR_EL1 value.

NUM_CORES unsigned int

The number of cores in this MP processor.

QUANTUM_SIZE unsigned int

The default quantum size of the core (in ticks).

SECURITY_FEATURES bool

Does the core have security features?.

CORE_LOADS

Processor load accesses. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ACQREL enum

Is this an acquire/release.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DATA unsigned int

The data read or written.

ELEMENT_SIZE unsigned int

Width of each element.

ENDIANNESS enum

Byte ordering of this access.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

LOCK enum

Normal, exclusive or locked access.

MECID unsigned int

Memory Encryption Context ID, if applicable.

MECID2 unsigned int

Memory Encryption Context ID of the second page, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PAGE_SIZE unsigned int

Page size as $\log_2(\text{size in bytes})$.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK, 2=Exclusive Failed, 3=Unsupported.

SIZE unsigned int

Width of the access in bytes. Only required if DATA is not traced.

TRANS bool

Is this a translated access.

VADDR unsigned int

The virtual address of the access.

CORE_REGS

Changes of the core registers R0 to R14. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number, 0 to 14.

MODE enum

Bank of the register accessed.

OLD_VALUE unsigned int

The old value overwritten.

PHYS_ID enum

The physical register accessed.

VALUE unsigned int

The new value written to the register.

CORE_STALLED_TRANSITION

Detect when a core is prevented from progressing PChannel states. Fields:

CURRENT_STATE enum

Current power mode of core.

REASON string

The reason a transition was stalled, and what must be done to unblock it.

TARGET_STATE enum

Desired power mode of core after transition.

CORE_STORES

Processor store accesses. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ACQREL enum

Is this an acquire/release.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DATA unsigned int

The data read or written.

ELEMENT_SIZE unsigned int

Width of each element.

ENDIANNESS enum

Byte ordering of this access.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

LOCK enum

Normal, exclusive or locked access.

MECID unsigned int

Memory Encryption Context ID, if applicable.

MECID2 unsigned int

Memory Encryption Context ID of the second page, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PAGE_SIZE unsigned int

Page size as $\log_2(\text{size in bytes})$.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK, 2=Exclusive Failed, 3=Unsupported.

SIZE unsigned int

Width of the access in bytes. Only required if DATA is not traced.

TRANS bool

Is this a translated access.

VADDR unsigned int

The virtual address of the access.

CP14_READ

Reads to the System Coprocessor registers. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP14 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc1 unsigned int

opcode 1.

opc2 unsigned int

opcode 2.

CP14_WRITE

Writes to the System Coprocessor registers. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP14 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The new value written.

opc1 unsigned int

opcode 1.

opc2 unsigned int

opcode 2.

CP15_READ

System Coprocessor register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP15 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc1 unsigned int

opcode 1.

opc2 unsigned int

opcode 2.

CP15_READ64

System Coprocessor register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP15 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc unsigned int

opcode 1.

CP15_WRITE

System Control Coprocessor register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP15 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc1 unsigned int

opcode 1.

opc2 unsigned int

opcode 2.

CP15_WRITE64

System Coprocessor register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP15 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc unsigned int

opcode 1.

CPSR

CPSR change. (In AArch64 PSTATE represented in SPSR format). Fields:

CORE_NUM unsigned int

Core number in a multi processor.

OLD_VALUE unsigned int

The old CPSR value.

UNKNOWN unsigned int

Bits within the register that have unknown value.

VALUE unsigned int

The new CPSR value.

CRYPTO_SPEC

Every crypto instruction speculatively executed.

DATA_CACHE_SET_ALLOC_TAGS

Set allocation tags from DC GVA instruction. Fields:

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory

attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

NSDESC unsigned int

The security state of the access.

PADDR unsigned int

The physical address of the access.

PAS enum

The physical address space of the page.

RESPONSE enum

0=Aborted, 1=OK.

SIZE unsigned int

The size of memory in bytes.

TAG unsigned int

The value of the tag being set.

TAGGING_TYPE enum

Type of Memory tagging (Physical or Virtual).

TAG_VADDR unsigned int

The virtual address of the Virtual Tag if Virtual tagging is used; otherwise 0.

VADDR unsigned int

Virtual address used for setting allocation tags.

DATA_CACHE_ZERO

Zero and invalidate cache line from DC ZVA instruction. Fields:

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

NSDESC unsigned int

The security state of the access.

PADDR unsigned int

The physical address of the access.

PADDR_TAG unsigned int

If tag checked, the physical address tag for the access.

PAS enum

The physical address space of the page.

RESPONSE enum

0=Aborted, 1=OK.

SIZE unsigned int

The size of zero'd memory in bytes.

TAG_CHECKED bool

Is this access tag checked.

VADDR unsigned int

Virtual address of the zero'd cache line.

DMI_ALLOCATE

DMI allocated. Fields:

ACCESS_DIRECTION enum

READ / WRITE.

PADDR unsigned int

The physical address of the DMI.

PTR unsigned int

DMI range pointer.

SIDE enum

DSIDE / ISIDE.

SIZE unsigned int

The size of DMI.

VADDR unsigned int

The virtual address of the DMI..

DMI_HIT

DMI hit, access used DMI (only fire for small DMI regions i.e. less than a full page). Fields:

ACCESS_DIRECTION enum

READ / WRITE.

PADDR unsigned int

The physical address of the DMI.

SIDE enum

DSIDE / ISIDE.

SIZE unsigned int

The size of the DMI.

VADDR unsigned int

The virtual address of the DMI..

DMI_REVOKE

DMI revoked. Fields:

ACCESS_DIRECTION enum

READ / WRITE.

PADDR unsigned int

The physical address of the DMI.

SIDE enum

DSIDE / ISIDE.

SIZE unsigned int

The size of DMI.

VADDR unsigned int

The virtual address of the DMI..

DTCM_ACCESS

Trace TCM accesses.

DTCM_ACCESS_RAZ_WI

Trace TCM accesses. Fields:

ADDR unsigned int

Address of the **RAZ/WI** access.

END_COMPILE

Compilation end. Fields:

END_OF_PAGE_COUNT unsigned int

Number of basic blocks exited due to page boundary since **START_COMPILE**.

FETCHFAIL_COUNT unsigned int

Number of basic blocks exited due to ifetch failure **START_COMPILE**.

INST_COUNT unsigned int

Number of instructions compiled since **START_COMPILE**.

NONSEQ_COUNT unsigned int

Number of basic blocks exited due to non-sequential instructions since **START_COMPILE**.

PAGE_STRADDLE_COUNT unsigned int

Number of basic blocks exited due to unaligned instructions crossing page since **START_COMPILE**.

EXCEPTION

Exception taken. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ESR unsigned int

The value assigned to ESR when exception is taken.

ESR_EC unsigned int

The Value assigned to ESR.EC when exception is taken.

IS_PHYSICAL bool

Physical or Virtual exception.

IS_TRAP bool

Exception coming from a trap.

LR unsigned int

The value assigned to the link register.

PC unsigned int

The location where the exception occurred.

PREFERRED_RETURN unsigned int

The preferred return address for the exception.

TAKEN_LOCALLY bool

Whether exception is taken locally(same exception level).

TARGET_ISET enum

The instruction set of the exception handler code.

TARGET_PC unsigned int

The address the exception branches to.

VECTOR enum

The exception vector.

VECTOR_OFFSET unsigned int

The offset of an exception vector from the base address.

EXCEPTION_END

Every exception completed.

EXCEPTION_INFO

Trace human-readable reason of the current exception for data/instruction aborts, floating point exceptions and unknown exceptions. Fields:

DESCRIPTION string

Reason for current exception.

ESR_EC unsigned int

The value of ESR.EC when exception is taken.

EXCEPTION_RAISE

Every exception raised.

EXCEPTION_RETURN

Branches on leaving exception. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISET enum

The instruction set of the branch instruction.

IS_COND bool

Indicates if this is a conditional branch.

IS_HINTED bool

Whether the conditional instruction is hinted.

IS_ISB bool

Whether the instruction is ISB.

IS_LINK bool

Indicates if this is a branch with link.

PC unsigned int

The address of the branch instruction.

TARGET_ISET enum

The instruction set after the branch.

TARGET_PC unsigned int

The address the instruction branches to.

EXCEPTION_RETURN_PREBRANCH

Exception return event that must occur before PC and processor state have updated.

EXCEPTION_START

Every exception started.

FP_EXCEPTIONS_SIGNALLED

Trace which FP exceptions were signalled as a part of the FP operations performed by an instruction. Fields:

FP_EXCEPTIONS_SIGNALLED_MASK unsigned int

Indicates which FP exceptions were signalled by an instruction. This uses the same encoding as the cumulative floating-point exception bitfields in FPSR.

FP_STATE

Floating point state. Fields:

FPSR unsigned int

FPSR register value.

MASK unsigned int

The mask indicating updated cumulative bits.

FREQ_CHANGED

Core frequency changed. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The instruction count of this core.

NEW_FREQ signed int

The new frequency of this core (expressed in Hz).

OLD_FREQ signed int

The old frequency of this core (expressed in Hz).

HLT

HLT instruction occurred. Fields:

IMM16 unsigned int

The 16-bit immediate value encoded in HLT instruction.

INST

Every instruction executed. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DEBUG_STATE bool

The instruction is executed in debug state.

DISASS string

Disassembly of instruction.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISSET enum

The current instruction set.

ITSTATE unsigned int

The current ITSTATE.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

MODE enum

The mode the core is in.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

OPCODE unsigned int

The opcode of the instruction.

PADDR unsigned int

The physical address of the instruction.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the instruction.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the page.

PC unsigned int

The address of the instruction.

SECURITY_STATE enum

The core's security state.

SIZE unsigned int

The size of the instruction in bytes.

INST_END

Every instruction completed.

INST_START

Every instruction started. Fields:

DEBUG_STATE bool

The instruction is executed in debug state.

ISSET enum

The current instruction set.

MODE enum

The mode the core is in.

NS enum

The current Secure State.

PC unsigned int

The address of the conditional instruction.

SECURITY_STATE enum

The current Security State.

INST_STRADDLE

Instruction straddles boundary. Fields:

DEBUG_STATE bool

The instruction is executed in debug state.

ISSET enum

The current instruction set.

MODE enum

The mode the core is in.

NS enum

The current Secure State.

PC unsigned int

The address of the conditional instruction.

SECURITY_STATE enum

The current Security State.

ISB

ISB instruction occurred. Fields:

PC unsigned int

The address of the ISB instruction.

ITCM_ACCESS

Trace TCM accesses.

ITCM_ACCESS_RAZ_WI

Trace TCM accesses. Fields:

ADDR unsigned int

Address of the **RAZ/WI** access.

LOAD_MULTIPLE_REGS_LIST

List of destination registers (not source) of a multiple registers load instruction. Fields:

REG string

Register to add into the list.

LOCAL_MONITOR

Local monitor activity. Fields:

PADDR unsigned int

Local Monitor Address.

State enum

State of the monitor (Open/Exclusive).

MODE_CHANGE

Mode change. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

MODE enum

The new mode.

OLD_MODE enum

The old mode.

MPU_TRANS

Address translation information. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

EXEC_PERM enum

Execution Permission.

INNERCACHE_RA bool

Is the inner cache allocate on read.

INNERCACHE_TYPE enum

Inner Caching scheme (NC/MB/WA).

INNERCACHE_WA bool

Is the inner cache allocate on write.

MEMTYPE enum

Memory type.

OUTERCACHE_RA bool

Is the outer cache allocate on read.

OUTERCACHE_TYPE enum

Outer Caching scheme (NC/MB/WA).

OUTERCACHE_WA bool

Is the outer cache allocate on write.

PADDR unsigned int

Address of the access.

PAGESIZE unsigned int

Page size as log2(size).

READ_PERM enum

Read Permission.

REG_NUM unsigned int

MPU register based region-number 0-15 (else 0xFF=none, 0xBB/0xDD=background/default).

SH enum

Shareability.

SIDE enum

Inst / Data.

WRITE_PERM enum

Write Permission.

PERIODIC

Called for every quantum. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The instruction count of this CPU.

PC unsigned int

The address of the next instruction to be executed on this CPU.

POTINDEX_UPDATE

Value of POT index for current pc changed. Fields:

NEW_POTINDEX unsigned int

New POTIndex.

OLD_POTINDEX unsigned int

Old POTIndex.

PRELOAD_DATA

Data preload from PLD instruction. Fields:

VADDR unsigned int

Virtual address of the data that should be preloaded.

PRELOAD_INST

Instruction preload from PLI instruction. Fields:

VADDR unsigned int

Virtual address of the instruction that should be preloaded.

PRE_CORE_LOAD

Trace just before a core load. Fields:

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

PRE_CORE_STORE

Trace just before a core store. Fields:

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

PSTATE

PSTATE fields update. Fields:

ALLINT unsigned int

All interrupt mask bit.

BTYPE unsigned int

Branch target identification bit.

CORE_NUM unsigned int

Core number in a multi processor.

DAIF unsigned int

Exception mask bits.

DIT unsigned int

Data Independent Timing bit.

EL unsigned int

Current Exception level field.

EXLOCK unsigned int

Exception return state lock bit.

IL unsigned int

Illegal Execution state bit.

NZCV unsigned int

Condition flags.

PACM unsigned int

PACM bit.

PAN unsigned int

Privileged Access Never state bit.

PM unsigned int

PMU exception mask bit.

PPEND unsigned int

PMU exception pending bit.

SM unsigned int

StreamingMode status bit.

SP unsigned int

Stack pointer register selection bit.

SS unsigned int

Software Step bit.

SSBS unsigned int

Speculative Store Bypass Safe bit.

TCO unsigned int

Tag Check Override bit.

TOUCHED unsigned int

Mask of the fields that have been touched since this trace has last been emitted. Note that touched does not necessarily mean the value changed. The index of the field is the bit index in the mask plus the index of this field plus one.

UAO unsigned int

User Access Override bit.

UINJ unsigned int

Undefined Instruction exceptions bit.

ZA unsigned int

ZA storage and ZT0 register enable bit.

nRW unsigned int

Current Execution state bit.

PSTATE_PACTIVE_OPMODE_TRANSITION

Trace pactive and Opmode mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Operating mode after the transition.

OLD_STATE enum

Operating mode before the transition.

PSTATE_PACTIVE_OPMODE_TRANSITION_DENIED

Trace the denied pactive and Opmode mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Operating mode after the transition.

OLD_STATE enum

Operating mode before the transition.

REASON string

Reason for the transition being denied.

PSTATE_PACTIVE_POWER_TRANSITION

Trace pactive and power mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Power mode after the transition.

OLD_STATE enum

Power mode before the transition.

PSTATE_PACTIVE_POWER_TRANSITION_DENIED

Trace the denied pactive and power mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Power mode after the transition.

OLD_STATE enum

Power mode before the transition.

REASON string

Reason for the transition being denied.

PSTATE_PACTIVE_THREAD_OPMODE_TRANSITION

Trace pactive and Opmode mode transitions for threads on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Operating mode after the transition.

OLD_STATE enum

Operating mode before the transition.

QUANTUM_START

Called at the start of every quantum.

RUN_STATE

Run state transition. Fields:

INST_COUNT unsigned int

Ticks count at point of transition.

NEW enum

New run state.

OLD enum

Old run state.

SEMIHOSTING_CALL

Call of a semihost function occurred. Fields:

PC unsigned int

The program counter after the semihosting call.

SEMIHOSTING_PRECALL

About to call semihost. Fields:

EL2 bool

Use EL2 translation regime.

NS bool

Is Non-Secure.

REG_WIDTH unsigned int

The current register width in bytes.

SIGNAL

External signal state change. Fields:

SIGNAL enum

Signal that changed.

STATE bool

Signal asserted state.

SPSR

SPSR change. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

MODE enum

Which of the banked SPSR registers is written.

OLD_VALUE unsigned int

The old SPSR value.

VALUE unsigned int

The new SPSR value.

START_COMPILE

Compilation started. Fields:

VADDR unsigned int

Instruction where compilation begins.

STG_MULTIPLE_MICRO_OPS_ADDRESS_RANGE

Address range for MTE stores with multiple uops. Fields:

ADDR_END unsigned int

Final address.

ADDR_START unsigned int

Starting address.

STORE_MULTIPLE_REGS_ADDRESS_RANGE

Address range for stores with multiple registers. Fields:

ADDR_END unsigned int

Final address.

ADDR_START unsigned int

Starting address.

SUBOPTIMAL_LDST_RETIRED

suboptimal load/store. Fields:

ACCESS_SIZE unsigned int

Log2 of the access width in bytes used for alignment checking (1: half-word, 2: word ...).

OPTIMAL_ALIGN_SELECTOR unsigned int

Selector to decide the optimal alignment size (1: access width, 2: 4-bytes, 3: 8-bytes, ..., 12: 4-kbytes).

VADDR unsigned int

The virtual address of the access.

SYNC

Called for every synchronization, that is every quantum break. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The instruction count of this CPU.

LOCAL_QUANTUM unsigned int

The local quantum of this CPU.

LOCAL_TIME unsigned int

The local time of this CPU.

SYNC_REASON enum

Reason for sync.

SYSCALL

System call instruction executed. Fields:

IMM unsigned int

Immediate value of the system call instruction.

TYPE enum

System call type.

VADDR unsigned int

Instruction that caused the system call.

SYSREG_READ128

System Coprocessor 128 bit register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

REGNUM enum

Internal register number.

REGNUM_PHYS enum

Internal register number physically accessed (e.g. in presense of VHE redirection).

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_READ64

System Coprocessor register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REGNUM enum

Internal register number.

REGNUM_OPERAND enum

Register number corresponding to the instruction operand.

REGNUM_PHYS enum

Internal register number physically accessed (e.g. in presense of VHE redirection).

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_UPDATE128

Triggers when the system updates a register. Fields:

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG enum

Register number.

UNKNOWN unsigned int

Bits of the register which became unknown.

VALUE unsigned int

Value written to the register.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_UPDATE32

Triggers when the system updates a register. Fields:

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG enum

Register number.

UNKNOWN unsigned int

Bits of the register which became unknown.

VALUE unsigned int

Value written to the register.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_UPDATE64

Triggers when the system updates a register. Fields:

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG enum

Register number.

UNKNOWN unsigned int

Bits of the register which became unknown.

VALUE unsigned int

Value written to the register.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_WRITE128

System Coprocessor 128 bit register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

REGNUM enum

Internal register number.

REGNUM_PHYS enum

Internal register number physically accessed (e.g. in presense of VHE redirection).

UNDEF bool

The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_WRITE64

System Coprocessor register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REGNUM enum

Internal register number.

REGNUM_OPERAND enum

Register number corresponding to the instruction operand.

REGNUM_PHYS enum

Internal register number physically accessed (e.g. in presense of VHE redirection).

UNDEF bool

The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

UNALIGNED_LDST_RETIRED

Processor unaligned load/store. Fields:

ACCESS_SIZE unsigned int

Log2 of the access width in bytes used for alignment checking (1=>half-word, 2->word ...).

NUMBER_OF_BEATS unsigned int

Number of accesses (beats) in this burst (Total size in bytes is ACCESS_SIZE * NUMBER_OF_BEATS).

VADDR unsigned int

The virtual address of the access.

VFP_D_REGS

VFP/NEON D 64 bit register write. Fields:

ALIASING enum

Type of register aliasing used.

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a write to the corresponding bit in the register. For example, 0x1 would indicate a write to VALUE[0].

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

VFP_D_REGS_READ

VFP/NEON D 64 bit register read. Fields:

ID unsigned int

The register number.

VALUE unsigned int

The current value of the register.

VFP_Q_REGS

VFP/NEON Q 128 bit register write. Fields:

ALIASING enum

Type of register aliasing used.

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a write to the corresponding byte in the register. For example, 0x1 would indicate a write to VALUE[7:0].

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

VFP_Q_REGS_READ

VFP/NEON Q 128 bit register read. Fields:

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a read to the corresponding byte in the register. For example, 0x1 would indicate a read to VALUE[7:0].

VALUE unsigned int

The current value of the register.

VFP_SYS_REGS

Writes to the VFP/NEON units system registers. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ID enum

Which VFP system register is written.

MASK unsigned int

The mask indicating updated cumulative bits.

OLD_VALUE unsigned int

The register's old value overwritten.

VALUE unsigned int

The new value written to the VFP system register.

VFP_S_REGS

VFP/NEON S 32 bit register write. Fields:

ALIASING enum

Type of register aliasing used.

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a write to the corresponding bit in the register. For example, 0x1 would indicate a write to VALUE[0].

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

VFP_S_REGS_READ

VFP/NEON S 32 bit register read. Fields:

ID unsigned int

The register number.

VALUE unsigned int

The current value of the register.

WAYPOINT

Signals end of basic block execution. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ISSET enum

Origin instruction set.

IS_COND bool

Indicates if this is a conditional waypoint.

PC unsigned int

Origin address (or 0 if unavailable).

TAKEN bool

Indicates if this waypoint was taken.

TARGET unsigned int

Destination address.

TARGET_ISET enum

Destination instruction set.

WFE_END

WFE ended. Fields:

INST_COUNT unsigned int

Ticks count when leaving WFE.

WFE_EVENT_REGISTER

WFE event register status: set/clear, reason. Fields:

INST_COUNT unsigned int

Ticks count.

REASON enum

Reason for set/clear. Only REASON==0 (Cleared by WFE) clears the bit, all other reasons set it.

WFE_IGNORED

WFE ignored. Fields:

EVENT bool

This WFE was ignored because the event register was set.

INST_COUNT unsigned int

Ticks count when ignoring WFE.

TRAPPED bool

This WFE was trapped.

WFE_START

WFE entered. Fields:

INST_COUNT unsigned int

Ticks count when entering WFE.

WFI_END

WFI ended. Fields:

INST_COUNT unsigned int

Ticks count when leaving WFI.

WFI_IGNORED

WFI ignored. Fields:

DISABLED bool

This WFI was ignored because WFI is disabled.

INST_COUNT unsigned int

Ticks count when ignoring WFI.

REASON enum

specifies reason why WFI trace was ignored.

TRAPPED bool

This WFI was trapped.

WFI_START

WFI entered. Fields:

INST_COUNT unsigned int

Ticks count when entering WFI.

WFI_WAKEUP

WFI wakeup. Fields:

INST_COUNT unsigned int

Ticks count when WFI wakeup occurred.

REASON enum

Reason for wakeup.

2.50 ARM_Cortex-R5

This section describes the trace sources.

ASYNC_MEMORY_FAULT

Context ID Register write. Fields:

FAULT unsigned int

Fault status present in ESR.ISS encoding for Serror in ARCH64 and present in DFSR encoding for external abort format in ARCH32.

PADDR unsigned int

Physical Address (or 0 if unavailable).

ATOMIC_END_ACCESS

Bus trace access for atomics. Fields:

ACCESS_FAIL bool

Memory access failed.

ADDR unsigned int

The virtual address of the access.

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

COMPARE_VALUE unsigned int

Compare value for CAS.

LOAD_VALUE unsigned int

Loaded values.

NSDESC unsigned int

The security state of the access.

OPERAND_VALUE unsigned int

Operation's operand.

OPERATION enum

Operation type.

PADDR unsigned int

The physical address of the access.

PRIV bool

Is this a privileged access?.

ATOMIC_SLAVE_ACCESS

Information on the atomic operation at slave side. Fields:

ACCESS_RESULT unsigned int

Result of atomic access.

COMPARE_VALUE unsigned int

Value to be compared with memory by CAS.

LOAD_VALUE unsigned int

Value to be loaded from memory as a result of atomic operation.

MANAGER unsigned int

ID of manager which initiated the transaction.

NS bool

Transaction is Non-Secure.

OPERAND_VALUE unsigned int

Incoming operand value provided by core.

OPERATION enum

Type of atomic operation.

PADDR unsigned int

Full physical address of TCM.

STORE_VALUE unsigned int

Value to be stored to memory as a result of atomic operation.

ATOMIC_START_ACCESS

Bus trace access for atomics. Fields:

ADDR unsigned int

The virtual address of the access.

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

COMPARE_VALUE unsigned int

Compare value for CAS.

NSDESC unsigned int

The security state of the access.

OPERAND_VALUE unsigned int

Operation's operand.

OPERATION enum

Operation type.

PADDR unsigned int

The physical address of the access.

PRIV bool

Is this a privileged access?.

ArchMsg.Error.exit_code

DISPLAY exit code: %{EXITCODE} component kind: %{KIND} reason: %{REASON}. Fields:

EXITCODE signed int

Exit code to be returned.

KIND string

Component kind that invoked the exit code trace.

REASON string

Reason why the exit code trace was invoked (optional).

ArchMsg.Warning.ReportedRevisionRegisterRead

DISPLAY Reported revision and patch do not match that of the cpu. Fields:

ORIGINAL_VALUE unsigned int

The value the register would have based on the modelled revision.

REGISTER string

The name of the register that has been read.

REPORTED_VALUE unsigned int

The value the register has been overridden to.

ArchMsg.Warning.cache_contents_unknown

A cache has been activated without being invalidated and may contain **UNKNOWN** content. DISPLAY %{SIDE:(D|I)}-Cache was enabled in %{REGIME:(SECURE|NON_SECURE|HYP)} regime but was not invalidated since power-on: cache lines could contain **UNKNOWN** content %{SIDE:(Note: D-side caches must invalidate SECURE lines even if the cache is only used by NON_SECURE code in order to ensure that unexpected dirty lines tagged SECURE are not naturally evicted)}. Fields:

REGIME unsigned int

Current translation regime.

SIDE unsigned int

0 d-side cache or unified cache, 1 i-side.

ArchMsg.Warning.recursive_branch

An instruction is performing a branch that targets the same instruction. If this was intended then a WFI might be more effective. DISPLAY recursive branch detected at PC=%{PC}. PRIMARY KEY PC. Fields:

PC unsigned int

Address of instruction causing the branch.

ArchMsg.Warning.recursive_exception

An instruction has generated an exception that targets the same instruction. DISPLAY recursive %{TYPE} exception detected at PC=%{PC} (CPSR=%{CPSR}). PRIMARY KEY PC. Fields:

CPSR unsigned int

Processor state.

PC unsigned int

Address of instruction causing the exception.

TYPE enum

Type of exception.

ArchMsg.Warning.secure_vector_fetch_from_nonsecure

DISPLAY Secure vector table is being fetched from Non-secure memory.

BRANCH_MISPREDICT

Simulating branch mispredict. Fields:

PC unsigned int

Origin address (or 0 if unavailable).

BRA_DIR

Direct branches, to immediate address. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISET enum

The instruction set of the branch instruction.

IS_COND bool

Indicates if this is a conditional branch.

IS_HINTED bool

Whether the conditional instruction is hinted.

IS_ISB bool

Whether the instruction is ISB.

IS_LINK bool

Indicates if this is a branch with link.

PC unsigned int

The address of the branch instruction.

TARGET_ISET enum

The instruction set after the branch.

TARGET_PC unsigned int

The address the instruction branches to.

BRA_INDIR

Indirect branches, perhaps to a register. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISET enum

The instruction set of the branch instruction.

IS_COND bool

Indicates if this is a conditional branch.

IS_HINTED bool

Whether the conditional instruction is hinted.

IS_ISB bool

Whether the instruction is ISB.

IS_LINK bool

Indicates if this is a branch with link.

PC unsigned int

The address of the branch instruction.

TARGET_ISET enum

The instruction set after the branch.

TARGET_PC unsigned int

The address the instruction branches to.

BRA_INDIR_CCFAIL

Indirect branches fail, perhaps to a register.

BRA_RET_CCFAIL

Branch return fail, perhaps to a register.

CACHE_MAINTENANCE_OP

Cache Maintenance Operation. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

DATA unsigned int

Specified MVA or Set/way.

FUNCTION enum

Clean / Invalidate.

SCOPE enum

Affected region.

SIDE enum

Inst / Data.

CCFAIL

Conditional instruction condition check fail. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CCFAIL_UNC

Unconditional instruction condition check fail. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CCPASS

Conditional instruction condition check pass. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CCPASS_UNC

Unconditional instruction condition check pass. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CODE_CACHE_MAINT

Code cache maintenance. Fields:

END_INCL unsigned int

Inclusive end address for VA or PA requests.

NS bool

Non-secure world for PA.

START unsigned int

Start address for VA or PA requests.

TYPE enum

Request type.

COMPILE_BLOCK_END

Last instruction of basic block translated. Fields:

VADDR unsigned int

Address of next instruction after this basic block.

COMPILE_BLOCK_START

First instruction of basic block translated. Fields:

VADDR unsigned int

Address of first instruction in basic block.

COMPILE_INST

ARM instruction compiled. Fields:

DISASS string

Disassembly of the instruction.

ISET enum

The instruction set of this instruction.

ITSTATE unsigned int

The ITSTATE current for the instruction.

OPCODE unsigned int

The opcode of the instruction.

PC unsigned int

The address of the instruction.

SIZE unsigned int

The size of the instruction in bytes.

CONTEXTIDR

Context ID Register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

NS bool

Secure or nonsecure banked register is accessed.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The new value written.

CONTEXT_SYNC

Called for every context synchronization event. Fields:

ARCH_CSE bool

Architectural Context Synchronization Event.

PC unsigned int

The address of the synchronization instruction.

SYSREG_SYNC bool

System register synchronization event.

CORE_ENDIAN

Core BE8 Big-Endian state changed. Fields:

BE8 bool

Core BE8 Big-Endian state.

CORE_INFO

Static processor attributes. Only triggered by a call to DumpState(). Fields:

ARCH_PROFILE enum

The architecture profile of the core.

CLUSTER_ID unsigned int

The cluster ID of this processor.

CORE_NUM unsigned int

The number of this core in an MP processor.

DETAILED_CORE_FEATURES enum

Detailed core features as a bitfield. Enum values are bitmasks to extract the info from the field.

FPU_VERSION enum

The VFP version implemented by the core.

MEM_ARCH enum

The memory architecture of the core.

MIDR_EL1 unsigned int

MIDR_EL1 value.

NUM_CORES unsigned int

The number of cores in this MP processor.

QUANTUM_SIZE unsigned int

The default quantum size of the core (in ticks).

SECURITY_FEATURES bool

Does the core have security features?.

CORE_LOADS

Processor load accesses. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ACQREL enum

Is this an acquire/release.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DATA unsigned int

The data read or written.

ELEMENT_SIZE unsigned int

Width of each element.

ENDIANNESS enum

Byte ordering of this access.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

LOCK enum

Normal, exclusive or locked access.

MECID unsigned int

Memory Encryption Context ID, if applicable.

MECID2 unsigned int

Memory Encryption Context ID of the second page, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PAGE_SIZE unsigned int

Page size as $\log_2(\text{size in bytes})$.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK, 2=Exclusive Failed, 3=Unsupported.

SIZE unsigned int

Width of the access in bytes. Only required if DATA is not traced.

TRANS bool

Is this a translated access.

VADDR unsigned int

The virtual address of the access.

CORE_REGS

Changes of the core registers R0 to R14. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number, 0 to 14.

MODE enum

Bank of the register accessed.

OLD_VALUE unsigned int

The old value overwritten.

PHYS_ID enum

The physical register accessed.

VALUE unsigned int

The new value written to the register.

CORE_STALLED_TRANSITION

Detect when a core is prevented from progressing PChannel states. Fields:

CURRENT_STATE enum

Current power mode of core.

REASON string

The reason a transition was stalled, and what must be done to unblock it.

TARGET_STATE enum

Desired power mode of core after transition.

CORE_STORES

Processor store accesses. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ACQREL enum

Is this an acquire/release.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DATA unsigned int

The data read or written.

ELEMENT_SIZE unsigned int

Width of each element.

ENDIANNESS enum

Byte ordering of this access.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

LOCK enum

Normal, exclusive or locked access.

MECID unsigned int

Memory Encryption Context ID, if applicable.

MECID2 unsigned int

Memory Encryption Context ID of the second page, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PAGE_SIZE unsigned int

Page size as $\log_2(\text{size in bytes})$.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK, 2=Exclusive Failed, 3=Unsupported.

SIZE unsigned int

Width of the access in bytes. Only required if DATA is not traced.

TRANS bool

Is this a translated access.

VADDR unsigned int

The virtual address of the access.

CP14_READ

Reads to the System Coprocessor registers. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP14 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc1 unsigned int

opcode 1.

opc2 unsigned int

opcode 2.

CP14_WRITE

Writes to the System Coprocessor registers. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP14 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The new value written.

opc1 unsigned int

opcode 1.

opc2 unsigned int

opcode 2.

CP15_READ

System Coprocessor register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP15 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc1 unsigned int

opcode 1.

opc2 unsigned int

opcode 2.

CP15_READ64

System Coprocessor register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP15 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc unsigned int

opcode 1.

CP15_WRITE

System Control Coprocessor register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP15 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc1 unsigned int

opcode 1.

opc2 unsigned int

opcode 2.

CP15_WRITE64

System Coprocessor register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP15 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc unsigned int

opcode 1.

CPSR

CPSR change. (In AArch64 PSTATE represented in SPSR format). Fields:

CORE_NUM unsigned int

Core number in a multi processor.

OLD_VALUE unsigned int

The old CPSR value.

UNKNOWN unsigned int

Bits within the register that have unknown value.

VALUE unsigned int

The new CPSR value.

CRYPTO_SPEC

Every crypto instruction speculatively executed.

DATA_CACHE_SET_ALLOC_TAGS

Set allocation tags from DC GVA instruction. Fields:

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

NSDESC unsigned int

The security state of the access.

PADDR unsigned int

The physical address of the access.

PAS enum

The physical address space of the page.

RESPONSE enum

0=Aborted, 1=OK.

SIZE unsigned int

The size of memory in bytes.

TAG unsigned int

The value of the tag being set.

TAGGING_TYPE enum

Type of Memory tagging (Physical or Virtual).

TAG_VADDR unsigned int

The virtual address of the Virtual Tag if Virtual tagging is used; otherwise 0.

VADDR unsigned int

Virtual address used for setting allocation tags.

DATA_CACHE_ZERO

Zero and invalidate cache line from DC ZVA instruction. Fields:

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

NSDESC unsigned int

The security state of the access.

PADDR unsigned int

The physical address of the access.

PADDR_TAG unsigned int

If tag checked, the physical address tag for the access.

PAS enum

The physical address space of the page.

RESPONSE enum

0=Aborted, 1=OK.

SIZE unsigned int

The size of zero'd memory in bytes.

TAG_CHECKED bool

Is this access tag checked.

VADDR unsigned int

Virtual address of the zero'd cache line.

DMI_ALLOCATE

DMI allocated. Fields:

ACCESS_DIRECTION enum

READ / WRITE.

PADDR unsigned int

The physical address of the DMI.

PTR unsigned int

DMI range pointer.

SIDE enum

DSIDE / ISIDE.

SIZE unsigned int

The size of DMI.

VADDR unsigned int

The virtual address of the DMI..

DMI_HIT

DMI hit, access used DMI (only fire for small DMI regions i.e. less than a full page). Fields:

ACCESS_DIRECTION enum

READ / WRITE.

PADDR unsigned int

The physical address of the DMI.

SIDE enum

DSIDE / ISIDE.

SIZE unsigned int

The size of the DMI.

VADDR unsigned int

The virtual address of the DMI..

DMI_REVOKE

DMI revoked. Fields:

ACCESS_DIRECTION enum

READ / WRITE.

PADDR unsigned int

The physical address of the DMI.

SIDE enum

DSIDE / ISIDE.

SIZE unsigned int

The size of DMI.

VADDR unsigned int

The virtual address of the DMI..

DTCM_ACCESS

Trace TCM accesses.

DTCM_ACCESS_RAZ_WI

Trace TCM accesses. Fields:

ADDR unsigned int

Address of the **RAZ/WI** access.

END_COMPILE

Compilation end. Fields:

END_OF_PAGE_COUNT unsigned int

Number of basic blocks exited due to page boundary since START_COMPILE.

FETCHFAIL_COUNT unsigned int

Number of basic blocks exited due to ifetch failure START_COMPILE.

INST_COUNT unsigned int

Number of instructions compiled since START_COMPILE.

NONSEQ_COUNT unsigned int

Number of basic blocks exited due to non-sequential instructions since START_COMPILE.

PAGE_STRADDLE_COUNT unsigned int

Number of basic blocks exited due to unaligned instructions crossing page since START_COMPILE.

EXCEPTION

Exception taken. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ESR unsigned int

The value assigned to ESR when exception is taken.

ESR_EC unsigned int

The Value assigned to ESR.EC when exception is taken.

IS_PHYSICAL bool

Physical or Virtual exception.

IS_TRAP bool

Exception coming from a trap.

LR unsigned int

The value assigned to the link register.

PC unsigned int

The location where the exception occurred.

PREFERRED_RETURN unsigned int

The preferred return address for the exception.

TAKEN_LOCALLY bool

Whether exception is taken locally(same exception level).

TARGET_ISET enum

The instruction set of the exception handler code.

TARGET_PC unsigned int

The address the exception branches to.

VECTOR enum

The exception vector.

VECTOR_OFFSET unsigned int

The offset of an exception vector from the base address.

EXCEPTION_END

Every exception completed.

EXCEPTION_INFO

Trace human-readable reason of the current exception for data/instruction aborts, floating point exceptions and unknown exceptions. Fields:

DESCRIPTION string

Reason for current exception.

ESR_EC unsigned int

The value of ESR.EC when exception is taken.

EXCEPTION_RAISE

Every exception raised.

EXCEPTION_RETURN

Branches on leaving exception. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISET enum

The instruction set of the branch instruction.

IS_COND bool

Indicates if this is a conditional branch.

IS_HINTED bool

Whether the conditional instruction is hinted.

IS_ISB bool

Whether the instruction is ISB.

IS_LINK bool

Indicates if this is a branch with link.

PC unsigned int

The address of the branch instruction.

TARGET_ISET enum

The instruction set after the branch.

TARGET_PC unsigned int

The address the instruction branches to.

EXCEPTION_RETURN_PREBRANCH

Exception return event that must occur before PC and processor state have updated.

EXCEPTION_START

Every exception started.

FP_EXCEPTIONS_SIGNALLED

Trace which FP exceptions were signalled as a part of the FP operations performed by an instruction. Fields:

FP_EXCEPTIONS_SIGNALLED_MASK unsigned int

Indicates which FP exceptions were signalled by an instruction. This uses the same encoding as the cumulative floating-point exception bitfields in FPSR.

FP_STATE

Floating point state. Fields:

FPSR unsigned int

FPSR register value.

MASK unsigned int

The mask indicating updated cumulative bits.

FREQ_CHANGED

Core frequency changed. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The instruction count of this core.

NEW_FREQ signed int

The new frequency of this core (expressed in Hz).

OLD_FREQ signed int

The old frequency of this core (expressed in Hz).

HLT

HLT instruction occurred. Fields:

IMM16 unsigned int

The 16-bit immediate value encoded in HLT instruction.

INST

Every instruction executed. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DEBUG_STATE bool

The instruction is executed in debug state.

DISASS string

Disassembly of instruction.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISET enum

The current instruction set.

ITSTATE unsigned int

The current ITSTATE.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

MODE enum

The mode the core is in.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

OPCODE unsigned int

The opcode of the instruction.

PADDR unsigned int

The physical address of the instruction.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the instruction.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the page.

PC unsigned int

The address of the instruction.

SECURITY_STATE enum

The core's security state.

SIZE unsigned int

The size of the instruction in bytes.

INST_END

Every instruction completed.

INST_START

Every instruction started. Fields:

DEBUG_STATE bool

The instruction is executed in debug state.

ISET enum

The current instruction set.

MODE enum

The mode the core is in.

NS enum

The current Secure State.

PC unsigned int

The address of the conditional instruction.

SECURITY_STATE enum

The current Security State.

INST_STRADDLE

Instruction straddles boundary. Fields:

DEBUG_STATE bool

The instruction is executed in debug state.

ISET enum

The current instruction set.

MODE enum

The mode the core is in.

NS enum

The current Secure State.

PC unsigned int

The address of the conditional instruction.

SECURITY_STATE enum

The current Security State.

ISB

ISB instruction occurred. Fields:

PC unsigned int

The address of the ISB instruction.

ITCM_ACCESS

Trace TCM accesses.

ITCM_ACCESS_RAZ_WI

Trace TCM accesses. Fields:

ADDR unsigned int

Address of the **RAZ/WI** access.

LOAD_MULTIPLE_REGS_LIST

List of destination registers (not source) of a multiple registers load instruction. Fields:

REG string

Register to add into the list.

LOCAL_MONITOR

Local monitor activity. Fields:

PADDR unsigned int

Local Monitor Address.

State enum

State of the monitor (Open/Exclusive).

MODE_CHANGE

Mode change. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

MODE enum

The new mode.

OLD_MODE enum

The old mode.

MPU_TRANS

Address translation information. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

EXEC_PERM enum

Execution Permission.

INNERCACHE_RA bool

Is the inner cache allocate on read.

INNERCACHE_TYPE enum

Inner Caching scheme (NC/MB/WA).

INNERCACHE_WA bool

Is the inner cache allocate on write.

MEMTYPE enum

Memory type.

OUTERCACHE_RA bool

Is the outer cache allocate on read.

OUTERCACHE_TYPE enum

Outer Caching scheme (NC/MB/WA).

OUTERCACHE_WA bool

Is the outer cache allocate on write.

PADDR unsigned int

Address of the access.

PAGESIZE unsigned int

Page size as log2(size).

READ_PERM enum

Read Permission.

REG_NUM unsigned int

MPU register based region-number 0-15 (else 0xFF=none, 0xBB/0xDD=background/default).

SH enum

Shareability.

SIDE enum

Inst / Data.

WRITE_PERM enum

Write Permission.

PERIODIC

Called for every quantum. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The instruction count of this CPU.

PC unsigned int

The address of the next instruction to be executed on this CPU.

POTINDEX_UPDATE

Value of POT index for current pc changed. Fields:

NEW_POTINDEX unsigned int

New POTIndex.

OLD_POTINDEX unsigned int

Old POTIndex.

PRELOAD_DATA

Data preload from PLD instruction. Fields:

VADDR unsigned int

Virtual address of the data that should be preloaded.

PRELOAD_INST

Instruction preload from PLI instruction. Fields:

VADDR unsigned int

Virtual address of the instruction that should be preloaded.

PRE_CORE_LOAD

Trace just before a core load. Fields:

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

PRE_CORE_STORE

Trace just before a core store. Fields:

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

PSTATE

PSTATE fields update. Fields:

ALLINT unsigned int

All interrupt mask bit.

BTYPE unsigned int

Branch target identification bit.

CORE_NUM unsigned int

Core number in a multi processor.

DAIF unsigned int

Exception mask bits.

DIT unsigned int

Data Independent Timing bit.

EL unsigned int

Current Exception level field.

EXLOCK unsigned int

Exception return state lock bit.

IL unsigned int

Illegal Execution state bit.

NZCV unsigned int

Condition flags.

PACM unsigned int

PACM bit.

PAN unsigned int

Privileged Access Never state bit.

PM unsigned int

PMU exception mask bit.

PPEND unsigned int

PMU exception pending bit.

SM unsigned int

StreamingMode status bit.

SP unsigned int

Stack pointer register selection bit.

SS unsigned int

Software Step bit.

SSBS unsigned int

Speculative Store Bypass Safe bit.

TCO unsigned int

Tag Check Override bit.

TOUCHED unsigned int

Mask of the fields that have been touched since this trace has last been emitted. Note that touched does not necessarily mean the value changed. The index of the field is the bit index in the mask plus the index of this field plus one.

UAO unsigned int

User Access Override bit.

UINJ unsigned int

Undefined Instruction exceptions bit.

ZA unsigned int

ZA storage and ZT0 register enable bit.

nRW unsigned int

Current Execution state bit.

PSTATE_PACTIVE_OPMODE_TRANSITION

Trace pactive and Opmode mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Operating mode after the transition.

OLD_STATE enum

Operating mode before the transition.

PSTATE_PACTIVE_OPMODE_TRANSITION_DENIED

Trace the denied pactive and Opmode mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Operating mode after the transition.

OLD_STATE enum

Operating mode before the transition.

REASON string

Reason for the transition being denied.

PSTATE_PACTIVE_POWER_TRANSITION

Trace pactive and power mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Power mode after the transition.

OLD_STATE enum

Power mode before the transition.

PSTATE_PACTIVE_POWER_TRANSITION_DENIED

Trace the denied pactive and power mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Power mode after the transition.

OLD_STATE enum

Power mode before the transition.

REASON string

Reason for the transition being denied.

PSTATE_PACTIVE_THREAD_OPMODE_TRANSITION

Trace pactive and Opmode mode transitions for threads on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Operating mode after the transition.

OLD_STATE enum

Operating mode before the transition.

QUANTUM_START

Called at the start of every quantum.

RUN_STATE

Run state transition. Fields:

INST_COUNT **unsigned int**

Ticks count at point of transition.

NEW **enum**

New run state.

OLD **enum**

Old run state.

SEMIHOSTING_CALL

Call of a semihost function occurred. Fields:

PC **unsigned int**

The program counter after the semihosting call.

SEMIHOSTING_PRECALL

About to call semihost. Fields:

EL2 **bool**

Use EL2 translation regime.

NS **bool**

Is Non-Secure.

REG_WIDTH **unsigned int**

The current register width in bytes.

SIGNAL

External signal state change. Fields:

SIGNAL **enum**

Signal that changed.

STATE **bool**

Signal asserted state.

SPSR

SPSR change. Fields:

CORE_NUM **unsigned int**

Core number in a multi processor.

MODE **enum**

Which of the banked SPSR registers is written.

OLD_VALUE **unsigned int**

The old SPSR value.

VALUE unsigned int

The new SPSR value.

START_COMPILE

Compilation started. Fields:

VADDR unsigned int

Instruction where compilation begins.

STG_MULTIPLE_MICRO_OPS_ADDRESS_RANGE

Address range for MTE stores with multiple uops. Fields:

ADDR_END unsigned int

Final address.

ADDR_START unsigned int

Starting address.

STORE_MULTIPLE_REGS_ADDRESS_RANGE

Address range for stores with multiple registers. Fields:

ADDR_END unsigned int

Final address.

ADDR_START unsigned int

Starting address.

SUBOPTIMAL_LDST_RETIRED

suboptimal load/store. Fields:

ACCESS_SIZE unsigned int

Log2 of the access width in bytes used for alignment checking (1: half-word, 2: word ...).

OPTIMAL_ALIGN_SELECTOR unsigned int

Selector to decide the optimal alignment size (1: access width, 2: 4-bytes, 3: 8-bytes, ..., 12: 4-kbytes).

VADDR unsigned int

The virtual address of the access.

SYNC

Called for every synchronization, that is every quantum break. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The instruction count of this CPU.

LOCAL_QUANTUM unsigned int

The local quantum of this CPU.

LOCAL_TIME unsigned int

The local time of this CPU.

SYNC_REASON enum

Reason for sync.

SYSCALL

System call instruction executed. Fields:

IMM unsigned int

Immediate value of the system call instruction.

TYPE enum

System call type.

VADDR unsigned int

Instruction that caused the system call.

SYSREG_READ128

System Coprocessor 128 bit register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

REGNUM enum

Internal register number.

REGNUM_PHYS enum

Internal register number physically accessed (e.g. in presense of VHE redirection).

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_READ64

System Coprocessor register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REGNUM enum

Internal register number.

REGNUM_OPERAND enum

Register number corresponding to the instruction operand.

REGNUM_PHYS enum

Internal register number physically accessed (e.g. in presense of VHE redirection).

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_UPDATE128

Triggers when the system updates a register. Fields:

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG enum

Register number.

UNKNOWN unsigned int

Bits of the register which became unknown.

VALUE unsigned int

Value written to the register.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_UPDATE32

Triggers when the system updates a register. Fields:

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG enum

Register number.

UNKNOWN unsigned int

Bits of the register which became unknown.

VALUE unsigned int

Value written to the register.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_UPDATE64

Triggers when the system updates a register. Fields:

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG enum

Register number.

UNKNOWN unsigned int

Bits of the register which became unknown.

VALUE unsigned int

Value written to the register.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_WRITE128

System Coprocessor 128 bit register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

REGNUM enum

Internal register number.

REGNUM_PHYS enum

Internal register number physically accessed (e.g. in presense of VHE redirection).

UNDEF bool

The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_WRITE64

System Coprocessor register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REGNUM enum

Internal register number.

REGNUM_OPERAND enum

Register number corresponding to the instruction operand.

REGNUM_PHYS enum

Internal register number physically accessed (e.g. in presense of VHE redirection).

UNDEF bool

The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

UNALIGNED_LDST_RETIRED

Processor unaligned load/store. Fields:

ACCESS_SIZE unsigned int

Log2 of the access width in bytes used for alignment checking (1=>half-word, 2->word ...).

NUMBER_OF_BEATS unsigned int

Number of accesses (beats) in this burst (Total size in bytes is ACCESS_SIZE * NUMBER_OF_BEATS).

VADDR unsigned int

The virtual address of the access.

VFP_D_REGS

VFP/NEON D 64 bit register write. Fields:

ALIASING enum

Type of register aliasing used.

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a write to the corresponding bit in the register. For example, 0x1 would indicate a write to VALUE[0].

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

VFP_D_REGS_READ

VFP/NEON D 64 bit register read. Fields:

ID unsigned int

The register number.

VALUE unsigned int

The current value of the register.

VFP_Q_REGS

VFP/NEON Q 128 bit register write. Fields:

ALIASING enum

Type of register aliasing used.

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a write to the corresponding byte in the register. For example, 0x1 would indicate a write to VALUE[7:0].

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

VFP_Q_REGS_READ

VFP/NEON Q 128 bit register read. Fields:

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a read to the corresponding byte in the register. For example, 0x1 would indicate a read to VALUE[7:0].

VALUE unsigned int

The current value of the register.

VFP_SYS_REGS

Writes to the VFP/NEON units system registers. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ID enum

Which VFP system register is written.

MASK unsigned int

The mask indicating updated cumulative bits.

OLD_VALUE unsigned int

The register's old value overwritten.

VALUE unsigned int

The new value written to the VFP system register.

VFP_S_REGS

VFP/NEON S 32 bit register write. Fields:

ALIASING enum

Type of register aliasing used.

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a write to the corresponding bit in the register. For example, 0x1 would indicate a write to VALUE[0].

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

VFP_S_REGS_READ

VFP/NEON S 32 bit register read. Fields:

ID **unsigned int**

The register number.

VALUE **unsigned int**

The current value of the register.

WAYPOINT

Signals end of basic block execution. Fields:

CORE_NUM **unsigned int**

Core number in a multi processor.

ISET **enum**

Origin instruction set.

IS_COND **bool**

Indicates if this is a conditional waypoint.

PC **unsigned int**

Origin address (or 0 if unavailable).

TAKEN **bool**

Indicates if this waypoint was taken.

TARGET **unsigned int**

Destination address.

TARGET_ISET **enum**

Destination instruction set.

WFE_END

WFE ended. Fields:

INST_COUNT **unsigned int**

Ticks count when leaving WFE.

WFE_EVENT_REGISTER

WFE event register status: set/clear, reason. Fields:

INST_COUNT **unsigned int**

Ticks count.

REASON **enum**

Reason for set/clear. Only REASON==0 (Cleared by WFE) clears the bit, all other reasons set it.

WFE_IGNORED

WFE ignored. Fields:

EVENT bool

This WFE was ignored because the event register was set.

INST_COUNT unsigned int

Ticks count when ignoring WFE.

TRAPPED bool

This WFE was trapped.

WFE_START

WFE entered. Fields:

INST_COUNT unsigned int

Ticks count when entering WFE.

WFI_END

WFI ended. Fields:

INST_COUNT unsigned int

Ticks count when leaving WFI.

WFI_IGNORED

WFI ignored. Fields:

DISABLED bool

This WFI was ignored because WFI is disabled.

INST_COUNT unsigned int

Ticks count when ignoring WFI.

REASON enum

specifies reason why WFI trace was ignored.

TRAPPED bool

This WFI was trapped.

WFI_START

WFI entered. Fields:

INST_COUNT unsigned int

Ticks count when entering WFI.

WFI_WAKEUP

WFI wakeup. Fields:

INST_COUNT unsigned int

Ticks count when WFI wakeup occurred.

REASON enum

Reason for wakeup.

2.51 ARM_Cortex-R52Plus

This section describes the trace sources.

AA64_ASE_SVE_REGS

Modification of AArch64 Advanced SIMD or SVE Vector Registers (via SIMD or Scalar operations), SVE Predicate Registers and ZT0. Fields:

ID **enum**

The register identifier.

MASK **unsigned int**

Bitmask of the register to signal the modified bits in the VALUE field.

SM **bool**

Whether the PE is in Streaming Mode.

VALUE **unsigned int**

Value of AArch64 vector register.

ASYNC_MEMORY_FAULT

Context ID Register write. Fields:

FAULT **unsigned int**

Fault status present in ESR.ISS encoding for Serror in ARCH64 and present in DFSR encoding for external abort format in ARCH32.

PADDR **unsigned int**

Physical Address (or 0 if unavailable).

VADDR **signed int**

Virtual Address (or 0 if unavailable).

ATOMIC_END_ACCESS

Bus trace access for atomics. Fields:

ACCESS_FAIL **bool**

Memory access failed.

ADDR **unsigned int**

The virtual address of the access.

ATTR **unsigned int**

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

COMPARE_VALUE **unsigned int**

Compare value for CAS.

LOAD_VALUE unsigned int

Loaded values.

NSDESC unsigned int

The security state of the access.

OPERAND_VALUE unsigned int

Operation's operand.

OPERATION enum

Operation type.

PADDR unsigned int

The physical address of the access.

PRIV bool

Is this a privileged access?.

ATOMIC_SLAVE_ACCESS

Information on the atomic operation at slave side. Fields:

ACCESS_RESULT unsigned int

Result of atomic access.

COMPARE_VALUE unsigned int

Value to be compared with memory by CAS.

LOAD_VALUE unsigned int

Value to be loaded from memory as a result of atomic operation.

MANAGER unsigned int

ID of manager which initiated the transaction.

NS bool

Transaction is Non-Secure.

OPERAND_VALUE unsigned int

Incoming operand value provided by core.

OPERATION enum

Type of atomic operation.

PADDR unsigned int

Full physical address of TCM.

STORE_VALUE unsigned int

Value to be stored to memory as a result of atomic operation.

ATOMIC_START_ACCESS

Bus trace access for atomics. Fields:

ADDR unsigned int

The virtual address of the access.

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

COMPARE_VALUE unsigned int

Compare value for CAS.

NSDESC unsigned int

The security state of the access.

OPERAND_VALUE unsigned int

Operation's operand.

OPERATION enum

Operation type.

PADDR unsigned int

The physical address of the access.

PRIV bool

Is this a privileged access?.

ArchMsg.Error.exit_code

DISPLAY exit code: %{EXITCODE} component kind: %{KIND} reason: %{REASON}. Fields:

EXITCODE signed int

Exit code to be returned.

KIND string

Component kind that invoked the exit code trace.

REASON string

Reason why the exit code trace was invoked (optional).

ArchMsg.Info.invalid_region_programmed

DISPLAY MPU Stage %{STAGE:d}; Region ID %{REGION_ID:d} has base address greater than end address Base Addr=%{BASE_ADDR:x}, End Addr=%{END_ADDR:x}. Fields:

BASE_ADDR unsigned int

Base Address.

END_ADDR unsigned int

End Address.

REGION_ID unsigned int

Region ID.

STAGE unsigned int

MPU Stage.

ArchMsg.Warning.IgnoredWarmResetRequest

Event fires when software attempts to request warm reset through write to deprecated DBGPRCR.CWRR.

ArchMsg.Warning.ReportedRevisionRegisterRead

DISPLAY Reported revision and patch do not match that of the cpu. Fields:

ORIGINAL_VALUE unsigned int

The value the register would have based on the modelled revision.

REGISTER string

The name of the register that has ben read.

REPORTED_VALUE unsigned int

The value the register has been overridden to.

ArchMsg.Warning.Unpredictably Indexed PM Event Register

TAGS **UNPREDICTABLE** DISPLAY Event %{IsDirect:(Selector value|Register)} %{SEL} is >= %{N}. Fields:

IsDirect unsigned int

Direct Access rather than indirect.

N unsigned int

Number of Accessible Registers.

SEL unsigned int

Selector.

ArchMsg.Warning.cache_contents_unknown

DISPLAY %{SIDE:(D|I)}-Cache was enabled but was not invalidated since power-on: cache lines could contain **UNKNOWN** content. Fields:

REGIME unsigned int

Current translation regime.

SIDE unsigned int

0 d-side cache or unified cache, 1 i-side.

ArchMsg.Warning.dcimvac_matches_watchpoint

TAGS **UNPREDICTABLE** DISPLAY Watchpoints matching an AArch32 DCIMVA are implementation defined.

ArchMsg.Warning.decode_fieldmismatch

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with different fields %{FIELD1} and %{FIELD2}. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_frachitsnegative

TAGS **UNPREDICTABLE** DISPLAY VCVT Instruction is unpredictable with a negative number of fraction bits.

ArchMsg.Warning.decode_initblock

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable in an IT block.

ArchMsg.Warning.decode_initblocknotlast

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable in an IT block when not the last.

ArchMsg.Warning.decode_invalid_condition

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable when the condition is not AL.

ArchMsg.Warning.decode_invalidfieldcombination

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with the values in %{FIELD1} and %{FIELD2}. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_invalidvalue

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with the value in %{FIELD}. Fields:

FIELD enum

Field name.

ArchMsg.Warning.decode_mem_hint_unallocated

TAGS **UNPREDICTABLE** DISPLAY There is no memory hint allocated to this bit pattern.

ArchMsg.Warning.decode_registermatch

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with %{FIELD1} and %{FIELD2} as the same register. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_registermismatch

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with %{FIELD1} and %{FIELD2} as different registers. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_registeroutofrange

TAGS UNPREDICTABLE DISPLAY Floating-point instruction is unpredictable indexing beyond the end of the register bank.

ArchMsg.Warning.decode_sbzsbo

TAGS UNPREDICTABLE DISPLAY Reserved bits in the instruction are not canonical.

ArchMsg.Warning.decode_transactiontoobig

TAGS UNPREDICTABLE DISPLAY Attempt to perform a memory transaction of more than 128 bytes.

ArchMsg.Warning.decode_unpred_other

TAGS UNPREDICTABLE DISPLAY Instruction is unpredictable.

ArchMsg.Warning.decode_unpreduseofpc

TAGS UNPREDICTABLE DISPLAY Instruction is unpredictable when %{FIELD} is PC. Fields:

FIELD enum

Field name.

ArchMsg.Warning.decode_unpreduseofr13

TAGS UNPREDICTABLE DISPLAY Instruction is unpredictable when %{FIELD} is R13. Fields:

FIELD enum

Field name.

ArchMsg.Warning.decode_unpreduseofr13inreglist

TAGS UNPREDICTABLE DISPLAY Instruction is unpredictable with R13 in register-list.

ArchMsg.Warning.decode_useofd16orhigher

TAGS UNPREDICTABLE DISPLAY Attempt to use register D16 or higher where it is not allowed.

ArchMsg.Warning.decode_writebackandbaseinlist

TAGS UNPREDICTABLE DISPLAY Instruction is unpredictable with write back when the base register is in the transfer list.

ArchMsg.Warning.decode_zeroregistersinlist

TAGS UNPREDICTABLE DISPLAY Instruction is unpredictable when no registers are to be transferred.

ArchMsg.Warning.recursive_branch

An instruction is performing a branch that targets the same instruction. If this was intended then a WFI might be more effective DISPLAY recursive branch detected at PC=%{PC}. PRIMARY KEY PC. Fields:

PC unsigned int

Address of instruction causing the branch.

ArchMsg.Warning.recursive_exception

An instruction has generated an exception that targets the same instruction. DISPLAY recursive %{TYPE} exception detected at PC=%{PC} (CPSR=%{CPSR}). PRIMARY KEY PC. Fields:

CPSR unsigned int

Processor state.

PC unsigned int

Address of instruction causing the exception.

TYPE enum

Type of exception.

ArchMsg.Warning.reentrant_vector_catch

TAGS **UNPREDICTABLE** DISPLAY Vector catch debug event on Prefetch or Data abort vector %**{ADDR}**. Fields:

ADDR unsigned int

vector address.

ArchMsg.Warning.reserved_it_state

TAGS **UNPREDICTABLE** DISPLAY Execution with reserved IT state %**{ITSTATE}**. Fields:

ITSTATE unsigned int

reserved IT state.

ArchMsg.Warning.tlb_contents_unknown

Translation has been enabled without invalidating the TLB. DISPLAY Translation was enabled in the %**{PAS:(Secure|Non-secure|Root|Realm)}** EL%**{EL:d}** translation regime (vmid= %**{VMID:x}**) without invalidating %**{INVALIDITY:(|d-side |i-side |)}**entries since power-on: %**{INVALIDITY:(|D||)}**TLB could contain **UNKNOWN** content. Fields:

EL unsigned int

Exception level of translation regime.

INVALIDITY unsigned int

Which TLB may be invalid (if TLBs are separate).

NON_SECURE bool

Whether the translation regime is non-secure.

PAS unsigned int

Physical address space of translation regime.

VMID unsigned int

Current VMID, if applicable.

ArchMsg.Warning.unknown_DLR_DSPSR

Exiting debug state with unknown DLR or DSPSR.

ArchMsg.Warning.unknown_ELR_SPSR

Returning from debug exception with unknown ELR or SPSR.

ArchMsg.Warning.unpred_FAR_on_LRCPC3_non_atomic_mem_access

TAGS **UNPREDICTABLE** Unpredictable FAR/FSTATUS on LRCPC3 pair instructions resulting in two separate memory accessesDISPLAY Unpredictable FAR on LRCPC3 non atomic

memory access at %{ADDR1} with FSTATUS %{FSTATUS1} or %{ADDR2} with FSTATUS %{FSTATUS2}. Fields:

ADDR1 unsigned int

Possible FAR address 1.

ADDR2 unsigned int

Possible FAR address 2.

FSTATUS1 unsigned int

Possible fault status value 1.

FSTATUS2 unsigned int

Possible fault status value 2.

ArchMsg.Warning.unpred_async_tag_check_fail_on_sve_speculative_load_do_update_ffr

Unpredictable set to FALSE of the FFR element associated to SVE speculative load with an async tag check fail TAGS **UNPREDICTABLE**.

ArchMsg.Warning.unpredictable_a32_breakpoint

TAGS **UNPREDICTABLE** DISPLAY Unpredictable breakpoint on a A32 instruction at %{ADDR1} Offset %{OFFSET} BAS %{BAS} for a %{IS_ADDRESS_MISMATCH_BREAKPOINT:(Mismatch|Match)} Breakpoint number %{N:d} Breakpoint address %{ADDR2}. Fields:

ADDR1 unsigned int

Address.

ADDR2 unsigned int

Breakpoint address.

BAS unsigned int

BAS field.

IS_ADDRESS_MISMATCH_BREAKPOINT bool

Is a Mismatch breakpoint, not a Match breakpoint.

N unsigned int

Breakpoint number.

OFFSET unsigned int

Always has a value of zero.

ArchMsg.Warning.unpredictable_banked_register_access

TAGS **UNPREDICTABLE** DISPLAY Access to banked register (R=%{R:d} SYSm=%{SYSM:2x}) is unpredictable. Fields:

R unsigned int

R bit.

SYSM unsigned int

SYSm.

ArchMsg.Warning.unpredictable_ccfail_undef

TAGS **UNPREDICTABLE** DISPLAY An instruction %{OPCODE} at VA=%{VADDR} failed its condition codes check but would otherwise have caused a non-data-dependent trap or undefined exception. Behaviour is implementation-defined choice of exception or nop. Fields:

OPCODE unsigned int

opcode of the instruction.

VADDR unsigned int

Virtual address of the instruction being executed.

ArchMsg.Warning.unpredictable_ici_bits

TAGS **UNPREDICTABLE** DISPLAY ICI bits do not apply or are architecturally unpredictable for instruction.

ArchMsg.Warning.unpredictable_t32_breakpoint

TAGS **UNPREDICTABLE** DISPLAY Unpredictable breakpoint on a T32 instruction at %{ADDR1} Offset %{OFFSET} BAS %{BAS} for a %{IS_ADDRESS_MISMATCH_BREAKPOINT:(Mismatch|Match)} Breakpoint number %{N:d} Breakpoint address %{ADDR2}. Fields:

ADDR1 unsigned int

Address.

ADDR2 unsigned int

Breakpoint address.

BAS unsigned int

BAS field.

IS_ADDRESS_MISMATCH_BREAKPOINT bool

Is a Mismatch breakpoint, not a Match breakpoint.

N unsigned int

Breakpoint number.

OFFSET unsigned int

Indicates which halfword of a 32-bit Thumb instruction the breakpoint is set on. 0 for the first halfword or if the instruction is 16 bits wide, or 1 for the second halfword.

ArchMsg.Warning.unpredictable_t32_vector_catch

TAGS **UNPREDICTABLE** DISPLAY Unpredictable Vector catch exception on a T32 instruction at %{ADDR} Offset %{OFFSET}. Fields:

ADDR unsigned int

Address.

OFFSET unsigned int

Indicates which halfword of a 32-bit Thumb instruction the vector catch exception is set on. 0 for the first halfword or if the instruction is 16 bits wide, or 1 for the second halfword.

ArchMsg.Warning.unpredictable_watchpoint_far

TAGS **UNPREDICTABLE** DISPLAY The value for FAR/EDWAR for hitting this watchpoint may be anywhere in the range %{LOWER_BOUND} to %{UPPER_BOUND}. Fields:

LOWER_BOUND unsigned int

the lowest address accessed by the instruction that triggered the watchpoint.

UPPER_BOUND unsigned int

the highest watchpointed address accessed by that instruction.

ArchMsg.Warning.unpredictable_watchpoint_far_isscmp

DISPLAY The value for FAR/EDWAR for hitting this watchpoint may be anywhere in the range %{LOWER_BOUND} to %{UPPER_BOUND}. Fields:

LOWER_BOUND unsigned int

the lowest address accessed by the instruction that triggered the watchpoint.

UPPER_BOUND unsigned int

the highest watchpointed address accessed by that instruction.

ArchMsg.Warning.warning_AdvSIMDExpandImmUnexpectedZero

TAGS **UNPREDICTABLE** DISPLAY AdvSIMDExpandImm may treat this immediate value as **UNPREDICTABLE**.

ArchMsg.Warning.warning_ConditionalSMC

TAGS **UNPREDICTABLE** DISPLAY SMC instruction has **UNPREDICTABLE** effects when conditional and when combined with traps.

ArchMsg.Warning.warning_TLBID_out_of_range

TLBID out of range in TLBI according to the configuration of NIS, NVIS, NOS and NVOS. Fields:

NIS unsigned int

NIS.

NOS unsigned int

NOS.

NVIS unsigned int

NVIS.

NVOS unsigned int

NVOS.

TLBID unsigned int

The TLBID value used in the TLBI instruction.

ArchMsg.Warning.warning_TTL_or_TG_mismatches_current_translation_granule

TTL or TG mismatches current translation granule in TLBI VA/RVA/IPA/RIPA. Fields:

CURRENT_TG unsigned int

Current TG as indicated by TCR_ELx or VTCR_EL2.

TLBI_TG unsigned int

TG (or TTL[3:2]) used in TLBI instruction.

ArchMsg.Warning.warning_access_crosses_page_boundary

TAGS **UNPREDICTABLE** DISPLAY Access crosses page boundary.

ArchMsg.Warning.warning_access_crosses_page_boundary_has_fault

TAGS **UNPREDICTABLE** DISPLAY Access that crosses page boundary has fault(s).

ArchMsg.Warning.warning_access_crosses_page_boundary_has_fault_on_both_pages

TAGS **UNPREDICTABLE** DISPLAY Access that crosses page boundary has a fault on each page.

Fields:

ADDR1 unsigned int

FAR address on first page.

ADDR2 unsigned int

FAR address on second page.

FSTATUS1 unsigned int

Fault status on first page.

FSTATUS2 unsigned int

Fault status on second page.

ArchMsg.Warning.warning_access_crosses_page_boundary_spanning_different_memory

TAGS **UNPREDICTABLE** DISPLAY Access at address %{ADDR} crosses page from MEM_TYPE %{MEMTYPE_PAGE1} to %{MEMTYPE_PAGE2}. Fields:

ADDR unsigned int

address of access crossing page.

MEMTYPE_PAGE1 signed int

First Page's memory type (0-Normal, 1-Non-Normal).

MEMTYPE_PAGE2 signed int

Second Page's memory type (0-Normal, 1-Non-Normal).

ArchMsg.Warning.warning_access_wraps_around_memory

TAGS **UNPREDICTABLE** DISPLAY Access wraps around memory in %{WIDTH:d}bit mode.

Fields:

WIDTH unsigned int

Address width.

ArchMsg.Warning.warning_bcr_linking_status

TAGS **UNPREDICTABLE** DISPLAY DBGBCR.{BT2,BT} combination gives unpredictable behaviour.

ArchMsg.Warning.warning_bcr_mask_reserved

TAGS **UNPREDICTABLE** DISPLAY DBGBCR_ELn.MASK is set to a reserved value.

ArchMsg.Warning.warning_bvr_masked_bit_not_zero

TAGS **UNPREDICTABLE** DISPLAY DBGBVRn_EL1.VA contains a bit masked by DBGBCRn_EL1.MASK that is not zero.

ArchMsg.Warning.warning_ccsidr_unimplemented_level

TAGS **UNPREDICTABLE** DISPLAY CCSIDR read while CSSELR %{CSSELR:x} points to an unimplemented cache level. Fields:

CSSELR unsigned int

current effective value of CSSELR.

ArchMsg.Warning.warning_change_to_ns_when_tge_set

TAGS **UNPREDICTABLE** DISPLAY Attempting to change NS to 1 when HCR.TGE = 1.

ArchMsg.Warning.warning_contiguous_bit_check_abort

TAGS **UNPREDICTABLE** DISPLAY An abort occurred whilst attempting to check TLB entry is contiguous at %{ENTRY_ADDR}. Fields:

ENTRY_ADDR unsigned int

address of conflicting TLB entry.

ArchMsg.Warning.warning_contiguous_bit_error

TAGS **UNPREDICTABLE** DISPLAY TLB entry at %{ENTRY_ADDR} was not contiguous with entry at %{CONTIG_ADDR} Contents are %{ENTRY_DATA} but expected %{CONTIG_DATA}. Fields:

CONTIG_ADDR unsigned int

address of the TLB first read, with which this entry is expected to be contiguous.

CONTIG_DATA unsigned int

expected contents based on the entry at CONTIG_ADDR.

ENTRY_ADDR unsigned int

address of conflicting TLB entry.

ENTRY_DATA unsigned int

contents of conflicting TLB entry.

ArchMsg.Warning.warning_contiguous_bit_error_D128

TAGS **UNPREDICTABLE** DISPLAY TLB entry at %{ENTRY_ADDR} was not contiguous with entry at %{CONTIG_ADDR} Contents are %{ENTRY_DATA_HI} %{ENTRY_DATA_LO} but expected %{CONTIG_DATA}. Fields:

CONTIG_ADDR unsigned int

address of the TLB first read, with which this entry is expected to be contiguous.

CONTIG_DATA unsigned int

expected contents based on the entry at CONTIG_ADDR.

ENTRY_ADDR unsigned int

address of conflicting TLB entry.

ENTRY_DATA_HI unsigned int

Higher bit contents of conflicting TLB entry.

ENTRY_DATA_LO unsigned int

Lower bit contents of conflicting TLB entry.

ArchMsg.Warning.warning_cp10_cp11_mismatch

TAGS **UNPREDICTABLE** DISPLAY Unpredictable configuration of CP10 and CP11 controls at EL **%{EL:d}**. Fields:

EL unsigned int

EL owning the control.

ArchMsg.Warning.warning_cp10_cp11_reserved_value

TAGS **UNPREDICTABLE** DISPLAY Unpredictable configuration of CPACR CP10 (**%{CP10}**) or CP11 (**%{CP11}**). Fields:

CP10 unsigned int

CP10 access.

CP11 unsigned int

CP11 access.

ArchMsg.Warning.warning_csselr_level_out_of_range

TAGS **UNPREDICTABLE** DISPLAY CSSELR.Level out of range, written value **%{LEVEL}** with only **%{IMPLEMENTED:d}** levels implemented. Fields:

IMPLEMENTED unsigned int

number of cache levels implemented.

LEVEL unsigned int

written value of CSSELR.Level.

TYPE enum

The type of cache selected by CSSELR.InD and TnD.

ArchMsg.Warning.warning_debug_flow_control_bits_not_obeyed

TAGS **UNPREDICTABLE** DISPLAY Flow control bits for **%{REGISTER}** were not obeyed, causing **%{ERROR}**. Fields:

ERROR enum

type of error resulting.

REGISTER enum

register under consideration.

ArchMsg.Warning.warning_debug_register_access_during_reset

TAGS **UNPREDICTABLE** DISPLAY Unpredictable **%{IS_WRITE:(read|write)}** access to debug register offset **%{OFFSET}** during reset. Fields:

IS_WRITE unsigned int

Write Not Read.

OFFSET unsigned int

Register Offset.

ArchMsg.Warning.warning_decode_cps_inconsistent_fields

TAGS **UNPREDICTABLE** DISPLAY The mode setting fields in the CPS instruction are inconsistent.

ArchMsg.Warning.warning_decode_invalid_state

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable in current exception-level/security state.

ArchMsg.Warning.warning_default_cacheable_mmu_on

TAGS **UNPREDICTABLE** DISPLAY Default cacheable is enabled (HCR.DC = 1) while MMU is enabled.

ArchMsg.Warning.warning_default_cacheable_vmmu_off

TAGS **UNPREDICTABLE** DISPLAY Default cacheable is enabled (HCR.DC = 1) while HCR.VM = 0.

ArchMsg.Warning.warning_deprecated_wvr_bit_2_set

TAGS **UNPREDICTABLE** DISPLAY DBGWVR%{N:d}_EL1=%{WVR:h} has bit 2 set, deprecated in ARMv8. Fields:

N unsigned int

index.

WVR unsigned int

DBGWVR.

ArchMsg.Warning.warning_eret_while_software_step_active_pending

Missing ISB between setting MDSCR_EL1.SS and ERET TAGS **UNPREDICTABLE**.

ArchMsg.Warning.warning_exclusive_to_non_normal

TAGS **UNPREDICTABLE** DISPLAY Exclusive Access to Non-Normal Memory %{ADDR}. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_exclusive_to_non_writeback

TAGS **UNPREDICTABLE** DISPLAY Exclusive Access to Non-Writeback-Cacheable Memory %{ADDR}. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_execute_from_device_memory

TAGS **UNPREDICTABLE** DISPLAY An attempt was made to execute from Device memory.

ArchMsg.Warning.warning_illegal_cpsr_mode

TAGS **UNPREDICTABLE** DISPLAY Writing an illegal or unimplement mode (%{NEW_MODE:h}) to CPSR was unpredictable. Fields:

NEW_MODE unsigned int

New value of CPSR.M.

ArchMsg.Warning.warning_illegal_srs_mode

TAGS **UNPREDICTABLE** DISPLAY Illegal or **UNPREDICTABLE** mode (%{MODE:h}) used for SRS instruction. Fields:

MODE unsigned int

mode for Banked SP.

ArchMsg.Warning.warning_implementation_defined_read_debug_register_in_SCS

Read Access to Debug registers from the processor is **IMPLEMENTATION DEFINED** TAGS **IMP_DEF**. Fields:

PPB_OFFSET unsigned int

debug register offset.

ArchMsg.Warning.warning_implementation_defined_sequential_security_transitions_supported

The behavior of sequential instruction fetches that cross from non-secure to secure memory and contain an SG instruction is **CONSTRAINED UNPREDICTABLE** TAGS **UNPREDICTABLE**.

ArchMsg.Warning.warning_implementation_defined_stack_limit_check_supported

It is **IMPLEMENTATION DEFINED** whether stack pointer limit checking is performed for this instruction TAGS **IMP_DEF**. Fields:

ADDRESS unsigned int

Address of instruction.

OPCODE unsigned int

instruction opcode.

ArchMsg.Warning.warning_implementation_defined_taggednonsharedmemorytype

TAGS **IMP_DEF** DISPLAY Memory attribute has Tagged, non shared memory type; it is **IMPLEMENTATION DEFINED** whether the memory location is treated as Tagged or non-Tagged.

ArchMsg.Warning.warning_implementation_defined_write_debug_register_in_SCS

Write Access to Debug registers from the processor is **IMPLEMENTATION DEFINED** TAGS **IMP_DEF**. Fields:

DATA unsigned int

data attempted to be written.

PPB_OFFSET unsigned int

debug register offset.

ArchMsg.Warning.warning_invalid_tcr_granule

TAGS **UNPREDICTABLE** DISPLAY TCR.TG%{TG_ID:(0|1)} bits have been set to a granule size %{REQUEST:d}K not implemented - using %{SUBSTITUTE:d}K. Fields:

REQUEST signed int

page size requested (or 0 for reserved).

SUBSTITUTE signed int

best guess available page size.

TG_ID bool

bits TG0 or TG1.

ArchMsg.Warning.warning_load_multiple_user_registers_from_user_mode

TAGS UNPREDICTABLE DISPLAY An LDM(user registers) instruction executed from user mode.

ArchMsg.Warning.warning_mte_in_mem_attr_not_supported

TAGS UNPREDICTABLE DISPLAY Memory attribute %{ATTR} has Tagged memory type but Armv8.5-MTE is not implemented. Fields:

ATTR unsigned int

Memory attribute.

ArchMsg.Warning.warning_reserved_breakpoint_state_match

TAGS UNPREDICTABLE DISPLAY %{IS_BREAKPOINT:(Watchpoint|Breakpoint)} programmed with a reserved combination of HMC, SSC and %{IS_BREAKPOINT:(PAC|PMC)}. Fields:

IS_BREAKPOINT unsigned int

Is a breakpoint, not a watchpoint.

ArchMsg.Warning.warning_shareability

TAGS UNPREDICTABLE DISPLAY Unpredictable: combination of the 1st and 2nd stages of translation is Normal Inner Non-Cacheable, Outer Non-Cacheable.

ArchMsg.Warning.warning_software_step_set_while_enabled

TAGS UNPREDICTABLE DISPLAY MDSCR_EL1.SS set to 1 while software step debug exceptions are enabled.

ArchMsg.Warning.warning_thumb_instruction_wraps_around_memory

TAGS UNPREDICTABLE DISPLAY Thumb instruction wraps around memory.

ArchMsg.Warning.warning_ttbr_sbz_bits_are_not_zero

TAGS UNPREDICTABLE DISPLAY %{TTBR}[%{MSB:d}:0] should be zero, but are not. Fields:

MSB unsigned int

x-1.

TTBR enum

which TTBR register.

ArchMsg.Warning.warning_unaligned_address_dbgdtrrx_write

TAGS UNPREDICTABLE DISPLAY Unaligned address %{ADDR} in X0/R0 when DBGDTRRX written in memory access mode. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unaligned_address_dbgdtrtx_read

TAGS **UNPREDICTABLE** DISPLAY Unaligned address %{ADDR} in XO/RO when DBGDTRTX read in memory access mode. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unaligned_to_device

TAGS **UNPREDICTABLE** DISPLAY Unaligned Access to Device Memory %{ADDR}. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unaligned_to_strongly_ordered

TAGS **UNPREDICTABLE** DISPLAY Unaligned Access to Strongly Ordered Memory %{ADDR}. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unknown_sau_rnr

TAGS **UNPREDICTABLE** DISPLAY SAU_RNR was set to an unsupported value.

ArchMsg.Warning.warning_unpredictable_AIRCR_PRIS_and_BFHFNMINs

TAGS **UNPREDICTABLE** DISPLAY The effect of setting both AIRCR.BFHFNMINs and AIRCR.PRIS to 1 is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_AIRCR_VECTCLRACTIVE_when_not_in_debug

TAGS **UNPREDICTABLE** DISPLAY The effect of writing a 1 to AIRCR.VECTCLRACTIVE if the processor is not halted in Debug state is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_AIRCR_VECTRESET_and_SYSRESETREQ

TAGS **UNPREDICTABLE** DISPLAY When the processor is halted in Debug state, if a write to the register writes a 1 to both VECTRESET and SYSRESETREQ, the behavior is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_AIRCR_VECTRESET_when_not_in_debug

TAGS **UNPREDICTABLE** DISPLAY The effect of writing a 1 to AIRCR.VECTRESET if the processor is not halted in Debug state is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_AIRCR_incorrect_VKEY

TAGS **UNPREDICTABLE** DISPLAY The value 0x05FA must be written to AIRCR.VKEY, otherwise the register write is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_EXC_RETURN_Reserved_Bit

EXC_RETURN[23:7] are reserved with the special condition that all bits should be written as one. Values other than all 1s are **UNPREDICTABLE**. TAGS **UNPREDICTABLE**. Fields:

EXC_RETURN unsigned int

Special PC value.

SBO_MASK unsigned int

Which bits Should Be One to not UNPRED.

ArchMsg.Warning.warning_unpredictable_PMEVTYPErn_ELO_VS_reserved_value

TAGS UNPREDICTABLE DISPLAY Unpredictable reserved value of PMEVTYPER<n>_ELO.VS.

ArchMsg.Warning.warning_unpredictable_change_to_DEMCR_MON_STEP_at_insufficient_priority

TAGS UNPREDICTABLE DISPLAY The effect of changing DEMCR.MON_STEP at an execution priority that is lower than the priority of the DebugMonitor exception is **CONSTRAINED UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_change_to_priority_of_active_exception

TAGS UNPREDICTABLE DISPLAY Changing the priority of an active exception is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_clear_lspact

TAGS UNPREDICTABLE DISPLAY **UNPREDICTABLE** state on context switch. Fields:

DRVEXC_TAKEN unsigned int

Derived exception taken, not pended.

REACHED_LAST_STORE unsigned int

Only the last faulted.

ArchMsg.Warning.warning_unpredictable_dbg_exit_unaligned_dlr

TAGS UNPREDICTABLE DISPLAY Debug exit to AArch32 with DLR[0] set to 1. PC[0] can be set to 0 or to DLR[0].

ArchMsg.Warning.warning_unpredictable_exception_catch

TAGS UNPREDICTABLE DISPLAY Generation of an exception catch event is unpredictable.

ArchMsg.Warning.warning_unpredictable_exception_return_inconsistent_state

TAGS UNPREDICTABLE DISPLAY state on exception return is unpredictable.

ArchMsg.Warning.warning_unpredictable_exception_return_instruction

TAGS UNPREDICTABLE DISPLAY exception return instruction is unpredictable.

ArchMsg.Warning.warning_unpredictable_in_debug_state

TAGS UNPREDICTABLE DISPLAY Instruction is **UNPREDICTABLE** when executed in debug state.

ArchMsg.Warning.warning_unpredictable_mair_encoding

TAGS UNPREDICTABLE DISPLAY Memory attribute %{ATTR} has invalid MAIR encoding; this may lead to unpredictable results. Fields:

ATTR unsigned int

Memory attribute.

ArchMsg.Warning.warning_unpredictable_pmu_counter_access

TAGS UNPREDICTABLE DISPLAY Access to PMU counters from non-secure ELO or EL1 is **UNPREDICTABLE** with MDCR_EL2.HPMN set to 0.

ArchMsg.Warning.warning_unpredictable_prioritization_breakpoint_match_vector_catch

TAGS UNPREDICTABLE DISPLAY Two events with the same priority occurred at the same instruction: Address Matching Vector Catch debug event, and Breakpoint debug event. It is constrained unpredictable which is taken.

ArchMsg.Warning.warning_unpredictable_stack_selection

TAGS **UNPREDICTABLE** DISPLAY **UNPREDICTABLE** state on context switch.

ArchMsg.Warning.warning_unpredictable_tsize_out_of_range

TAGS **UNPREDICTABLE** DISPLAY **%{TCR}.T%{TTBR:d}SZ (%{TSIZE:d} bits)** is out of range. Must be between 25 and **%{TSIZE_MAX:d}**. Fields:

TCR enum

Which TCR.

TSIZE signed int

Value of TSize.

TSIZE_MAX signed int

The maximum allowed value.

TTBR unsigned int

Which TTBR region.

ArchMsg.Warning.warning_unpredictable_unaligned_pc_as_base_register

TAGS **UNPREDICTABLE** DISPLAY The PC value must be word-aligned, otherwise the behavior of the instruction is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_unaligned_pop_stack

TAGS **UNPREDICTABLE** DISPLAY **CONSTRAINED UNPREDICTABLE** unaligned stack frame **%{FRAME_ADDR:x}** in exception-return. Fields:

FRAME_ADDR unsigned int

unaligned stack frame address.

ArchMsg.Warning.warning_unpredictable_unaligned_to_non_writeback

TAGS **UNPREDICTABLE** DISPLAY Unaligned Access to Non-Writeback-Cacheable Memory.

ArchMsg.Warning.warning_unpredictable_vmsa_memattrib

TAGS **UNPREDICTABLE** DISPLAY Unpredictable vmsa memory attribute with texcb value programmed to **%{TEXCB}** when tex remap is **%{TEX_REMAP}**. Fields:

TEXCB unsigned int

texcb value.

TEX_REMAP bool

use tex remap.

ArchMsg.Warning.warning_unpredictable_vtcr_t0sz_sl0

TAGS **UNPREDICTABLE** DISPLAY The combination of **VTCR.TOSZ=%{TOSZ:d}** and **VTCR.SL0=%{SL0:d}** is unpredictable. Fields:

SL0 signed int

VTCR.SL0, starting level for stage 2 translation table walks.

TOSZ signed int

VTCR.TOSZ, required input address range.

ArchMsg.Warning.warning_unpredictable_write_DHCSR

TAGS UNPREDICTABLE DISPLAY UNPREDICTABLE set of bit-changes in DHCSR. Fields:

NEW unsigned int

data attempted to be written.

OLD unsigned int

previous value.

ArchMsg.Warning.warning_unpredictable_write_SHCSR

TAGS UNPREDICTABLE DISPLAY UNPREDICTABLE setting a pending bit to 1 for an exception with priority >= execution priority.

ArchMsg.Warning.warning_unsupported_access_to_memory_mapped_register

TAGS UNPREDICTABLE DISPLAY Access to memory mapped register is unsupported.

ArchMsg.Warning.warning_user_jmcr_access

TAGS UNPREDICTABLE DISPLAY User %{WRITE:(write|read)} access to JMCR is unpredictable. Fields:

WRITE unsigned int

access is write.

ArchMsg.Warning.warning_wcr_mask_and_bas

TAGS UNPREDICTABLE DISPLAY DBGWCR_EL1.MASK is non-zero and BAS is not 0xff.

ArchMsg.Warning.warning_wcr_mask_reserved

TAGS UNPREDICTABLE DISPLAY DBGWCR_EL1.MASK is set to a reserved value.

ArchMsg.Warning.warning_wcr_non_configuous_bas

TAGS UNPREDICTABLE DISPLAY DBGWCR_EL1.BAS has non-contiguous set of ones.

ArchMsg.Warning.warning_wvr_masked_bit_not_zero

TAGS UNPREDICTABLE DISPLAY DBGWVRn_EL1.VA contains a bit masked by DBGWCRn_EL1.MASK that is not zero.

ArchMsg.Why.why_branch_target_exception

A branch target exception has occurred. DISPLAY branch target exception: target pc=%{PC}, target opcode=%{OPCODE}. PRIMARY KEY PC. Fields:

BTYPE unsigned int

pstate.BTYPE at time branch taken.

OPCODE unsigned int

opcode at branch target.

PC unsigned int

Virtual Address of branch target.

ArchMsg.Why.why_illegal_state

An illegal mode change has occurred. DISPLAY illegal mode change: %{MESSAGE}. Fields:

MESSAGE string

The message.

BRANCH_MISPREDICT

Simulating branch mispredict. Fields:

PC unsigned int

Origin address (or 0 if unavailable).

BRA_DIR

Direct branches, to immediate address. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISSET enum

The instruction set of the branch instruction.

IS_COND bool

Indicates if this is a conditional branch.

IS_HINTED bool

Whether the conditional instruction is hinted.

IS_ISB bool

Whether the instruction is ISB.

IS_LINK bool

Indicates if this is a branch with link.

PC unsigned int

The address of the branch instruction.

TARGET_ISET enum

The instruction set after the branch.

TARGET_PC unsigned int

The address the instruction branches to.

BRA_INDIR

Indirect branches, perhaps to a register. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISSET enum

The instruction set of the branch instruction.

IS_COND bool

Indicates if this is a conditional branch.

IS_HINTED bool

Whether the conditional instruction is hinted.

IS_ISB bool

Whether the instruction is ISB.

IS_LINK bool

Indicates if this is a branch with link.

PC unsigned int

The address of the branch instruction.

TARGET_ISET enum

The instruction set after the branch.

TARGET_PC unsigned int

The address the instruction branches to.

BRA_INDIR_CCFAIL

Indirect branches fail, perhaps to a register.

BRA_RET_CCFAIL

Branch return fail, perhaps to a register.

CACHE_MAINTENANCE_OP

Cache Maintenance Operation. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

DATA unsigned int

Specified MVA or Set/way.

FUNCTION enum

Clean / Invalidate.

SCOPE enum

Affected region.

SIDE enum

Inst / Data.

CCFAIL

Conditional instruction condition check fail. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CCFAIL_UNC

Unconditional instruction condition check fail. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CCPASS

Conditional instruction condition check pass. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CCPASS_UNC

Unconditional instruction condition check pass. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CODE_CACHE_MAINT

Code cache maintenance. Fields:

END_INCL unsigned int

Inclusive end address for VA or PA requests.

NS bool

Non-secure world for PA.

START unsigned int

Start address for VA or PA requests.

TYPE enum

Request type.

COMPILE_BLOCK_END

Last instruction of basic block translated. Fields:

VADDR unsigned int

Address of next instruction after this basic block.

COMPILE_BLOCK_START

First instruction of basic block translated. Fields:

VADDR unsigned int

Address of first instruction in basic block.

COMPILE_INST

ARM instruction compiled. Fields:

DISASS string

Disassembly of the instruction.

ISSET enum

The instruction set of this instruction.

ITSTATE unsigned int

The ITSTATE current for the instruction.

OPCODE unsigned int

The opcode of the instruction.

PC unsigned int

The address of the instruction.

SIZE unsigned int

The size of the instruction in bytes.

CONTEXTIDR

Context ID Register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

NS bool

Secure or nonsecure banked register is accessed.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The new value written.

CONTEXT_SYNC

Called for every context synchronization event. Fields:

ARCH_CSE bool

Architectural Context Synchronization Event.

PC unsigned int

The address of the synchronization instruction.

SYSREG_SYNC bool

System register synchronization event.

CORE_ENDIAN

Core BE8 Big-Endian state changed. Fields:

BE8 bool

Core BE8 Big-Endian state.

CORE_INFO

Static processor attributes. Only triggered by a call to DumpState(). Fields:

ARCH_PROFILE enum

The architecture profile of the core.

CLUSTER_ID unsigned int

The cluster ID of this processor.

CORE_NUM unsigned int

The number of this core in an MP processor.

DETAILED_CORE_FEATURES enum

Detailed core features as a bitfield. Enum values are bitmasks to extract the info from the field.

FPU_VERSION enum

The VFP version implemented by the core.

MEM_ARCH enum

The memory architecture of the core.

MIDR_EL1 unsigned int

MIDR_EL1 value.

NUM_CORES unsigned int

The number of cores in this MP processor.

QUANTUM_SIZE unsigned int

The default quantum size of the core (in ticks).

SECURITY_FEATURES bool

Does the core have security features?.

CORE_LOADS

Processor load accesses. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ACQREL enum

Is this an acquire/release.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DATA unsigned int

The data read or written.

ELEMENT_SIZE unsigned int

Width of each element.

ENDIANNESS enum

Byte ordering of this access.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

LOCK enum

Normal, exclusive or locked access.

MECID unsigned int

Memory Encryption Context ID, if applicable.

MECID2 unsigned int

Memory Encryption Context ID of the second page, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PAGE_SIZE unsigned int

Page size as $\log_2(\text{size in bytes})$.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK, 2=Exclusive Failed, 3=Unsupported.

SIZE unsigned int

Width of the access in bytes. Only required if DATA is not traced.

TRANS bool

Is this a translated access.

VADDR unsigned int

The virtual address of the access.

CORE_REGS

Changes of the core registers R0 to R14. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number, 0 to 14.

MODE enum

Bank of the register accessed.

OLD_VALUE unsigned int

The old value overwritten.

PHYS_ID enum

The physical register accessed.

VALUE unsigned int

The new value written to the register.

CORE_REGS64

Changes of the core registers X0..X30, SP_ELn. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ID enum

The register number, 0-30 for X0-X30, ≥ 32 for SP_ELn.

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

CORE_REGS64_READ

General purpose AArch64 register has been read X0..X30, SP_ELn. Fields:

ID enum

The register number, 0-30 for X0-X30, >=32 for SP_ELn.

VALUE unsigned int

The value read from the register.

CORE_STALLED_TRANSITION

Detect when a core is prevented from progressing PChannel states. Fields:

CURRENT_STATE enum

Current power mode of core.

REASON string

The reason a transition was stalled, and what must be done to unblock it.

TARGET_STATE enum

Desired power mode of core after transition.

CORE_STORES

Processor store accesses. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ACQREL enum

Is this an acquire/release.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DATA unsigned int

The data read or written.

ELEMENT_SIZE unsigned int

Width of each element.

ENDIANNESS enum

Byte ordering of this access.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

LOCK enum

Normal, exclusive or locked access.

MECID unsigned int

Memory Encryption Context ID, if applicable.

MECID2 unsigned int

Memory Encryption Context ID of the second page, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PAGE_SIZE unsigned int

Page size as $\log_2(\text{size in bytes})$.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK, 2=Exclusive Failed, 3=Unsupported.

SIZE unsigned int

Width of the access in bytes. Only required if DATA is not traced.

TRANS bool

Is this a translated access.

VADDR unsigned int

The virtual address of the access.

CP14_READ

System Coprocessor register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP14 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc1 unsigned int

opcode 1.

opc2 unsigned int

opcode 2.

CP14_WRITE

System Coprocessor register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP14 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc1 unsigned int

opcode 1.

opc2 unsigned int

opcode 2.

CP15_READ

System Coprocessor register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP15 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc1 unsigned int

opcode 1.

opc2 unsigned int

opcode 2.

CP15_READ64

System Coprocessor register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP15 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc unsigned int

opcode 1.

CP15_WRITE

System Control Coprocessor register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP15 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc1 unsigned int

opcode 1.

opc2 unsigned int

opcode 2.

CP15_WRITE64

System Coprocessor register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP15 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc unsigned int

opcode 1.

CPSR

CPSR change. (In AArch64 PSTATE represented in SPSR format). Fields:

CORE_NUM unsigned int

Core number in a multi processor.

OLD_VALUE unsigned int

The old CPSR value.

UNKNOWN unsigned int

Bits within the register that have unknown value.

VALUE unsigned int

The new CPSR value.

CRYPTO_SPEC

Every crypto instruction speculatively executed.

CorePowerStateChange

Core power-up state has changed. DISPLAY Core powered %{PU:(up|down)}. Fields:

PU bool

Whether the core powered up.

DATA_CACHE_SET_ALLOC_TAGS

Set allocation tags from DC GVA instruction. Fields:

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

NSDESC unsigned int

The security state of the access.

PADDR unsigned int

The physical address of the access.

PAS enum

The physical address space of the page.

RESPONSE enum

0=Aborted, 1=OK.

SIZE unsigned int

The size of memory in bytes.

TAG unsigned int

The value of the tag being set.

TAGGING_TYPE enum

Type of Memory tagging (Physical or Virtual).

TAG_VADDR unsigned int

The virtual address of the Virtual Tag if Virtual tagging is used; otherwise 0.

VADDR unsigned int

Virtual address used for setting allocation tags.

DATA_CACHE_ZERO

Zero and invalidate cache line from DC ZVA instruction. Fields:

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

NSDESC unsigned int

The security state of the access.

PADDR unsigned int

The physical address of the access.

PADDR_TAG unsigned int

If tag checked, the physical address tag for the access.

PAS enum

The physical address space of the page.

RESPONSE enum

0=Aborted, 1=OK.

SIZE unsigned int

The size of zero'd memory in bytes.

TAG_CHECKED bool

Is this access tag checked.

VADDR unsigned int

Virtual address of the zero'd cache line.

DEBUG_EVENT

Hardware debug support event. Fields:

EVENT enum

Description of event.

VALUE unsigned int

data value.

DEBUG_TARGET_ISET

Hardware debug target ISA. Fields:

TARGET_ISET enum

The target instruction set of the debug state.

DMI_ALLOCATE

DMI allocated. Fields:

ACCESS_DIRECTION enum

READ / WRITE.

PADDR unsigned int

The physical address of the DMI.

PTR unsigned int

DMI range pointer.

SIDE enum

DSIDE / ISIDE.

SIZE unsigned int

The size of DMI.

VADDR unsigned int

The virtual address of the DMI..

DMI_HIT

DMI hit, access used DMI (only fire for small DMI regions i.e. less than a full page). Fields:

ACCESS_DIRECTION enum

READ / WRITE.

PADDR unsigned int

The physical address of the DMI.

SIDE enum

DSIDE / ISIDE.

SIZE unsigned int

The size of the DMI.

VADDR unsigned int

The virtual address of the DMI..

DMI_REVOKE

DMI revoked. Fields:

ACCESS_DIRECTION enum

READ / WRITE.

PADDR unsigned int

The physical address of the DMI.

SIDE enum

DSIDE / ISIDE.

SIZE unsigned int

The size of DMI.

VADDR unsigned int

The virtual address of the DMI..

END_COMPILE

Compilation end. Fields:

END_OF_PAGE_COUNT unsigned int

Number of basic blocks exited due to page boundary since START_COMPILE.

FETCHFAIL_COUNT unsigned int

Number of basic blocks exited due to ifetch failure START_COMPILE.

INST_COUNT unsigned int

Number of instructions compiled since START_COMPILE.

NONSEQ_COUNT unsigned int

Number of basic blocks exited due to non-sequential instructions since START_COMPILE.

PAGE_STRADDLE_COUNT unsigned int

Number of basic blocks exited due to unaligned instructions crossing page since START_COMPILE.

ESB

Error Synchronization Barrier hit. Fields:

ASYNC bool

ESB is async.

IMPLICIT bool

ESB is implicit.

EXCEPTION

Exception taken. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ESR unsigned int

The value assigned to ESR when exception is taken.

ESR_EC unsigned int

The Value assigned to ESR.EC when exception is taken.

IS_PHYSICAL bool

Physical or Virtual exception.

IS_TRAP bool

Exception coming from a trap.

LR unsigned int

The value assigned to the link register.

PC unsigned int

The location where the exception occurred.

PREFERRED_RETURN unsigned int

The preferred return address for the exception.

TAKEN_LOCALLY bool

Whether exception is taken locally(same exception level).

TARGET_ISET enum

The instruction set of the exception handler code.

TARGET_PC unsigned int

The address the exception branches to.

VECTOR enum

The exception vector.

VECTOR_OFFSET unsigned int

The offset of an exception vector from the base address.

EXCEPTION_END

Every exception completed.

EXCEPTION_INFO

Trace human-readable reason of the current exception for data/instruction aborts, floating point exceptions and unknown exceptions. Fields:

DESCRIPTION string

Reason for current exception.

ESR_EC unsigned int

The value of ESR.EC when exception is taken.

EXCEPTION_RAISE

Every exception raised.

EXCEPTION_RETURN

Branches on leaving exception. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISET enum

The instruction set of the branch instruction.

IS_COND bool

Indicates if this is a conditional branch.

IS_HINTED bool

Whether the conditional instruction is hinted.

IS_ISB bool

Whether the instruction is ISB.

IS_LINK bool

Indicates if this is a branch with link.

PC unsigned int

The address of the branch instruction.

TARGET_ISET enum

The instruction set after the branch.

TARGET_PC unsigned int

The address the instruction branches to.

EXCEPTION_RETURN_PREBRANCH

Exception return event that must occur before PC and processor state have updated.

EXCEPTION_START

Every exception started.

FIQ_TAKEN

FIQ taken exception.

FP_EXCEPTIONS_SIGNALLED

Trace which FP exceptions were signalled as a part of the FP operations performed by an instruction. Fields:

FP_EXCEPTIONS_SIGNALLED_MASK unsigned int

Indicates which FP exceptions were signalled by an instruction. This uses the same encoding as the cumulative floating-point exception bitfields in FPSR.

FP_STATE

Floating point state. Fields:

FPSR unsigned int

FPSR register value.

MASK unsigned int

The mask indicating updated cumulative bits.

FREQ_CHANGED

Core frequency changed. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The instruction count of this core.

NEW_FREQ signed int

The new frequency of this core (expressed in Hz).

OLD_FREQ signed int

The old frequency of this core (expressed in Hz).

HLT

HLT instruction occurred. Fields:

IMM16 unsigned int

The 16-bit immediate value encoded in HLT instruction.

INST

Every instruction executed. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DEBUG_STATE bool

The instruction is executed in debug state.

DISASS string

Disassembly of instruction.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISSET enum

The current instruction set.

ITSTATE unsigned int

The current ITSTATE.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

MODE enum

The mode the core is in.

NS unsigned int

The core's non-secure bit.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

OPCODE unsigned int

The opcode of the instruction.

PADDR unsigned int

The physical address of the instruction.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the instruction.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the page.

PC unsigned int

The address of the instruction.

SECURITY_STATE enum

The core's security state.

SIZE unsigned int

The size of the instruction in bytes.

INST_END

Every instruction completed.

INST_START

Every instruction started. Fields:

DEBUG_STATE bool

The instruction is executed in debug state.

ISET enum

The current instruction set.

MODE enum

The mode the core is in.

NS enum

The current Secure State.

PC unsigned int

The address of the conditional instruction.

SECURITY_STATE enum

The current Security State.

INST_STRADDLE

Instruction straddles boundary. Fields:

DEBUG_STATE bool

The instruction is executed in debug state.

ISET enum

The current instruction set.

MODE enum

The mode the core is in.

NS enum

The current Secure State.

PC unsigned int

The address of the conditional instruction.

SECURITY_STATE enum

The current Security State.

IRQ_TAKEN

IRQ taken exception.

ISB

ISB instruction occurred. Fields:

PC unsigned int

The address of the ISB instruction.

LOAD_MULTIPLE_REGS_LIST

List of destination registers (not source) of a multiple registers load instruction. Fields:

REG string

Register to add into the list.

LOCAL_MONITOR

Local monitor activity. Fields:

PADDR unsigned int

Local Monitor Address.

State enum

State of the monitor (Open/Exclusive).

MAPPING_CREATE

A memory mapping has been created. This is an implementation detail of the model and so may change without notice. Fields:

ASID unsigned int

The ASID used during translation, or 0 if no ASID was used.

IS_VALID bool

Whether the mapping is valid or not.

PADDR unsigned int

Base physical address of region, or 0 if not known.

SIDE enum

I-side or D-side.

SIZE unsigned int

Size of the page, or 0 if not known.

VADDR unsigned int

Base virtual address of region.

VMID unsigned int

The VMID used during translation, or 0 if no VMID was used.

MAPPING_DESTROY

A memory mapping has been destroyed. This is an implementation detail of the model and so may change without notice. Fields:

ASID unsigned int

The ASID used during translation.

CAUSE enum

The cause of invalidation.

PADDR unsigned int

Base physical address of page.

SIDE enum

I-side or D-side.

SIZE unsigned int

Size of the page.

VADDR unsigned int

Base virtual address of page.

VMID unsigned int

The VMID used during translation.

MEMMAP_DEBUG_READ

Memory mapped reads to the debug registers. Fields:

ADDR unsigned int

address.

CORE_NUM unsigned int

Core number in a multi processor.

EXT bool

Whether access is from an external device (such as the DAP).

NS enum

Secure state of the access.

REG_NAME string

Name of the debug register accessed.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

MEMMAP_DEBUG_WRITE

Memory mapped writes to the debug registers. Fields:

ADDR unsigned int

address.

CORE_NUM unsigned int

Core number in a multi processor.

EXT bool

Whether access is from an external device (such as the DAP).

NS enum

Secure state of the access.

REG_NAME string

Name of the debug register accessed.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The new value written.

MMU_D128_TTB_READ

This event is triggered by reads caused by a D128 translation table walk. Fields:

ABORTED bool

Was the walk successful.

DATA_HIGH unsigned int

The data upper bits read.

DATA_LOW unsigned int

The data lower bits read.

IPA unsigned int

For stage 1, the IPA of the read.

LEVEL unsigned int

Translation table level.

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

OUTADDR unsigned int

The output physical address.

PADDR unsigned int

The physical address of the read.

PASmem enum

Physical Address Space of the accessed memory.

PASreq enum

Physical Address Space of the requested lookup.

REGIME_EL enum

Entry matches in this translation regime.

REQUESTER enum

The requester of this table walk.

SIDE enum

Inst / Data.

STAGE unsigned int

Translation stage.

MMU_D128_TTB_WRITE

This event is triggered by writes caused by a D128 translation table walk. Fields:

ABORTED bool

Was the walk successful.

DATA_HIGH unsigned int

The data upper bits written.

DATA_LOW unsigned int

The data lower bits written.

IPA unsigned int

For stage 1, the IPA of the write.

IS_HACDBS_ACCESS bool

If the access was due to HACDBS_CLEAN.

LEVEL unsigned int

Translation table level.

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

OUTADDR unsigned int

The output physical address.

PADDR unsigned int

The physical address of the write.

PASmem enum

Physical Address Space of the accessed memory.

PASreq enum

Physical Address Space of the requested lookup.

REGIME_EL enum

Entry matches in this translation regime.

SIDE enum

Inst / Data.

STAGE unsigned int

Translation stage.

MMU_TRANS

Address translation information. Fields:

ASID unsigned int

Address space identifier.

CORE_NUM unsigned int

Core number in a multi processor.

Hyp bool

Entry matches in Hyp state only.

INNERCACHE_RA bool

Is the inner cache allocate on read.

INNERCACHE_TRANSIENT bool

Is the inner write-through transient.

INNERCACHE_TYPE enum

Inner Caching scheme (NC/MB/WA).

INNERCACHE_WA bool

Is the inner cache allocate on write.

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MEMTYPE enum

Memory type.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC enum

Is secure side supposed to access secure or nonsecure memory.

OUTERCACHE_RA bool

Is the outer cache allocate on read.

OUTERCACHE_TRANSIENT bool

Is the outer write-through transient.

OUTERCACHE_TYPE enum

Outer Caching scheme (NC/MB/WA).

OUTERCACHE_WA bool

Is the outer cache allocate on write.

PADDR unsigned int

Physical address of the access.

PAGESIZE unsigned int

Page size as $\log_2(\text{size})$.

PAS enum

PAS supposed to be accessed is secure or nonsecure memory or Root/Realm if RME is implemented.

SH enum

Shareability.

SIDE enum

Inst / Data.

STAGE1_PERM unsigned int

Stage 1 permissions, [5:3] Privileged access (XWR) [2:0] User access (XWR).

STAGE2_PERM unsigned int

Stage 2 permission mask [5:3] Privileged access (XWR) [2:0] User access (XWR).

VADDR unsigned int

Virtual address of the access.

VMID unsigned int

Virtual machine identifier.

XS bool

XS Attribute.

nG enum

Flag indicating whether ASID will be matched.

MMU_TTB_READ

This event is triggered by reads caused by a translation table walk. Fields:

ABORTED bool

Was the walk successful.

DATA unsigned int

The data read.

DESC_KIND enum

The encoding scheme of the descriptors used for this walk.

IPA unsigned int

For stage 1, the IPA of the read.

LEVEL unsigned int

Translation table level.

LPAE bool

Is this for an LPAE translation (*DEPRECATED*, use DESC_KIND).

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSmem bool

Non secure memory access.

NSreq bool

Non secure request.

OUTADDR unsigned int

The output physical address.

PADDR unsigned int

The physical address of the read.

PASmem enum

Physical Address Space of the accessed memory.

PASreq enum

Physical Address Space of the requested lookup.

REGIME_EL enum

Entry matches in this translation regime.

REQUESTER enum

The requester of this table walk.

SIDE enum

Inst / Data.

STAGE unsigned int

Translation stage.

MMU_TTB_WRITE

This event is triggered by writes caused by a translation table walk. Fields:

ABORTED bool

Was the walk successful.

DATA unsigned int

The data written.

DESC_KIND enum

The encoding scheme of the descriptors used for this walk.

IPA unsigned int

For stage 1, the IPA of the write.

IS_HACDBS_ACCESS bool

If the access was due to HACDBS_CLEAN.

LEVEL unsigned int

Translation table level.

LPAE bool

Is this for an LPAE translation (*DEPRECATED*, use DESC_KIND).

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSmem bool

Non secure memory access.

NSreq bool

Non secure request.

OUTADDR unsigned int

The output physical address.

PADDR unsigned int

The physical address of the write.

PASmem enum

Physical Address Space of the accessed memory.

PASreq enum

Physical Address Space of the requested lookup.

REGIME_EL enum

Entry matches in this translation regime.

SIDE enum

Inst / Data.

STAGE unsigned int

Translation stage.

MODE_CHANGE

Mode change. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

MODE enum

The new mode.

OLD_MODE enum

The old mode.

MPU_TRANS

Address translation information. Fields:

ACCESS_STATUS enum

Access status(0xAA/0xBB/0xDD/0xEE=overlap/bg hit/default/bg fault).

CORE_NUM unsigned int

Core number in a multi processor.

EXEC_PERM enum

Execution Permission.

INNERCACHE_RA bool

Is the inner cache allocate on read.

INNERCACHE_TYPE enum

Inner Caching scheme (NC/MB/WA).

INNERCACHE_WA bool

Is the inner cache allocate on write.

MEMTYPE enum

Memory type.

NSDESC enum

Is non secure access.

OUTERCACHE_RA bool

Is the outer cache allocate on read.

OUTERCACHE_TYPE enum

Outer Caching scheme (NC/MB/WA).

OUTERCACHE_WA bool

Is the outer cache allocate on write.

PADDR unsigned int

Address of the access.

PAGESIZE unsigned int

Page size as $\log_2(\text{size})$.

PAS enum

PAS accessed is secure or nonsecure memory.

READ_PERM enum

Read Permission.

REG_NUM unsigned int

MPU region-number 0-31.

SH enum

Shareability.

SIDE enum

Inst / Data.

STAGE unsigned int

The translation stage.

WRITE_PERM enum

Write Permission.

NOT_TAG_CHECKED_INST

MTE Tag not checked for current instruction.

PERIODIC

Called for every quantum. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The instruction count of this CPU.

PC unsigned int

The address of the next instruction to be executed on this CPU.

PMU_COUNTER_OVERFLOW

PMU counter overflow. Fields:

COUNTER_GROUP enum

Counter group.

COUNTER_RANGE enum

Counter range.

EVENT_ID unsigned int

Event Identifier, field is valid only for Event counter group.

INDEX unsigned int

Counter index (as selected by PMSELR), field is valid only for Event/Cycle counter groups.

INTERRUPT bool

Is interrupt enabled on overflow for this counter.

POTINDEX_UPDATE

Value of POT index for current pc changed. Fields:

NEW_POTINDEX unsigned int

New POTIndex.

OLD_POTINDEX unsigned int

Old POTIndex.

PREFETCH_MEMORY64

Prefetch from PRFM or PRFUM instructions. Fields:

PRFOP enum

Prefetch hint.

VADDR unsigned int

Virtual address of the location that should be prefetched.

PRELOAD64_REQUEST

Preload requests from PRFM or PRFUM instructions. Fields:

PRFOP enum

Prefetch hint.

VADDR unsigned int

Virtual address of the location that should be prefetched.

PRELOAD_DATA

Data preload from PLD instruction. Fields:

VADDR unsigned int

Virtual address of the data that should be preloaded.

PRELOAD_INST

Instruction preload from PLI instruction. Fields:

VADDR unsigned int

Virtual address of the instruction that should be preloaded.

PRE_CORE_LOAD

Trace just before a core load. Fields:

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

PRE_CORE_STORE

Trace just before a core store. Fields:

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

PRE_TTB_READ

This event is triggered before reads caused by a translation table walk. Fields:

ADDR unsigned int

The physical address of the read.

PSTATE

PSTATE fields update. Fields:

ALLINT unsigned int

All interrupt mask bit.

BTYPE unsigned int

Branch target identification bit.

CORE_NUM unsigned int

Core number in a multi processor.

DAIF unsigned int

Exception mask bits.

DIT unsigned int

Data Independent Timing bit.

EL unsigned int

Current Exception level field.

EXLOCK unsigned int

Exception return state lock bit.

IL unsigned int

Illegal Execution state bit.

NZCV unsigned int

Condition flags.

PACM unsigned int

PACM bit.

PAN unsigned int

Privileged Access Never state bit.

PM unsigned int

PMU exception mask bit.

PPEND unsigned int

PMU exception pending bit.

SM unsigned int

StreamingMode status bit.

SP unsigned int

Stack pointer register selection bit.

SS unsigned int

Software Step bit.

SSBS unsigned int

Speculative Store Bypass Safe bit.

TCO unsigned int

Tag Check Override bit.

TOUCHED unsigned int

Mask of the fields that have been touched since this trace has last been emitted. Note that touched does not necessarily mean the value changed. The index of the field is the bit index in the mask plus the index of this field plus one.

UAO unsigned int

User Access Override bit.

UINJ unsigned int

Undefined Instruction exceptions bit.

ZA unsigned int

ZA storage and ZT0 register enable bit.

nRW unsigned int

Current Execution state bit.

PSTATE_PACTIVE_OPMODE_TRANSITION

Trace pactive and Opmode mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Operating mode after the transition.

OLD_STATE enum

Operating mode before the transition.

PSTATE_PACTIVE_OPMODE_TRANSITION_DENIED

Trace the denied pactive and Opmode mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Operating mode after the transition.

OLD_STATE enum

Operating mode before the transition.

REASON string

Reason for the transition being denied.

PSTATE_PACTIVE_POWER_TRANSITION

Trace pactive and power mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Power mode after the transition.

OLD_STATE enum

Power mode before the transition.

PSTATE_PACTIVE_POWER_TRANSITION_DENIED

Trace the denied pactive and power mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Power mode after the transition.

OLD_STATE enum

Power mode before the transition.

REASON string

Reason for the transition being denied.

PSTATE_PACTIVE_THREAD_OPMODE_TRANSITION

Trace pactive and Opmode mode transitions for threads on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Operating mode after the transition.

OLD_STATE enum

Operating mode before the transition.

QUANTUM_START

Called at the start of every quantum.

RANGE_PRELOAD_REQUEST

Preload requests from RPRFM. Fields:

COUNT unsigned int

Number of blocks of data to access.

LENGTH unsigned int

Number of contiguous bytes to be accessed in each block.

OP unsigned int

Range prefetch operation .

REUSE unsigned int

Maximum number of bytes to be accessed before next RPRFM .

STRIDE signed int

Stride to advance after accessing LENGTH bytes.

VADDR unsigned int

Base virtual address of range prefetch request.

RUN_STATE

Run state transition. Fields:

INST_COUNT unsigned int

Ticks count at point of transition.

NEW enum

New run state.

OLD enum

Old run state.

SEMIHOSTING_CALL

Call of a semihost function occurred. Fields:

PC unsigned int

The program counter after the semihosting call.

SEMIHOSTING_PRECALL

About to call semihost. Fields:

EL2 bool

Use EL2 translation regime.

NS bool

Is Non-Secure.

REG_WIDTH unsigned int

The current register width in bytes.

SIGNAL

External signal state change. Fields:

SIGNAL enum

Signal that changed.

STATE bool

Signal asserted state.

SOFTWARE_STEP

Return the Debug Software Step state. Fields:

STATE enum

Software Step state.

SPSR

SPSR change. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

MODE enum

Which of the banked SPSR registers is written.

OLD_VALUE unsigned int

The old SPSR value.

VALUE unsigned int

The new SPSR value.

START_COMPILE

Compilation started. Fields:

VADDR unsigned int

Instruction where compilation begins.

STG_MULTIPLE_MICRO_OPS_ADDRESS_RANGE

Address range for MTE stores with multiple uops. Fields:

ADDR_END unsigned int

Final address.

ADDR_START unsigned int

Starting address.

STORE_MULTIPLE_REGS_ADDRESS_RANGE

Address range for stores with multiple registers. Fields:

ADDR_END unsigned int

Final address.

ADDR_START unsigned int

Starting address.

SUBOPTIMAL_LDST_RETIRED

suboptimal load/store. Fields:

ACCESS_SIZE unsigned int

Log2 of the access width in bytes used for alignment checking (1: half-word, 2: word ...).

OPTIMAL_ALIGN_SELECTOR unsigned int

Selector to decide the optimal alignment size (1: access width, 2: 4-bytes, 3: 8-bytes, ..., 12: 4-kbytes).

VADDR unsigned int

The virtual address of the access.

SYNC

Called for every synchronization, that is every quantum break. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The instruction count of this CPU.

LOCAL_QUANTUM unsigned int

The local quantum of this CPU.

LOCAL_TIME unsigned int

The local time of this CPU.

SYNC_REASON enum

Reason for sync.

SYSCALL

System call instruction executed. Fields:

IMM unsigned int

Immediate value of the system call instruction.

TYPE enum

System call type.

VADDR unsigned int

Instruction that caused the system call.

SYSREG_READ128

System Coprocessor 128 bit register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

REGNUM enum

Internal register number.

REGNUM_PHYS enum

Internal register number physically accessed (e.g. in presense of VHE redirection).

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_READ64

System Coprocessor register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REGNUM enum

Internal register number.

REGNUM_OPERAND enum

Register number corresponding to the instruction operand.

REGNUM_PHYS enum

Internal register number physically accessed (e.g. in presense of VHE redirection).

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_UPDATE128

Triggers when the system updates a register. Fields:

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG enum

Register number.

UNKNOWN unsigned int

Bits of the register which became unknown.

VALUE unsigned int

Value written to the register.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_UPDATE32

Triggers when the system updates a register. Fields:

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG enum

Register number.

UNKNOWN unsigned int

Bits of the register which became unknown.

VALUE unsigned int

Value written to the register.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_UPDATE64

Triggers when the system updates a register. Fields:

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG enum

Register number.

UNKNOWN unsigned int

Bits of the register which became unknown.

VALUE unsigned int

Value written to the register.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_WRITE128

System Coprocessor 128 bit register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

REGNUM enum

Internal register number.

REGNUM_PHYS enum

Internal register number physically accessed (e.g. in presense of VHE redirection).

UNDEF bool

The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_WRITE64

System Coprocessor register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REGNUM enum

Internal register number.

REGNUM_OPERAND enum

Register number corresponding to the instruction operand.

REGNUM_PHYS enum

Internal register number physically accessed (e.g. in presense of VHE redirection).

UNDEF bool

The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

UNALIGNED_LDST_RETIRE

Processor unaligned load/store. Fields:

ACCESS_SIZE unsigned int

Log2 of the access width in bytes used for alignment checking (1=>half-word, 2->word ...).

NUMBER_OF_BEATS unsigned int

Number of accesses (beats) in this burst (Total size in bytes is ACCESS_SIZE * NUMBER_OF_BEATS).

VADDR unsigned int

The virtual address of the access.

VFP_D_REGS

VFP/NEON D 64 bit register write. Fields:

ALIASING enum

Type of register aliasing used.

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a write to the corresponding bit in the register. For example, 0x1 would indicate a write to VALUE[0].

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

VFP_D_REGS_READ

VFP/NEON D 64 bit register read. Fields:

ID unsigned int

The register number.

VALUE unsigned int

The current value of the register.

VFP_Q_REGS

VFP/NEON Q 128 bit register write. Fields:

ALIASING enum

Type of register aliasing used.

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a write to the corresponding byte in the register. For example, 0x1 would indicate a write to VALUE[7:0].

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

VFP_Q_REGS_READ

VFP/NEON Q 128 bit register read. Fields:

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a read to the corresponding byte in the register. For example, 0x1 would indicate a read to VALUE[7:0].

VALUE unsigned int

The current value of the register.

VFP_SYS_REGS

Writes to the VFP/NEON units system registers. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ID enum

Which VFP system register is written.

MASK unsigned int

The mask indicating updated cumulative bits.

OLD_VALUE unsigned int

The register's old value overwritten.

VALUE unsigned int

The new value written to the VFP system register.

VFP_S_REGS

VFP/NEON S 32 bit register write. Fields:

ALIASING enum

Type of register aliasing used.

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a write to the corresponding bit in the register. For example, 0x1 would indicate a write to VALUE[0].

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

VFP_S_REGS_READ

VFP/NEON S 32 bit register read. Fields:

ID unsigned int

The register number.

VALUE unsigned int

The current value of the register.

VIRTUAL_TAG_ALLOCATED

Virtual tag information when allocated. Fields:

TAG_VADDR unsigned int

Virtual tag virtual address.

VADDR unsigned int

Data Virtual address.

VIRTUAL_TAG unsigned int

Virtual tag value.

WAYPOINT

Signals end of basic block execution. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ISET enum

Origin instruction set.

IS_COND bool

Indicates if this is a conditional waypoint.

PC unsigned int

Origin address (or 0 if unavailable).

TAKEN bool

Indicates if this waypoint was taken.

TARGET unsigned int

Destination address.

TARGET_ISET enum

Destination instruction set.

WFE_END

WFE ended. Fields:

INST_COUNT unsigned int

Ticks count when leaving WFE.

WFE_EVENT_REGISTER

WFE event register status: set/clear, reason. Fields:

INST_COUNT unsigned int

Ticks count.

REASON enum

Reason for set/clear. Only REASON==0 (Cleared by WFE) clears the bit, all other reasons set it.

WFE_IGNORED

WFE ignored. Fields:

EVENT bool

This WFE was ignored because the event register was set.

INST_COUNT unsigned int

Ticks count when ignoring WFE.

TRAPPED bool

This WFE was trapped.

WFE_START

WFE entered. Fields:

INST_COUNT unsigned int

Ticks count when entering WFE.

WFI_END

WFI ended. Fields:

INST_COUNT **unsigned int**

Ticks count when leaving WFI.

WFI_IGNORED

WFI ignored. Fields:

DISABLED **bool**

This WFI was ignored because WFI is disabled.

INST_COUNT **unsigned int**

Ticks count when ignoring WFI.

REASON **enum**

specifies reason why WFI trace was ignored.

TRAPPED **bool**

This WFI was trapped.

WFI_START

WFI entered. Fields:

INST_COUNT **unsigned int**

Ticks count when entering WFI.

WFI_WAKEUP

WFI wakeup. Fields:

INST_COUNT **unsigned int**

Ticks count when WFI wakeup occurred.

REASON **enum**

Reason for wakeup.

tcm0

Trace TCM accesses.

tcm0_RAZ_WI

Trace TCM accesses. Fields:

ADDR **unsigned int**

Address of the **RAZ/WI** access.

tcm1

Trace TCM accesses.

tcm1_RAZ_WI

Trace TCM accesses. Fields:

ADDR **unsigned int**

Address of the **RAZ/WI** access.

tcm2

Trace TCM accesses.

tcm2_RAZ_WI

Trace TCM accesses. Fields:

ADDR unsigned int

Address of the **RAZ/WI** access.

2.52 ARM_Cortex-R7

This section describes the trace sources.

ASYNC_MEMORY_FAULT

Context ID Register write. Fields:

FAULT unsigned int

Fault status present in ESR.ISS encoding for Serror in ARCH64 and present in DFSR encoding for external abort format in ARCH32.

PADDR unsigned int

Physical Address (or 0 if unavailable).

ATOMIC_END_ACCESS

Bus trace access for atomics. Fields:

ACCESS_FAIL bool

Memory access failed.

ADDR unsigned int

The virtual address of the access.

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

COMPARE_VALUE unsigned int

Compare value for CAS.

LOAD_VALUE unsigned int

Loaded values.

NSDESC unsigned int

The security state of the access.

OPERAND_VALUE unsigned int

Operation's operand.

OPERATION enum

Operation type.

PADDR unsigned int

The physical address of the access.

PRIV bool

Is this a privileged access?.

ATOMIC_SLAVE_ACCESS

Information on the atomic operation at slave side. Fields:

ACCESS_RESULT unsigned int

Result of atomic access.

COMPARE_VALUE unsigned int

Value to be compared with memory by CAS.

LOAD_VALUE unsigned int

Value to be loaded from memory as a result of atomic operation.

MANAGER unsigned int

ID of manager which initiated the transaction.

NS bool

Transaction is Non-Secure.

OPERAND_VALUE unsigned int

Incoming operand value provided by core.

OPERATION enum

Type of atomic operation.

PADDR unsigned int

Full physical address of TCM.

STORE_VALUE unsigned int

Value to be stored to memory as a result of atomic operation.

ATOMIC_START_ACCESS

Bus trace access for atomics. Fields:

ADDR unsigned int

The virtual address of the access.

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

COMPARE_VALUE unsigned int

Compare value for CAS.

NSDESC unsigned int

The security state of the access.

OPERAND_VALUE unsigned int

Operation's operand.

OPERATION enum

Operation type.

PADDR unsigned int

The physical address of the access.

PRIV bool

Is this a privileged access?.

ArchMsg.Error.exit_code

DISPLAY exit code: %{EXITCODE} component kind: %{KIND} reason: %{REASON}. Fields:

EXITCODE signed int

Exit code to be returned.

KIND string

Component kind that invoked the exit code trace.

REASON string

Reason why the exit code trace was invoked (optional).

ArchMsg.Warning.IgnoredWarmResetRequest

Event fires when software attempts to request warm reset through write to deprecated DBGPRCR.CWRR.

ArchMsg.Warning.ReportedRevisionRegisterRead

DISPLAY Reported revision and patch do not match that of the cpu. Fields:

ORIGINAL_VALUE unsigned int

The value the register would have based on the modelled revision.

REGISTER string

The name of the register that has ben read.

REPORTED_VALUE unsigned int

The value the register has been overridden to.

ArchMsg.Warning.Unpredictably Indexed PM Event Register

TAGS **UNPREDICTABLE** DISPLAY Event %{IsDirect:(Selector value|Register)} %{SEL} is >= %{N}. Fields:

IsDirect unsigned int

Direct Access rather than indirect.

N unsigned int

Number of Accessible Registers.

SEL unsigned int

Selector.

ArchMsg.Warning.cache_contents_unknown

A cache has been activated without being invalidated and may contain **UNKNOWN** content. DISPLAY %{SIDE:(D|I)}-Cache was enabled in %{REGIME:(SECURE|NON_SECURE|HYP)} regime but was not invalidated since power-on: cache lines could contain **UNKNOWN** content %{SIDE:(Note: D-side caches must invalidate SECURE lines even if the cache is only used by NON_SECURE code in order to ensure that unexpected dirty lines tagged SECURE are not naturally evicted)}. Fields:

REGIME unsigned int

Current translation regime.

SIDE unsigned int

0 d-side cache or unified cache, 1 i-side.

ArchMsg.Warning.dcimvac_matches_watchpoint

TAGS **UNPREDICTABLE** DISPLAY Watchpoints matching an AArch32 DCIMVA are implementation defined.

ArchMsg.Warning.decode_fieldmismatch

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with different fields %{FIELD1} and %{FIELD2}. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_fracbitsnegative

TAGS **UNPREDICTABLE** DISPLAY VCVT Instruction is unpredictable with a negative number of fraction bits.

ArchMsg.Warning.decode_initblock

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable in an IT block.

ArchMsg.Warning.decode_initblocknotlast

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable in an IT block when not the last.

ArchMsg.Warning.decode_invalid_condition

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable when the condition is not AL.

ArchMsg.Warning.decode_invalidfieldcombination

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with the values in %{FIELD1} and %{FIELD2}. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_invalidvalue

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with the value in %{FIELD}. Fields:

FIELD enum

Field name.

ArchMsg.Warning.decode_mem_hint_unallocated

TAGS **UNPREDICTABLE** DISPLAY There is no memory hint allocated to this bit pattern.

ArchMsg.Warning.decode_registermatch

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with %{FIELD1} and %{FIELD2} as the same register. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_registermismatch

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with %{FIELD1} and %{FIELD2} as different registers. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_registeroutofrange

TAGS **UNPREDICTABLE** DISPLAY Floating-point instruction is unpredictable indexing beyond the end of the register bank.

ArchMsg.Warning.decode_sbzsbo

TAGS **UNPREDICTABLE** DISPLAY Reserved bits in the instruction are not canonical.

ArchMsg.Warning.decode_transactiontoobig

TAGS **UNPREDICTABLE** DISPLAY Attempt to perform a memory transaction of more than 128 bytes.

ArchMsg.Warning.decode_unpred_other

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable.

ArchMsg.Warning.decode_unpreduseofpc

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable when %{FIELD} is PC. Fields:

FIELD enum

Field name.

ArchMsg.Warning.decode_unpreduseofr13

TAGS UNPREDICTABLE DISPLAY Instruction is unpredictable when %{FIELD} is R13. Fields:

FIELD enum

Field name.

ArchMsg.Warning.decode_unpreduseofr13inreglist

TAGS UNPREDICTABLE DISPLAY Instruction is unpredictable with R13 in register-list.

ArchMsg.Warning.decode_useofd16orhigher

TAGS UNPREDICTABLE DISPLAY Attempt to use register D16 or higher where it is not allowed.

ArchMsg.Warning.decode_writebackandbaseinlist

TAGS UNPREDICTABLE DISPLAY Instruction is unpredictable with write back when the base register is in the transfer list.

ArchMsg.Warning.decode_zeroregistersinlist

TAGS UNPREDICTABLE DISPLAY Instruction is unpredictable when no registers are to be transferred.

ArchMsg.Warning.recursive_branch

An instruction is performing a branch that targets the same instruction. If this was intended then a WFI might be more effective DISPLAY recursive branch detected at PC=%{PC}. PRIMARY KEY PC. Fields:

PC unsigned int

Address of instruction causing the branch.

ArchMsg.Warning.recursive_exception

An instruction has generated an exception that targets the same instruction. DISPLAY recursive %{TYPE} exception detected at PC=%{PC} (CPSR=%{CPSR}). PRIMARY KEY PC. Fields:

CPSR unsigned int

Processor state.

PC unsigned int

Address of instruction causing the exception.

TYPE enum

Type of exception.

ArchMsg.Warning.reentrant_vector_catch

TAGS UNPREDICTABLE DISPLAY Vector catch debug event on Prefetch or Data abort vector %{ADDR}. Fields:

ADDR unsigned int

vector address.

ArchMsg.Warning.reserved_it_state

TAGS UNPREDICTABLE DISPLAY Execution with reserved IT state %{ITSTATE}. Fields:

ITSTATE unsigned int

reserved IT state.

ArchMsg.Warning.secure_vector_fetch_from_nonsecure

DISPLAY Secure vector table is being fetched from Non-secure memory.

ArchMsg.Warning.unpred_FAR_on_LRCPC3_non_atomic_mem_access

TAGS **UNPREDICTABLE** Unpredictable FAR/FSTATUS on LRCPC3 pair instructions resulting in two separate memory accessesDISPLAY Unpredictable FAR on LRCPC3 non atomic memory access at %{ADDR1} with FSTATUS %{FSTATUS1} or %{ADDR2} with FSTATUS %{FSTATUS2}. Fields:

ADDR1 unsigned int

Possible FAR address 1.

ADDR2 unsigned int

Possible FAR address 2.

FSTATUS1 unsigned int

Possible fault status value 1.

FSTATUS2 unsigned int

Possible fault status value 2.

ArchMsg.Warning.unpred_async_tag_check_fail_on_sve_speculative_load_do_update_ffr

Unpredictable set to FALSE of the FFR element associated to SVE speculative load with an async tag check fail TAGS **UNPREDICTABLE**.

ArchMsg.Warning.unpredictable_a32_breakpoint

TAGS **UNPREDICTABLE** DISPLAY Unpredictable breakpoint on a A32 instruction at %{ADDR1} Offset %{OFFSET} BAS %{BAS} for a %{IS_ADDRESS_MISMATCH_BREAKPOINT:(Mismatch|Match)} Breakpoint number %{N:d} Breakpoint address %{ADDR2}. Fields:

ADDR1 unsigned int

Address.

ADDR2 unsigned int

Breakpoint address.

BAS unsigned int

BAS field.

IS_ADDRESS_MISMATCH_BREAKPOINT bool

Is a Mismatch breakpoint, not a Match breakpoint.

N unsigned int

Breakpoint number.

OFFSET unsigned int

Always has a value of zero.

ArchMsg.Warning.unpredictable_banked_register_access

TAGS **UNPREDICTABLE** DISPLAY Access to banked register (R=%{R:d} SYSm=%{SYSM:2x}) is unpredictable. Fields:

R unsigned int

R bit.

SYSM unsigned int

SYSm.

ArchMsg.Warning.unpredictable_ccfail_undef

TAGS **UNPREDICTABLE** DISPLAY An instruction %{OPCODE} at VA=%{VADDR} failed its condition codes check but would otherwise have caused a non-data-dependent trap or undefined exception. Behaviour is implementation-defined choice of exception or nop. Fields:

OPCODE unsigned int

opcode of the instruction.

VADDR unsigned int

Virtual address of the instruction being executed.

ArchMsg.Warning.unpredictable_ici_bits

TAGS **UNPREDICTABLE** DISPLAY ICI bits do not apply or are architecturally unpredictable for instruction.

ArchMsg.Warning.unpredictable_t32_breakpoint

TAGS **UNPREDICTABLE** DISPLAY Unpredictable breakpoint on a T32 instruction at %{ADDR1} Offset %{OFFSET} BAS %{BAS} for a %{IS_ADDRESS_MISMATCH_BREAKPOINT:(Mismatch|Match)} Breakpoint number %{N:d} Breakpoint address %{ADDR2}. Fields:

ADDR1 unsigned int

Address.

ADDR2 unsigned int

Breakpoint address.

BAS unsigned int

BAS field.

IS_ADDRESS_MISMATCH_BREAKPOINT bool

Is a Mismatch breakpoint, not a Match breakpoint.

N unsigned int

Breakpoint number.

OFFSET unsigned int

Indicates which halfword of a 32-bit Thumb instruction the breakpoint is set on. 0 for the first halfword or if the instruction is 16 bits wide, or 1 for the second halfword.

ArchMsg.Warning.unpredictable_t32_vector_catch

TAGS **UNPREDICTABLE** DISPLAY Unpredictable Vector catch exception on a T32 instruction at %{ADDR} Offset %{OFFSET}. Fields:

ADDR unsigned int

Address.

OFFSET unsigned int

Indicates which halfword of a 32-bit Thumb instruction the vector catch exception is set on. 0 for the first halfword or if the instruction is 16 bits wide, or 1 for the second halfword.

ArchMsg.Warning.unpredictable_watchpoint_far

TAGS **UNPREDICTABLE** DISPLAY The value for FAR/EDWAR for hitting this watchpoint may be anywhere in the range %{LOWER_BOUND} to %{UPPER_BOUND}. Fields:

LOWER_BOUND unsigned int

the lowest address accessed by the instruction that triggered the watchpoint.

UPPER_BOUND unsigned int

the highest watchpointed address accessed by that instruction.

ArchMsg.Warning.unpredictable_watchpoint_far_isscmp

DISPLAY The value for FAR/EDWAR for hitting this watchpoint may be anywhere in the range %{LOWER_BOUND} to %{UPPER_BOUND}. Fields:

LOWER_BOUND unsigned int

the lowest address accessed by the instruction that triggered the watchpoint.

UPPER_BOUND unsigned int

the highest watchpointed address accessed by that instruction.

ArchMsg.Warning.warning_AdvSIMDExpandImmUnexpectedZero

TAGS **UNPREDICTABLE** DISPLAY AdvSIMDExpandImm may treat this immediate value as **UNPREDICTABLE**.

ArchMsg.Warning.warning_ConditionalSMC

TAGS **UNPREDICTABLE** DISPLAY SMC instruction has **UNPREDICTABLE** effects when conditional and when combined with traps.

ArchMsg.Warning.warning_access_crosses_page_boundary

TAGS **UNPREDICTABLE** DISPLAY Access crosses page boundary.

ArchMsg.Warning.warning_access_crosses_page_boundary_has_fault

TAGS **UNPREDICTABLE** DISPLAY Access that crosses page boundary has fault(s).

ArchMsg.Warning.warning_access_crosses_page_boundary_has_fault_on_both_pages

TAGS **UNPREDICTABLE** DISPLAY Access that crosses page boundary has a fault on each page. Fields:

ADDR1 unsigned int

FAR address on first page.

ADDR2 unsigned int

FAR address on second page.

FSTATUS1 unsigned int

Fault status on first page.

FSTATUS2 unsigned int

Fault status on second page.

ArchMsg.Warning.warning_access_crosses_page_boundary_spanning_different_memory

TAGS UNPREDICTABLE DISPLAY Access at address %{ADDR} crosses page from MEM_TYPE %{MEMTYPE_PAGE1} to %{MEMTYPE_PAGE2}. Fields:

ADDR unsigned int

address of access crossing page.

MEMTYPE_PAGE1 signed int

First Page's memory type (0-Normal, 1-Non-Normal).

MEMTYPE_PAGE2 signed int

Second Page's memory type (0-Normal, 1-Non-Normal).

ArchMsg.Warning.warning_access_wraps_around_memory

TAGS UNPREDICTABLE DISPLAY Access wraps around memory in %{WIDTH:d}bit mode. Fields:

WIDTH unsigned int

Address width.

ArchMsg.Warning.warning_bcr_linking_status

TAGS UNPREDICTABLE DISPLAY DBGBCR.{BT2,BT} combination gives unpredictable behaviour.

ArchMsg.Warning.warning_bcr_mask_reserved

TAGS UNPREDICTABLE DISPLAY DBGBCR_ELn.MASK is set to a reserved value.

ArchMsg.Warning.warning_bvr_masked_bit_not_zero

TAGS UNPREDICTABLE DISPLAY DBGBVRn_EL1.VA contains a bit masked by DBGBCRn_EL1.MASK that is not zero.

ArchMsg.Warning.warning_ccsidr_unimplemented_level

TAGS UNPREDICTABLE DISPLAY CCSIDR read while CSSELR %{CSSELR:x} points to an unimplemented cache level. Fields:

CSSELR unsigned int

current effective value of CSSELR.

ArchMsg.Warning.warning_change_to_ns_when_tge_set

TAGS UNPREDICTABLE DISPLAY Attempting to change NS to 1 when HCR.TGE = 1.

ArchMsg.Warning.warning_contiguous_bit_check_abort

TAGS UNPREDICTABLE DISPLAY An abort occurred whilst attempting to check TLB entry is contiguous at %{ENTRY_ADDR}. Fields:

ENTRY_ADDR unsigned int

address of conflicting TLB entry.

ArchMsg.Warning.warning_contiguous_bit_error

TAGS **UNPREDICTABLE** DISPLAY TLB entry at %{ENTRY_ADDR} was not contiguous with entry at %{CONTIG_ADDR} Contents are %{ENTRY_DATA} but expected %{CONTIG_DATA}. Fields:

CONTIG_ADDR unsigned int

address of the TLB first read, with which this entry is expected to be contiguous.

CONTIG_DATA unsigned int

expected contents based on the entry at CONTIG_ADDR.

ENTRY_ADDR unsigned int

address of conflicting TLB entry.

ENTRY_DATA unsigned int

contents of conflicting TLB entry.

ArchMsg.Warning.warning_contiguous_bit_error_D128

TAGS **UNPREDICTABLE** DISPLAY TLB entry at %{ENTRY_ADDR} was not contiguous with entry at %{CONTIG_ADDR} Contents are %{ENTRY_DATA_HI} %{ENTRY_DATA_LO} but expected %{CONTIG_DATA}. Fields:

CONTIG_ADDR unsigned int

address of the TLB first read, with which this entry is expected to be contiguous.

CONTIG_DATA unsigned int

expected contents based on the entry at CONTIG_ADDR.

ENTRY_ADDR unsigned int

address of conflicting TLB entry.

ENTRY_DATA_HI unsigned int

Higher bit contents of conflicting TLB entry.

ENTRY_DATA_LO unsigned int

Lower bit contents of conflicting TLB entry.

ArchMsg.Warning.warning_cp10_cp11_mismatch

TAGS **UNPREDICTABLE** DISPLAY Unpredictable configuration of CP10 and CP11 controls at EL %{EL:d}. Fields:

EL unsigned int

EL owning the control.

ArchMsg.Warning.warning_cp10_cp11_reserved_value

TAGS **UNPREDICTABLE** DISPLAY Unpredictable configuration of CPACR CP10 (%{CP10}) or CP11 (%{CP11}). Fields:

CP10 unsigned int

CP10 access.

CP11 unsigned int

CP11 access.

ArchMsg.Warning.warning_csselr_level_out_of_range

TAGS **UNPREDICTABLE** DISPLAY CSSELR.Level out of range, written value %{LEVEL} with only %{IMPLEMENTED:d} levels implemented. Fields:

IMPLEMENTED unsigned int

number of cache levels implemented.

LEVEL unsigned int

written value of CSSELR.Level.

TYPE enum

The type of cache selected by CSSELR.InD and TnD.

ArchMsg.Warning.warning_debug_flow_control_bits_not_obeyed

TAGS **UNPREDICTABLE** DISPLAY Flow control bits for %{REGISTER} were not obeyed, causing %{ERROR}. Fields:

ERROR enum

type of error resulting.

REGISTER enum

register under consideration.

ArchMsg.Warning.warning_debug_register_access_during_reset

TAGS **UNPREDICTABLE** DISPLAY Unpredictable %{IS_WRITE:(read|write)} access to debug register offset %{OFFSET} during reset. Fields:

IS_WRITE unsigned int

Write Not Read.

OFFSET unsigned int

Register Offset.

ArchMsg.Warning.warning_decode_cps_inconsistent_fields

TAGS **UNPREDICTABLE** DISPLAY The mode setting fields in the CPS instruction are inconsistent.

ArchMsg.Warning.warning_decode_invalid_state

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable in current exception-level/security state.

ArchMsg.Warning.warning_default_cacheable_mmu_on

TAGS **UNPREDICTABLE** DISPLAY Default cacheable is enabled (HCR.DC = 1) while MMU is enabled.

ArchMsg.Warning.warning_default_cacheable_vmmu_off

TAGS **UNPREDICTABLE** DISPLAY Default cacheable is enabled (HCR.DC = 1) while HCR.VM = 0.

ArchMsg.Warning.warning_deprecated_wvr_bit_2_set

TAGS **UNPREDICTABLE** DISPLAY DBGWVR%{N:d}_EL1=%{WVR:h} has bit 2 set, deprecated in ARMv8. Fields:

N unsigned int

index.

WVR unsigned int

DBGWVR.

ArchMsg.Warning.warning_eret_while_software_step_active_pending

Missing ISB between setting MDSCR_EL1.SS and ERET TAGS **UNPREDICTABLE**.

ArchMsg.Warning.warning_exclusive_to_non_normal

TAGS **UNPREDICTABLE** DISPLAY Exclusive Access to Non-Normal Memory %{ADDR}. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_exclusive_to_non_writeback

TAGS **UNPREDICTABLE** DISPLAY Exclusive Access to Non-Writeback-Cacheable Memory %{ADDR}. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_execute_from_device_memory

TAGS **UNPREDICTABLE** DISPLAY An attempt was made to execute from Device memory.

ArchMsg.Warning.warning_illegal_cpsr_mode

TAGS **UNPREDICTABLE** DISPLAY Writing an illegal or unimplement mode (%{NEW_MODE:h}) to CPSR was unpredictable. Fields:

NEW_MODE unsigned int

New value of CPSR.M.

ArchMsg.Warning.warning_illegal_srs_mode

TAGS **UNPREDICTABLE** DISPLAY Illegal or **UNPREDICTABLE** mode (%{MODE:h}) used for SRS instruction. Fields:

MODE unsigned int

mode for Banked SP.

ArchMsg.Warning.warning_implementation_defined_read_debug_register_in_SCS

Read Access to Debug registers from the processor is IMPLEMENTATION DEFINED TAGS IMP_DEF. Fields:

PPB_OFFSET unsigned int

debug register offset.

ArchMsg.Warning.warning_implementation_defined_sequential_security_transitions_supported

The behavior of sequential instruction fetches that cross from non-secure to secure memory and contain an SG instruction is **CONSTRAINED UNPREDICTABLE TAGS UNPREDICTABLE**.

ArchMsg.Warning.warning_implementation_defined_stack_limit_check_supported

It is **IMPLEMENTATION DEFINED** whether stack pointer limit checking is performed for this instruction TAGS IMP_DEF. Fields:

ADDRESS unsigned int

Address of instruction.

OPCODE unsigned int

instruction opcode.

ArchMsg.Warning.warning_implementation_defined_taggednonsharedmemorytype

TAGS IMP_DEF DISPLAY Memory attribute has Tagged, non shared memory type; it is **IMPLEMENTATION DEFINED** whether the memory location is treated as Tagged or non-Tagged.

ArchMsg.Warning.warning_implementation_defined_write_debug_register_in_SCS

Write Access to Debug registers from the processor is **IMPLEMENTATION DEFINED** TAGS IMP_DEF. Fields:

DATA unsigned int

data attempted to be written.

PPB_OFFSET unsigned int

debug register offset.

ArchMsg.Warning.warning_invalid_tcr_granule

TAGS **UNPREDICTABLE** DISPLAY TCR.TG%{TG_ID:(0|1)} bits have been set to a granule size %{{REQUEST:d}K not implemented - using %{{SUBSTITUTE:d}K. Fields:

REQUEST signed int

page size requested (or 0 for reserved).

SUBSTITUTE signed int

best guess available page size.

TG_ID bool

bits TG0 or TG1.

ArchMsg.Warning.warning_load_multiple_user_registers_from_user_mode

TAGS **UNPREDICTABLE** DISPLAY An LDM(user registers) instruction executed from user mode.

ArchMsg.Warning.warning_mte_in_mem_attr_not_supported

TAGS **UNPREDICTABLE** DISPLAY Memory attribute %{{ATTR}} has Tagged memory type but Armv8.5-MTE is not implemented. Fields:

ATTR unsigned int

Memory attribute.

ArchMsg.Warning.warning_reserved_breakpoint_state_match

TAGS **UNPREDICTABLE** DISPLAY `%{IS_BREAKPOINT:(Watchpoint|Breakpoint)}` programmed with a reserved combination of HMC, SSC and `%{IS_BREAKPOINT:(PAC|PMC)}`. Fields:

IS_BREAKPOINT unsigned int

Is a breakpoint, not a watchpoint.

ArchMsg.Warning.warning_shareability

TAGS **UNPREDICTABLE** DISPLAY Unpredictable: combination of the 1st and 2nd stages of translation is Normal Inner Non-Cacheable, Outer Non-Cacheable.

ArchMsg.Warning.warning_software_step_set_while_enabled

TAGS **UNPREDICTABLE** DISPLAY MDSCR_EL1.SS set to 1 while software step debug exceptions are enabled.

ArchMsg.Warning.warning_thumb_instruction_wraps_around_memory

TAGS **UNPREDICTABLE** DISPLAY Thumb instruction wraps around memory.

ArchMsg.Warning.warning_ttbr_sbz_bits_are_not_zero

TAGS **UNPREDICTABLE** DISPLAY `%{TTBR}[%{MSB:d};0]` should be zero, but are not. Fields:

MSB unsigned int

x-1.

TTBR enum

which TTBR register.

ArchMsg.Warning.warning_unaligned_address_dbgdtrrx_write

TAGS **UNPREDICTABLE** DISPLAY Unaligned address `%{ADDR}` in X0/R0 when DBGDTRRX written in memory access mode. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unaligned_address_dbgdtrtx_read

TAGS **UNPREDICTABLE** DISPLAY Unaligned address `%{ADDR}` in X0/R0 when DBGDTRTX read in memory access mode. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unaligned_to_device

TAGS **UNPREDICTABLE** DISPLAY Unaligned Access to Device Memory `%{ADDR}`. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unaligned_to_strongly_ordered

TAGS UNPREDICTABLE DISPLAY Unaligned Access to Strongly Ordered Memory %{ADDR}.

Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unknown_sau_rnr

TAGS UNPREDICTABLE DISPLAY SAU_RNR was set to an unsupported value.

ArchMsg.Warning.warning_unpredictable_AIRCR_PRIS_and_BFHFNMINs

TAGS UNPREDICTABLE DISPLAY The effect of setting both AIRCR.BFHFNMINs and AIRCR.PRIS to 1 is UNPREDICTABLE.

ArchMsg.Warning.warning_unpredictable_AIRCR_VECTCLRACTIVE_when_not_in_debug

TAGS UNPREDICTABLE DISPLAY The effect of writing a 1 to AIRCR.VECTCLRACTIVE if the processor is not halted in Debug state is UNPREDICTABLE.

ArchMsg.Warning.warning_unpredictable_AIRCR_VECTRESET_and_SYSRESETREQ

TAGS UNPREDICTABLE DISPLAY When the processor is halted in Debug state, if a write to the register writes a 1 to both VECTRESET and SYSRESETREQ, the behavior is UNPREDICTABLE.

ArchMsg.Warning.warning_unpredictable_AIRCR_VECTRESET_when_not_in_debug

TAGS UNPREDICTABLE DISPLAY The effect of writing a 1 to AIRCR.VECTRESET if the processor is not halted in Debug state is UNPREDICTABLE.

ArchMsg.Warning.warning_unpredictable_AIRCR_incorrect_VKEY

TAGS UNPREDICTABLE DISPLAY The value 0x05FA must be written to AIRCR.VKEY, otherwise the register write is UNPREDICTABLE.

ArchMsg.Warning.warning_unpredictable_EXC_RETURN_Reserved_Bit

EXC_RETURN[23:7] are reserved with the special condition that all bits should be written as one. Values other than all 1s are UNPREDICTABLE. TAGS UNPREDICTABLE. Fields:

EXC_RETURN unsigned int

Special PC value.

SBO_MASK unsigned int

Which bits Should Be One to not UNPRED.

ArchMsg.Warning.warning_unpredictable_PMEVTYPEPn_ELO_VS_reserved_value

TAGS UNPREDICTABLE DISPLAY Unpredictable reserved value of PMEVTYPEn<n>_ELO.VS.

ArchMsg.Warning.warning_unpredictable_change_to_DEMCR_MON_STEP_at_insufficient_priority

TAGS UNPREDICTABLE DISPLAY The effect of changing DEMCR.MON_STEP at an execution priority that is lower than the priority of the DebugMonitor exception is **CONSTRAINED UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_change_to_priority_of_active_exception

TAGS UNPREDICTABLE DISPLAY Changing the priority of an active exception is UNPREDICTABLE.

ArchMsg.Warning.warning_unpredictable_clear_lspact

TAGS UNPREDICTABLE DISPLAY UNPREDICTABLE state on context switch. Fields:

DRVEXC_TAKEN unsigned int

Derived exception taken, not pended.

REACHED_LAST_STORE unsigned int

Only the last faulted.

ArchMsg.Warning.warning_unpredictable_dbg_exit_unaligned_dlr

TAGS UNPREDICTABLE DISPLAY Debug exit to AArch32 with DLR[0] set to 1. PC[0] can be set to 0 or to DLR[0].

ArchMsg.Warning.warning_unpredictable_exception_catch

TAGS UNPREDICTABLE DISPLAY Generation of an exception catch event is unpredictable.

ArchMsg.Warning.warning_unpredictable_exception_return_inconsistent_state

TAGS UNPREDICTABLE DISPLAY state on exception return is unpredictable.

ArchMsg.Warning.warning_unpredictable_exception_return_instruction

TAGS UNPREDICTABLE DISPLAY exception return instruction is unpredictable.

ArchMsg.Warning.warning_unpredictable_in_debug_state

TAGS UNPREDICTABLE DISPLAY Instruction is UNPREDICTABLE when executed in debug state.

ArchMsg.Warning.warning_unpredictable_mair_encoding

TAGS UNPREDICTABLE DISPLAY Memory attribute %{ATTR} has invalid MAIR encoding; this may lead to unpredictable results. Fields:

ATTR unsigned int

Memory attribute.

ArchMsg.Warning.warning_unpredictable_pmu_counter_access

TAGS UNPREDICTABLE DISPLAY Access to PMU counters from non-secure EL0 or EL1 is UNPREDICTABLE with MDCR_EL2.HPMN set to 0.

ArchMsg.Warning.warning_unpredictable_prioritization_breakpoint_match_vector_catch

TAGS UNPREDICTABLE DISPLAY Two events with the same priority occurred at the same instruction: Address Matching Vector Catch debug event, and Breakpoint debug event. It is constrained unpredictable which is taken.

ArchMsg.Warning.warning_unpredictable_stack_selection

TAGS UNPREDICTABLE DISPLAY UNPREDICTABLE state on context switch.

ArchMsg.Warning.warning_unpredictable_tsize_out_of_range

TAGS UNPREDICTABLE DISPLAY %{TCR}.T%{TTBR:d}SZ (%{TSIZE:d} bits) is out of range. Must be between 25 and %{TSIZE_MAX:d}. Fields:

TCR enum

Which TCR.

TSIZE signed int

Value of TSize.

TSIZE_MAX signed int

The maximum allowed value.

TTBR unsigned int

Which TTBR region.

ArchMsg.Warning.warning_unpredictable_unaligned_pc_as_base_register

TAGS **UNPREDICTABLE** DISPLAY The PC value must be word-aligned, otherwise the behavior of the instruction is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_unaligned_pop_stack

TAGS **UNPREDICTABLE** DISPLAY **CONSTRAINED UNPREDICTABLE** unaligned stack frame %{FRAME_ADDR:x} in exception-return. Fields:

FRAME_ADDR unsigned int

unaligned stack frame address.

ArchMsg.Warning.warning_unpredictable_unaligned_to_non_writeback

TAGS **UNPREDICTABLE** DISPLAY Unaligned Access to Non-Writeback-Cacheable Memory.

ArchMsg.Warning.warning_unpredictable_vmsa_memattrib

TAGS **UNPREDICTABLE** DISPLAY Unpredictable vmsa memory attribute with texcb value programmed to %{TEXCB} when tex remap is %{TEX_REMAP}. Fields:

TEXCB unsigned int

texcb value.

TEX_REMAP bool

use tex remap.

ArchMsg.Warning.warning_unpredictable_vtcr_t0sz_sl0

TAGS **UNPREDICTABLE** DISPLAY The combination of VTCR.TOSZ=%{TOSZ:d} and VTCR.SLO=%{SLO:d} is unpredictable. Fields:

SLO signed int

VTCR.SLO, starting level for stage 2 translation table walks.

TOSZ signed int

VTCR.TOSZ, required input address range.

ArchMsg.Warning.warning_unpredictable_write_DHCSR

TAGS **UNPREDICTABLE** DISPLAY **UNPREDICTABLE** set of bit-changes in DHCSR. Fields:

NEW unsigned int

data attempted to be written.

OLD unsigned int

previous value.

ArchMsg.Warning.warning_unpredictable_write_SHCSR

TAGS **UNPREDICTABLE** DISPLAY **UNPREDICTABLE** setting a pending bit to 1 for an exception with priority >= execution priority.

ArchMsg.Warning.warning_unsupported_access_to_memory_mapped_register

TAGS **UNPREDICTABLE** DISPLAY Access to memory mapped register is unsupported.

ArchMsg.Warning.warning_user_jmcr_access

TAGS UNPREDICTABLE DISPLAY User %{WRITE:(write|read)} access to JMCR is unpredictable.
Fields:

WRITE unsigned int
access is write.

ArchMsg.Warning.warning_wcr_mask_and_bas

TAGS UNPREDICTABLE DISPLAY DBGWCR_EL1.MASK is non-zero and BAS is not 0xff.

ArchMsg.Warning.warning_wcr_mask_reserved

TAGS UNPREDICTABLE DISPLAY DBGWCR_ELn.MASK is set to a reserved value.

ArchMsg.Warning.warning_wcr_non_configuous_bas

TAGS UNPREDICTABLE DISPLAY DBGWCR_EL1.BAS has non-contiguous set of ones.

ArchMsg.Warning.warning_wvr_masked_bit_not_zero

TAGS UNPREDICTABLE DISPLAY DBGWVRn_EL1.VA contains a bit masked by
DBGWCRn_EL1.MASK that is not zero.

BRANCH_MISPREDICT

Simulating branch mispredict. Fields:

PC unsigned int
Origin address (or 0 if unavailable).

BRA_DIR

Direct branches, to immediate address. Fields:

CORE_NUM unsigned int
Core number in a multi processor.

INST_COUNT unsigned int
The core's instruction counter, starting at 1 for the first instruction.

ISSET enum
The instruction set of the branch instruction.

IS_COND bool
Indicates if this is a conditional branch.

IS_HINTED bool
Whether the conditional instruction is hinted.

IS_ISB bool
Whether the instruction is ISB.

IS_LINK bool
Indicates if this is a branch with link.

PC unsigned int
The address of the branch instruction.

TARGET_ISET enum

The instruction set after the branch.

TARGET_PC unsigned int

The address the instruction branches to.

BRA_INDIR

Indirect branches, perhaps to a register. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISET enum

The instruction set of the branch instruction.

IS_COND bool

Indicates if this is a conditional branch.

IS_HINTED bool

Whether the conditional instruction is hinted.

IS_ISB bool

Whether the instruction is ISB.

IS_LINK bool

Indicates if this is a branch with link.

PC unsigned int

The address of the branch instruction.

TARGET_ISET enum

The instruction set after the branch.

TARGET_PC unsigned int

The address the instruction branches to.

BRA_INDIR_CCFAIL

Indirect branches fail, perhaps to a register.

BRA_RET_CCFAIL

Branch return fail, perhaps to a register.

CACHE_MAINTENANCE_OP

Cache Maintenance Operation. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

DATA unsigned int

Specified MVA or Set/way.

FUNCTION enum

Clean / Invalidate.

SCOPE enum

Affected region.

SIDE enum

Inst / Data.

CCFAIL

Conditional instruction condition check fail. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CCFAIL_UNC

Unconditional instruction condition check fail. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CCPASS

Conditional instruction condition check pass. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CCPASS_UNC

Unconditional instruction condition check pass. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CODE_CACHE_MAINT

Code cache maintenance. Fields:

END_INCL unsigned int

Inclusive end address for VA or PA requests.

NS bool

Non-secure world for PA.

START unsigned int

Start address for VA or PA requests.

TYPE enum

Request type.

COMPILE_BLOCK_END

Last instruction of basic block translated. Fields:

VADDR unsigned int

Address of next instruction after this basic block.

COMPILE_BLOCK_START

First instruction of basic block translated. Fields:

VADDR unsigned int

Address of first instruction in basic block.

COMPILE_INST

ARM instruction compiled. Fields:

DISASS string

Disassembly of the instruction.

ISSET enum

The instruction set of this instruction.

ITSTATE unsigned int

The ITSTATE current for the instruction.

OPCODE unsigned int

The opcode of the instruction.

PC unsigned int

The address of the instruction.

SIZE unsigned int

The size of the instruction in bytes.

CONTEXTIDR

Context ID Register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

NS bool

Secure or nonsecure banked register is accessed.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The new value written.

CONTEXT_SYNC

Called for every context synchronization event. Fields:

ARCH_CSE bool

Architectural Context Synchronization Event.

PC unsigned int

The address of the synchronization instruction.

SYSREG_SYNC bool

System register synchronization event.

CORE_ENDIAN

Core BE8 Big-Endian state changed. Fields:

BE8 bool

Core BE8 Big-Endian state.

CORE_INFO

Static processor attributes. Only triggered by a call to DumpState(). Fields:

ARCH_PROFILE enum

The architecture profile of the core.

CLUSTER_ID unsigned int

The cluster ID of this processor.

CORE_NUM unsigned int

The number of this core in an MP processor.

DETAILED_CORE_FEATURES enum

Detailed core features as a bitfield. Enum values are bitmasks to extract the info from the field.

FPU_VERSION enum

The VFP version implemented by the core.

MEM_ARCH enum

The memory architecture of the core.

MIDR_EL1 unsigned int

MIDR_EL1 value.

NUM_CORES unsigned int

The number of cores in this MP processor.

QUANTUM_SIZE unsigned int

The default quantum size of the core (in ticks).

SECURITY_FEATURES bool

Does the core have security features?.

CORE_LOADS

Processor load accesses. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ACQREL enum

Is this an acquire/release.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DATA unsigned int

The data read or written.

ELEMENT_SIZE unsigned int

Width of each element.

ENDIANNESS enum

Byte ordering of this access.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

LOCK enum

Normal, exclusive or locked access.

MECID unsigned int

Memory Encryption Context ID, if applicable.

MECID2 unsigned int

Memory Encryption Context ID of the second page, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PAGE_SIZE unsigned int

Page size as $\log_2(\text{size in bytes})$.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK, 2=Exclusive Failed, 3=Unsupported.

SIZE unsigned int

Width of the access in bytes. Only required if DATA is not traced.

TRANS bool

Is this a translated access.

VADDR unsigned int

The virtual address of the access.

CORE_REGS

Changes of the core registers R0 to R14. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number, 0 to 14.

MODE enum

Bank of the register accessed.

OLD_VALUE unsigned int

The old value overwritten.

PHYS_ID enum

The physical register accessed.

VALUE unsigned int

The new value written to the register.

CORE_STALLED_TRANSITION

Detect when a core is prevented from progressing PChannel states. Fields:

CURRENT_STATE enum

Current power mode of core.

REASON string

The reason a transition was stalled, and what must be done to unblock it.

TARGET_STATE enum

Desired power mode of core after transition.

CORE_STORES

Processor store accesses. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ACQREL enum

Is this an acquire/release.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DATA unsigned int

The data read or written.

ELEMENT_SIZE unsigned int

Width of each element.

ENDIANNESS enum

Byte ordering of this access.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

LOCK enum

Normal, exclusive or locked access.

MECID unsigned int

Memory Encryption Context ID, if applicable.

MECID2 unsigned int

Memory Encryption Context ID of the second page, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PAGE_SIZE unsigned int

Page size as $\log_2(\text{size in bytes})$.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK, 2=Exclusive Failed, 3=Unsupported.

SIZE unsigned int

Width of the access in bytes. Only required if DATA is not traced.

TRANS bool

Is this a translated access.

VADDR unsigned int

The virtual address of the access.

CP14_READ

System Coprocessor register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP14 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc1 unsigned int

opcode 1.

opc2 unsigned int

opcode 2.

CP14_WRITE

System Coprocessor register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP14 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc1 unsigned int

opcode 1.

opc2 unsigned int

opcode 2.

CP15_READ

System Coprocessor register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP15 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc1 unsigned int

opcode 1.

opc2 unsigned int

opcode 2.

CP15_READ64

System Coprocessor register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP15 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc unsigned int

opcode 1.

CP15_WRITE

System Control Coprocessor register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP15 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc1 unsigned int

opcode 1.

opc2 unsigned int
opcode 2.

CP15_WRITE64

System Coprocessor register write. Fields:

CORE_NUM unsigned int
Core number in a multi processor.

CRm unsigned int
CRm.

NS bool
Secure or nonsecure banked register is accessed.

REG_NAME string
Name of the CP15 register accessed.

SECURITY_STATE enum
The current security state.

UNDEF bool
The register accessed is undefined.

UPDATED_VALUE unsigned int
Updated value of the register now it has been written.

VALUE unsigned int
The new value written.

opc unsigned int
opcode 1.

CPSR

CPSR change. (In AArch64 PSTATE represented in SPSR format). Fields:

CORE_NUM unsigned int
Core number in a multi processor.

OLD_VALUE unsigned int
The old CPSR value.

UNKNOWN unsigned int
Bits within the register that have unknown value.

VALUE unsigned int
The new CPSR value.

CRYPTO_SPEC

Every crypto instruction speculatively executed.

CorePowerStateChange

Core power-up state has changed. DISPLAY Core powered %{PU:(up|down)}. Fields:

PU bool

Whether the core powered up.

DATA_CACHE_SET_ALLOC_TAGS

Set allocation tags from DC GVA instruction. Fields:

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

NSDESC unsigned int

The security state of the access.

PADDR unsigned int

The physical address of the access.

PAS enum

The physical address space of the page.

RESPONSE enum

0=Aborted, 1=OK.

SIZE unsigned int

The size of memory in bytes.

TAG unsigned int

The value of the tag being set.

TAGGING_TYPE enum

Type of Memory tagging (Physical or Virtual).

TAG_VADDR unsigned int

The virtual address of the Virtual Tag if Virtual tagging is used; otherwise 0.

VADDR unsigned int

Virtual address used for setting allocation tags.

DATA_CACHE_ZERO

Zero and invalidate cache line from DC ZVA instruction. Fields:

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

NSDESC unsigned int

The security state of the access.

PADDR unsigned int

The physical address of the access.

PADDR_TAG unsigned int

If tag checked, the physical address tag for the access.

PAS enum

The physical address space of the page.

RESPONSE enum

0=Aborted, 1=OK.

SIZE unsigned int

The size of zero'd memory in bytes.

TAG_CHECKED bool

Is this access tag checked.

VADDR unsigned int

Virtual address of the zero'd cache line.

DEBUG_EVENT

Hardware debug support event. Fields:

EVENT enum

Description of event.

VALUE unsigned int

data value.

DEBUG_TARGET_ISET

Hardware debug target ISA. Fields:

TARGET_ISET enum

The target instruction set of the debug state.

DMI_ALLOCATE

DMI allocated. Fields:

ACCESS_DIRECTION enum

READ / WRITE.

PADDR unsigned int

The physical address of the DMI.

PTR unsigned int

DMI range pointer.

SIDE enum

DSIDE / ISIDE.

SIZE unsigned int

The size of DMI.

VADDR unsigned int

The virtual address of the DMI..

DMI_HIT

DMI hit, access used DMI (only fire for small DMI regions i.e. less than a full page). Fields:

ACCESS_DIRECTION enum

READ / WRITE.

PADDR unsigned int

The physical address of the DMI.

SIDE enum

DSIDE / ISIDE.

SIZE unsigned int

The size of the DMI.

VADDR unsigned int

The virtual address of the DMI..

DMI_REVOKE

DMI revoked. Fields:

ACCESS_DIRECTION enum

READ / WRITE.

PADDR unsigned int

The physical address of the DMI.

SIDE enum

DSIDE / ISIDE.

SIZE unsigned int

The size of DMI.

VADDR unsigned int

The virtual address of the DMI..

DTCM_ACCESS

Trace TCM accesses.

DTCM_ACCESS_RAZ_WI

Trace TCM accesses. Fields:

ADDR unsigned int

Address of the **RAZ/WI** access.

END_COMPILE

Compilation end. Fields:

END_OF_PAGE_COUNT unsigned int

Number of basic blocks exited due to page boundary since **START_COMPILE**.

FETCHFAIL_COUNT unsigned int

Number of basic blocks exited due to ifetch failure START_COMPILE.

INST_COUNT unsigned int

Number of instructions compiled since START_COMPILE.

NONSEQ_COUNT unsigned int

Number of basic blocks exited due to non-sequential instructions since START_COMPILE.

PAGE_STRADDLE_COUNT unsigned int

Number of basic blocks exited due to unaligned instructions crossing page since START_COMPILE.

EXCEPTION

Exception taken. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ESR unsigned int

The value assigned to ESR when exception is taken.

ESR_EC unsigned int

The Value assigned to ESR.EC when exception is taken.

IS_PHYSICAL bool

Physical or Virtual exception.

IS_TRAP bool

Exception coming from a trap.

LR unsigned int

The value assigned to the link register.

PC unsigned int

The location where the exception occurred.

PREFERRED_RETURN unsigned int

The preferred return address for the exception.

TAKEN_LOCALLY bool

Whether exception is taken locally(same exception level).

TARGET_ISET enum

The instruction set of the exception handler code.

TARGET_PC unsigned int

The address the exception branches to.

VECTOR enum

The exception vector.

VECTOR_OFFSET unsigned int

The offset of an exception vector from the base address.

EXCEPTION_END

Every exception completed.

EXCEPTION_INFO

Trace human-readable reason of the current exception for data/instruction aborts, floating point exceptions and unknown exceptions. Fields:

DESCRIPTION string

Reason for current exception.

ESR_EC unsigned int

The value of ESR.EC when exception is taken.

EXCEPTION_RAISE

Every exception raised.

EXCEPTION_RETURN

Branches on leaving exception. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISSET enum

The instruction set of the branch instruction.

IS_COND bool

Indicates if this is a conditional branch.

IS_HINTED bool

Whether the conditional instruction is hinted.

IS_ISB bool

Whether the instruction is ISB.

IS_LINK bool

Indicates if this is a branch with link.

PC unsigned int

The address of the branch instruction.

TARGET_ISET enum

The instruction set after the branch.

TARGET_PC unsigned int

The address the instruction branches to.

EXCEPTION_RETURN_PREBRANCH

Exception return event that must occur before PC and processor state have updated.

EXCEPTION_START

Every exception started.

FP_EXCEPTIONS_SIGNALLED

Trace which FP exceptions were signalled as a part of the FP operations performed by an instruction. Fields:

FP_EXCEPTIONS_SIGNALLED_MASK unsigned int

Indicates which FP exceptions were signalled by an instruction. This uses the same encoding as the cumulative floating-point exception bitfields in FPSR.

FP_STATE

Floating point state. Fields:

FPSR unsigned int

FPSR register value.

MASK unsigned int

The mask indicating updated cumulative bits.

FREQ_CHANGED

Core frequency changed. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The instruction count of this core.

NEW_FREQ signed int

The new frequency of this core (expressed in Hz).

OLD_FREQ signed int

The old frequency of this core (expressed in Hz).

HLT

HLT instruction occurred. Fields:

IMM16 unsigned int

The 16-bit immediate value encoded in HLT instruction.

INST

Every instruction executed. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DEBUG_STATE bool

The instruction is executed in debug state.

DISASS string

Disassembly of instruction.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISET enum

The current instruction set.

ITSTATE unsigned int

The current ITSTATE.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

MODE enum

The mode the core is in.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

OPCODE unsigned int

The opcode of the instruction.

PADDR unsigned int

The physical address of the instruction.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the instruction.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the page.

PC unsigned int

The address of the instruction.

SECURITY_STATE enum

The core's security state.

SIZE unsigned int

The size of the instruction in bytes.

INST_END

Every instruction completed.

INST_START

Every instruction started. Fields:

DEBUG_STATE bool

The instruction is executed in debug state.

ISSET enum

The current instruction set.

MODE enum

The mode the core is in.

NS enum

The current Secure State.

PC unsigned int

The address of the conditional instruction.

SECURITY_STATE enum

The current Security State.

INST_STRADDLE

Instruction straddles boundary. Fields:

DEBUG_STATE bool

The instruction is executed in debug state.

ISSET enum

The current instruction set.

MODE enum

The mode the core is in.

NS enum

The current Secure State.

PC unsigned int

The address of the conditional instruction.

SECURITY_STATE enum

The current Security State.

ISB

ISB instruction occurred. Fields:

PC unsigned int

The address of the ISB instruction.

ITCM_ACCESS

Trace TCM accesses.

ITCM_ACCESS_RAZ_WI

Trace TCM accesses. Fields:

ADDR unsigned int

Address of the **RAZ/WI** access.

LOAD_MULTIPLE_REGS_LIST

List of destination registers (not source) of a multiple registers load instruction. Fields:

REG string

Register to add into the list.

LOCAL_MONITOR

Local monitor activity. Fields:

PADDR unsigned int

Local Monitor Address.

State enum

State of the monitor (Open/Exclusive).

MEMMAP_DEBUG_READ

Memory mapped reads to the debug registers. Fields:

ADDR unsigned int

address.

CORE_NUM unsigned int

Core number in a multi processor.

EXT bool

Whether access is from an external device (such as the DAP).

NS enum

Secure state of the access.

REG_NAME string

Name of the debug register accessed.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

MEMMAP_DEBUG_WRITE

Memory mapped writes to the debug registers. Fields:

ADDR unsigned int

address.

CORE_NUM unsigned int

Core number in a multi processor.

EXT bool

Whether access is from an external device (such as the DAP).

NS enum

Secure state of the access.

REG_NAME string

Name of the debug register accessed.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The new value written.

MODE_CHANGE

Mode change. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

MODE enum

The new mode.

OLD_MODE enum

The old mode.

MPU_TRANS

Address translation information. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

EXEC_PERM enum

Execution Permission.

INNERCACHE_RA bool

Is the inner cache allocate on read.

INNERCACHE_TYPE enum

Inner Caching scheme (NC/MB/WA).

INNERCACHE_WA bool

Is the inner cache allocate on write.

MEMTYPE enum

Memory type.

OUTERCACHE_RA bool

Is the outer cache allocate on read.

OUTERCACHE_TYPE enum

Outer Caching scheme (NC/MB/WA).

OUTERCACHE_WA bool

Is the outer cache allocate on write.

PADDR unsigned int

Address of the access.

PAGESIZE unsigned int

Page size as $\log_2(\text{size})$.

READ_PERM enum

Read Permission.

REG_NUM unsigned int

MPU register based region-number 0-15 (else 0xFF=none, 0xBB/0xDD=background/default).

SH enum

Shareability.

SIDE enum

Inst / Data.

WRITE_PERM enum

Write Permission.

PERIODIC

Called for every quantum. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The instruction count of this CPU.

PC unsigned int

The address of the next instruction to be executed on this CPU.

PMU_COUNTER_OVERFLOW

PMU counter overflow. Fields:

COUNTER_GROUP enum

Counter group.

COUNTER_RANGE enum

Counter range.

EVENT_ID unsigned int

Event Identifier, field is valid only for Event counter group.

INDEX unsigned int

Counter index (as selected by PMSELR), field is valid only for Event/Cycle counter groups.

INTERRUPT bool

Is interrupt enabled on overflow for this counter.

POTINDEX_UPDATE

Value of POT index for current pc changed. Fields:

NEW_POTINDEX unsigned int

New POTIndex.

OLD_POTINDEX unsigned int

Old POTIndex.

PRELOAD_DATA

Data preload from PLD instruction. Fields:

VADDR unsigned int

Virtual address of the data that should be preloaded.

PRELOAD_INST

Instruction preload from PLI instruction. Fields:

VADDR unsigned int

Virtual address of the instruction that should be preloaded.

PRE_CORE_LOAD

Trace just before a core load. Fields:

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

PRE_CORE_STORE

Trace just before a core store. Fields:

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

PSTATE

PSTATE fields update. Fields:

ALLINT unsigned int

All interrupt mask bit.

BTYPE unsigned int

Branch target identification bit.

CORE_NUM unsigned int

Core number in a multi processor.

DAIF unsigned int

Exception mask bits.

DIT unsigned int

Data Independent Timing bit.

EL unsigned int

Current Exception level field.

EXLOCK unsigned int

Exception return state lock bit.

IL unsigned int

Illegal Execution state bit.

NZCV unsigned int

Condition flags.

PACM unsigned int

PACM bit.

PAN unsigned int

Privileged Access Never state bit.

PM unsigned int

PMU exception mask bit.

PPEND unsigned int

PMU exception pending bit.

SM unsigned int

StreamingMode status bit.

SP unsigned int

Stack pointer register selection bit.

SS unsigned int

Software Step bit.

SSBS unsigned int

Speculative Store Bypass Safe bit.

TCO unsigned int

Tag Check Override bit.

TOUCHED unsigned int

Mask of the fields that have been touched since this trace has last been emitted. Note that touched does not necessarily mean the value changed. The index of the field is the bit index in the mask plus the index of this field plus one.

UAO unsigned int

User Access Override bit.

UINJ unsigned int

Undefined Instruction exceptions bit.

ZA unsigned int

ZA storage and ZTO register enable bit.

nRW unsigned int

Current Execution state bit.

PSTATE_PACTIVE_OPMODE_TRANSITION

Trace pactive and Opmode mode transitions on PChannel. Fields:

Copyright © 2023–2025 Arm Limited (or its affiliates). All rights reserved.

Non-Confidential

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Operating mode after the transition.

OLD_STATE enum

Operating mode before the transition.

PSTATE_PACTIVE_OPMODE_TRANSITION_DENIED

Trace the denied pactive and Opmode mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Operating mode after the transition.

OLD_STATE enum

Operating mode before the transition.

REASON string

Reason for the transition being denied.

PSTATE_PACTIVE_POWER_TRANSITION

Trace pactive and power mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Power mode after the transition.

OLD_STATE enum

Power mode before the transition.

PSTATE_PACTIVE_POWER_TRANSITION_DENIED

Trace the denied pactive and power mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Power mode after the transition.

OLD_STATE enum

Power mode before the transition.

REASON string

Reason for the transition being denied.

PSTATE_PACTIVE_THREAD_OPMODE_TRANSITION

Trace pactive and Opmode mode transitions for threads on PChannel. Fields:

DEVPACTIVE **unsigned int**

Integer representation of DEVPACTIVE[] signals.

NEW_STATE **enum**

Operating mode after the transition.

OLD_STATE **enum**

Operating mode before the transition.

QUANTUM_START

Called at the start of every quantum.

RUN_STATE

Run state transition. Fields:

INST_COUNT **unsigned int**

Ticks count at point of transition.

NEW **enum**

New run state.

OLD **enum**

Old run state.

SEMIHOSTING_CALL

Call of a semihost function occurred. Fields:

PC **unsigned int**

The program counter after the semihosting call.

SEMIHOSTING_PRECALL

About to call semihost. Fields:

EL2 **bool**

Use EL2 translation regime.

NS **bool**

Is Non-Secure.

REG_WIDTH **unsigned int**

The current register width in bytes.

SIGNAL

External signal state change. Fields:

SIGNAL **enum**

Signal that changed.

STATE **bool**

Signal asserted state.

SPSR

SPSR change. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

MODE enum

Which of the banked SPSR registers is written.

OLD_VALUE unsigned int

The old SPSR value.

VALUE unsigned int

The new SPSR value.

START_COMPILE

Compilation started. Fields:

VADDR unsigned int

Instruction where compilation begins.

STG_MULTIPLE_MICRO_OPS_ADDRESS_RANGE

Address range for MTE stores with multiple uops. Fields:

ADDR_END unsigned int

Final address.

ADDR_START unsigned int

Starting address.

STORE_MULTIPLE_REGS_ADDRESS_RANGE

Address range for stores with multiple registers. Fields:

ADDR_END unsigned int

Final address.

ADDR_START unsigned int

Starting address.

SUBOPTIMAL_LDST_RETIRED

suboptimal load/store. Fields:

ACCESS_SIZE unsigned int

Log2 of the access width in bytes used for alignment checking (1: half-word, 2: word ...).

OPTIMAL_ALIGN_SELECTOR unsigned int

Selector to decide the optimal alignment size (1: access width, 2: 4-bytes, 3: 8-bytes, ..., 12: 4-kbytes).

VADDR unsigned int

The virtual address of the access.

SYNC

Called for every synchronization, that is every quantum break. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The instruction count of this CPU.

LOCAL_QUANTUM unsigned int

The local quantum of this CPU.

LOCAL_TIME unsigned int

The local time of this CPU.

SYNC_REASON enum

Reason for sync.

SYSCALL

System call instruction executed. Fields:

IMM unsigned int

Immediate value of the system call instruction.

TYPE enum

System call type.

VADDR unsigned int

Instruction that caused the system call.

SYSREG_READ128

System Coprocessor 128 bit register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

REGNUM enum

Internal register number.

REGNUM_PHYS enum

Internal register number physically accessed (e.g. in presense of VHE redirection).

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_READ64

System Coprocessor register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REGNUM enum

Internal register number.

REGNUM_OPERAND enum

Register number corresponding to the instruction operand.

REGNUM_PHYS enum

Internal register number physically accessed (e.g. in presense of VHE redirection).

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_UPDATE128

Triggers when the system updates a register. Fields:

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG enum

Register number.

UNKNOWN unsigned int

Bits of the register which became unknown.

VALUE unsigned int

Value written to the register.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_UPDATE32

Triggers when the system updates a register. Fields:

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG enum

Register number.

UNKNOWN unsigned int

Bits of the register which became unknown.

VALUE unsigned int

Value written to the register.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_UPDATE64

Triggers when the system updates a register. Fields:

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG enum

Register number.

UNKNOWN unsigned int

Bits of the register which became unknown.

VALUE unsigned int

Value written to the register.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_WRITE128

System Coprocessor 128 bit register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

REGNUM enum

Internal register number.

REGNUM_PHYS enum

Internal register number physically accessed (e.g. in presense of VHE redirection).

UNDEF bool

The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_WRITE64

System Coprocessor register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REGNUM enum

Internal register number.

REGNUM_OPERAND enum

Register number corresponding to the instruction operand.

REGNUM_PHYS enum

Internal register number physically accessed (e.g. in presense of VHE redirection).

UNDEF bool

The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

UNALIGNED_LDST_RETIRED

Processor unaligned load/store. Fields:

ACCESS_SIZE unsigned int

Log2 of the access width in bytes used for alignment checking (1=>half-word, 2->word ...).

NUMBER_OF_BEATS unsigned int

Number of accesses (beats) in this burst (Total size in bytes is ACCESS_SIZE * NUMBER_OF_BEATS).

VADDR unsigned int

The virtual address of the access.

VFP_D_REGS

VFP/NEON D 64 bit register write. Fields:

ALIASING enum

Type of register aliasing used.

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a write to the corresponding bit in the register. For example, 0x1 would indicate a write to VALUE[0].

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

VFP_D_REGS_READ

VFP/NEON D 64 bit register read. Fields:

ID unsigned int

The register number.

VALUE unsigned int

The current value of the register.

VFP_Q_REGS

VFP/NEON Q 128 bit register write. Fields:

ALIASING enum

Type of register aliasing used.

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a write to the corresponding byte in the register. For example, 0x1 would indicate a write to VALUE[7:0].

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

VFP_Q_REGS_READ

VFP/NEON Q 128 bit register read. Fields:

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a read to the corresponding byte in the register. For example, 0x1 would indicate a read to VALUE[7:0].

VALUE unsigned int

The current value of the register.

VFP_SYS_REGS

Writes to the VFP/NEON units system registers. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ID enum

Which VFP system register is written.

MASK unsigned int

The mask indicating updated cumulative bits.

OLD_VALUE unsigned int

The register's old value overwritten.

VALUE unsigned int

The new value written to the VFP system register.

VFP_S_REGS

VFP/NEON S 32 bit register write. Fields:

ALIASING enum

Type of register aliasing used.

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a write to the corresponding bit in the register. For example, 0x1 would indicate a write to VALUE[0].

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

VFP_S_REGS_READ

VFP/NEON S 32 bit register read. Fields:

ID unsigned int

The register number.

VALUE unsigned int

The current value of the register.

WAYPOINT

Signals end of basic block execution. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ISET enum

Origin instruction set.

IS_COND bool

Indicates if this is a conditional waypoint.

PC unsigned int

Origin address (or 0 if unavailable).

TAKEN bool

Indicates if this waypoint was taken.

TARGET unsigned int

Destination address.

TARGET_ISET enum

Destination instruction set.

WFE_END

WFE ended. Fields:

INST_COUNT unsigned int

Ticks count when leaving WFE.

WFE_EVENT_REGISTER

WFE event register status: set/clear, reason. Fields:

INST_COUNT unsigned int

Ticks count.

REASON enum

Reason for set/clear. Only REASON==0 (Cleared by WFE) clears the bit, all other reasons set it.

WFE_IGNORED

WFE ignored. Fields:

EVENT bool

This WFE was ignored because the event register was set.

INST_COUNT unsigned int

Ticks count when ignoring WFE.

TRAPPED bool

This WFE was trapped.

WFE_START

WFE entered. Fields:

INST_COUNT unsigned int

Ticks count when entering WFE.

WFI_END

WFI ended. Fields:

INST_COUNT unsigned int

Ticks count when leaving WFI.

WFI_IGNORED

WFI ignored. Fields:

DISABLED bool

This WFI was ignored because WFI is disabled.

INST_COUNT unsigned int

Ticks count when ignoring WFI.

REASON enum

specifies reason why WFI trace was ignored.

TRAPPED bool

This WFI was trapped.

WFI_START

WFI entered. Fields:

INST_COUNT unsigned int

Ticks count when entering WFI.

WFI_WAKEUP

WFI wakeup. Fields:

INST_COUNT unsigned int

Ticks count when WFI wakeup occurred.

REASON enum

Reason for wakeup.

2.53 ARM_Cortex-R8

This section describes the trace sources.

ASYNC_MEMORY_FAULT

Context ID Register write. Fields:

FAULT unsigned int

Fault status present in ESR.ISS encoding for Serror in ARCH64 and present in DFSR encoding for external abort format in ARCH32.

PADDR unsigned int

Physical Address (or 0 if unavailable).

ATOMIC_END_ACCESS

Bus trace access for atomics. Fields:

ACCESS_FAIL bool

Memory access failed.

ADDR unsigned int

The virtual address of the access.

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

COMPARE_VALUE unsigned int

Compare value for CAS.

LOAD_VALUE unsigned int

Loaded values.

NSDESC unsigned int

The security state of the access.

OPERAND_VALUE unsigned int

Operation's operand.

OPERATION enum

Operation type.

PADDR unsigned int

The physical address of the access.

PRIV bool

Is this a privileged access?.

ATOMIC_SLAVE_ACCESS

Information on the atomic operation at slave side. Fields:

ACCESS_RESULT unsigned int

Result of atomic access.

COMPARE_VALUE unsigned int

Value to be compared with memory by CAS.

LOAD_VALUE unsigned int

Value to be loaded from memory as a result of atomic operation.

MANAGER unsigned int

ID of manager which initiated the transaction.

NS bool

Transaction is Non-Secure.

OPERAND_VALUE unsigned int

Incoming operand value provided by core.

OPERATION enum

Type of atomic operation.

PADDR unsigned int

Full physical address of TCM.

STORE_VALUE unsigned int

Value to be stored to memory as a result of atomic operation.

ATOMIC_START_ACCESS

Bus trace access for atomics. Fields:

ADDR unsigned int

The virtual address of the access.

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

COMPARE_VALUE unsigned int

Compare value for CAS.

NSDESC unsigned int

The security state of the access.

OPERAND_VALUE unsigned int

Operation's operand.

OPERATION enum

Operation type.

PADDR unsigned int

The physical address of the access.

PRIV bool

Is this a privileged access?.

ArchMsg.Error.exit_code

DISPLAY exit code: %{EXITCODE} component kind: %{KIND} reason: %{REASON}. Fields:

EXITCODE signed int

Exit code to be returned.

KIND string

Component kind that invoked the exit code trace.

REASON string

Reason why the exit code trace was invoked (optional).

ArchMsg.Warning.IgnoredWarmResetRequest

Event fires when software attempts to request warm reset through write to deprecated DBGPRCR.CWRR.

ArchMsg.Warning.ReportedRevisionRegisterRead

DISPLAY Reported revision and patch do not match that of the cpu. Fields:

ORIGINAL_VALUE unsigned int

The value the register would have based on the modelled revision.

REGISTER string

The name of the register that has ben read.

REPORTED_VALUE unsigned int

The value the register has been overridden to.

ArchMsg.Warning.Unpredictably Indexed PM Event Register

TAGS **UNPREDICTABLE** DISPLAY Event %{IsDirect:(Selector value|Register)} %{SEL} is >= %{N}. Fields:

IsDirect unsigned int

Direct Access rather than indirect.

N unsigned int

Number of Accessible Registers.

SEL unsigned int

Selector.

ArchMsg.Warning.cache_contents_unknown

A cache has been activated without being invalidated and may contain **UNKNOWN** content. DISPLAY %{SIDE:(D|I)}-Cache was enabled in %{REGIME:(SECURE|NON_SECURE|HYP)} regime but was not invalidated since power-on: cache lines could contain **UNKNOWN** content %{SIDE:(Note: D-side caches must invalidate SECURE lines even if the cache is only used by NON_SECURE code in order to ensure that unexpected dirty lines tagged SECURE are not naturally evicted)}. Fields:

REGIME unsigned int

Current translation regime.

SIDE unsigned int

0 d-side cache or unified cache, 1 i-side.

ArchMsg.Warning.dcimvac_matches_watchpoint

TAGS **UNPREDICTABLE** DISPLAY Watchpoints matching an AArch32 DCIMVA are implementation defined.

ArchMsg.Warning.decode_fieldmismatch

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with different fields %{FIELD1} and %{FIELD2}. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_fracbitsnegative

TAGS **UNPREDICTABLE** DISPLAY VCVT Instruction is unpredictable with a negative number of fraction bits.

ArchMsg.Warning.decode_initblock

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable in an IT block.

ArchMsg.Warning.decode_initblocknotlast

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable in an IT block when not the last.

ArchMsg.Warning.decode_invalid_condition

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable when the condition is not AL.

ArchMsg.Warning.decode_invalidfieldcombination

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with the values in %{FIELD1} and %{FIELD2}. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_invalidvalue

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with the value in %{FIELD}. Fields:

FIELD enum

Field name.

ArchMsg.Warning.decode_mem_hint_unallocated

TAGS **UNPREDICTABLE** DISPLAY There is no memory hint allocated to this bit pattern.

ArchMsg.Warning.decode_registermatch

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with %{FIELD1} and %{FIELD2} as the same register. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_registermismatch

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with %{FIELD1} and %{FIELD2} as different registers. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_registeroutofrange

TAGS **UNPREDICTABLE** DISPLAY Floating-point instruction is unpredictable indexing beyond the end of the register bank.

ArchMsg.Warning.decode_sbzsbo

TAGS **UNPREDICTABLE** DISPLAY Reserved bits in the instruction are not canonical.

ArchMsg.Warning.decode_transactiontoobig

TAGS **UNPREDICTABLE** DISPLAY Attempt to perform a memory transaction of more than 128 bytes.

ArchMsg.Warning.decode_unpred_other

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable.

ArchMsg.Warning.decode_unpreduseofpc

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable when %{FIELD} is PC. Fields:

FIELD enum

Field name.

ArchMsg.Warning.decode_unpreduseofr13

TAGS UNPREDICTABLE DISPLAY Instruction is unpredictable when %{FIELD} is R13. Fields:

FIELD enum

Field name.

ArchMsg.Warning.decode_unpreduseofr13inreglist

TAGS UNPREDICTABLE DISPLAY Instruction is unpredictable with R13 in register-list.

ArchMsg.Warning.decode_useofd16orhigher

TAGS UNPREDICTABLE DISPLAY Attempt to use register D16 or higher where it is not allowed.

ArchMsg.Warning.decode_writebackandbaseinlist

TAGS UNPREDICTABLE DISPLAY Instruction is unpredictable with write back when the base register is in the transfer list.

ArchMsg.Warning.decode_zeroregistersinlist

TAGS UNPREDICTABLE DISPLAY Instruction is unpredictable when no registers are to be transferred.

ArchMsg.Warning.recursive_branch

An instruction is performing a branch that targets the same instruction. If this was intended then a WFI might be more effective DISPLAY recursive branch detected at PC=%{PC}. PRIMARY KEY PC. Fields:

PC unsigned int

Address of instruction causing the branch.

ArchMsg.Warning.recursive_exception

An instruction has generated an exception that targets the same instruction. DISPLAY recursive %{TYPE} exception detected at PC=%{PC} (CPSR=%{CPSR}). PRIMARY KEY PC. Fields:

CPSR unsigned int

Processor state.

PC unsigned int

Address of instruction causing the exception.

TYPE enum

Type of exception.

ArchMsg.Warning.reentrant_vector_catch

TAGS UNPREDICTABLE DISPLAY Vector catch debug event on Prefetch or Data abort vector %{ADDR}. Fields:

ADDR unsigned int

vector address.

ArchMsg.Warning.reserved_it_state

TAGS UNPREDICTABLE DISPLAY Execution with reserved IT state %{ITSTATE}. Fields:

ITSTATE unsigned int

reserved IT state.

ArchMsg.Warning.secure_vector_fetch_from_nonsecure

DISPLAY Secure vector table is being fetched from Non-secure memory.

ArchMsg.Warning.unpred_FAR_on_LRCPC3_non_atomic_mem_access

TAGS **UNPREDICTABLE** Unpredictable FAR/FSTATUS on LRCPC3 pair instructions resulting in two separate memory accessesDISPLAY Unpredictable FAR on LRCPC3 non atomic memory access at %{ADDR1} with FSTATUS %{FSTATUS1} or %{ADDR2} with FSTATUS %{FSTATUS2}. Fields:

ADDR1 unsigned int

Possible FAR address 1.

ADDR2 unsigned int

Possible FAR address 2.

FSTATUS1 unsigned int

Possible fault status value 1.

FSTATUS2 unsigned int

Possible fault status value 2.

ArchMsg.Warning.unpred_async_tag_check_fail_on_sve_speculative_load_do_update_ffr

Unpredictable set to FALSE of the FFR element associated to SVE speculative load with an async tag check fail TAGS **UNPREDICTABLE**.

ArchMsg.Warning.unpredictable_a32_breakpoint

TAGS **UNPREDICTABLE** DISPLAY Unpredictable breakpoint on a A32 instruction at %{ADDR1} Offset %{OFFSET} BAS %{BAS} for a %{IS_ADDRESS_MISMATCH_BREAKPOINT:(Mismatch|Match)} Breakpoint number %{N:d} Breakpoint address %{ADDR2}. Fields:

ADDR1 unsigned int

Address.

ADDR2 unsigned int

Breakpoint address.

BAS unsigned int

BAS field.

IS_ADDRESS_MISMATCH_BREAKPOINT bool

Is a Mismatch breakpoint, not a Match breakpoint.

N unsigned int

Breakpoint number.

OFFSET unsigned int

Always has a value of zero.

ArchMsg.Warning.unpredictable_banked_register_access

TAGS **UNPREDICTABLE** DISPLAY Access to banked register (R=%{R:d} SYSm=%{SYSM:2x}) is unpredictable. Fields:

R unsigned int

R bit.

SYSM unsigned int

SYSm.

ArchMsg.Warning.unpredictable_ccfail_undef

TAGS **UNPREDICTABLE** DISPLAY An instruction %{OPCODE} at VA=%{VADDR} failed its condition codes check but would otherwise have caused a non-data-dependent trap or undefined exception. Behaviour is implementation-defined choice of exception or nop. Fields:

OPCODE unsigned int

opcode of the instruction.

VADDR unsigned int

Virtual address of the instruction being executed.

ArchMsg.Warning.unpredictable_ici_bits

TAGS **UNPREDICTABLE** DISPLAY ICI bits do not apply or are architecturally unpredictable for instruction.

ArchMsg.Warning.unpredictable_t32_breakpoint

TAGS **UNPREDICTABLE** DISPLAY Unpredictable breakpoint on a T32 instruction at %{ADDR1} Offset %{OFFSET} BAS %{BAS} for a %{IS_ADDRESS_MISMATCH_BREAKPOINT:(Mismatch|Match)} Breakpoint number %{N:d} Breakpoint address %{ADDR2}. Fields:

ADDR1 unsigned int

Address.

ADDR2 unsigned int

Breakpoint address.

BAS unsigned int

BAS field.

IS_ADDRESS_MISMATCH_BREAKPOINT bool

Is a Mismatch breakpoint, not a Match breakpoint.

N unsigned int

Breakpoint number.

OFFSET unsigned int

Indicates which halfword of a 32-bit Thumb instruction the breakpoint is set on. 0 for the first halfword or if the instruction is 16 bits wide, or 1 for the second halfword.

ArchMsg.Warning.unpredictable_t32_vector_catch

TAGS **UNPREDICTABLE** DISPLAY Unpredictable Vector catch exception on a T32 instruction at %{ADDR} Offset %{OFFSET}. Fields:

ADDR unsigned int

Address.

OFFSET unsigned int

Indicates which halfword of a 32-bit Thumb instruction the vector catch exception is set on. 0 for the first halfword or if the instruction is 16 bits wide, or 1 for the second halfword.

ArchMsg.Warning.unpredictable_watchpoint_far

TAGS **UNPREDICTABLE** DISPLAY The value for FAR/EDWAR for hitting this watchpoint may be anywhere in the range %{LOWER_BOUND} to %{UPPER_BOUND}. Fields:

LOWER_BOUND unsigned int

the lowest address accessed by the instruction that triggered the watchpoint.

UPPER_BOUND unsigned int

the highest watchpointed address accessed by that instruction.

ArchMsg.Warning.unpredictable_watchpoint_far_isscmp

DISPLAY The value for FAR/EDWAR for hitting this watchpoint may be anywhere in the range %{LOWER_BOUND} to %{UPPER_BOUND}. Fields:

LOWER_BOUND unsigned int

the lowest address accessed by the instruction that triggered the watchpoint.

UPPER_BOUND unsigned int

the highest watchpointed address accessed by that instruction.

ArchMsg.Warning.warning_AdvSIMDExpandImmUnexpectedZero

TAGS **UNPREDICTABLE** DISPLAY AdvSIMDExpandImm may treat this immediate value as **UNPREDICTABLE**.

ArchMsg.Warning.warning_ConditionalSMC

TAGS **UNPREDICTABLE** DISPLAY SMC instruction has **UNPREDICTABLE** effects when conditional and when combined with traps.

ArchMsg.Warning.warning_access_crosses_page_boundary

TAGS **UNPREDICTABLE** DISPLAY Access crosses page boundary.

ArchMsg.Warning.warning_access_crosses_page_boundary_has_fault

TAGS **UNPREDICTABLE** DISPLAY Access that crosses page boundary has fault(s).

ArchMsg.Warning.warning_access_crosses_page_boundary_has_fault_on_both_pages

TAGS **UNPREDICTABLE** DISPLAY Access that crosses page boundary has a fault on each page. Fields:

ADDR1 unsigned int

FAR address on first page.

ADDR2 unsigned int

FAR address on second page.

FSTATUS1 unsigned int

Fault status on first page.

FSTATUS2 unsigned int

Fault status on second page.

ArchMsg.Warning.warning_access_crosses_page_boundary_spanning_different_memory

TAGS UNPREDICTABLE DISPLAY Access at address %{ADDR} crosses page from MEM_TYPE %{MEMTYPE_PAGE1} to %{MEMTYPE_PAGE2}. Fields:

ADDR unsigned int

address of access crossing page.

MEMTYPE_PAGE1 signed int

First Page's memory type (0-Normal, 1-Non-Normal).

MEMTYPE_PAGE2 signed int

Second Page's memory type (0-Normal, 1-Non-Normal).

ArchMsg.Warning.warning_access_wraps_around_memory

TAGS UNPREDICTABLE DISPLAY Access wraps around memory in %{WIDTH:d}bit mode. Fields:

WIDTH unsigned int

Address width.

ArchMsg.Warning.warning_bcr_linking_status

TAGS UNPREDICTABLE DISPLAY DBGBCR.{BT2,BT} combination gives unpredictable behaviour.

ArchMsg.Warning.warning_bcr_mask_reserved

TAGS UNPREDICTABLE DISPLAY DBGBCR_ELn.MASK is set to a reserved value.

ArchMsg.Warning.warning_bvr_masked_bit_not_zero

TAGS UNPREDICTABLE DISPLAY DBGBVRn_EL1.VA contains a bit masked by DBGBCRn_EL1.MASK that is not zero.

ArchMsg.Warning.warning_ccsidr_unimplemented_level

TAGS UNPREDICTABLE DISPLAY CCSIDR read while CSSELR %{CSSELR:x} points to an unimplemented cache level. Fields:

CSSELR unsigned int

current effective value of CSSELR.

ArchMsg.Warning.warning_change_to_ns_when_tge_set

TAGS UNPREDICTABLE DISPLAY Attempting to change NS to 1 when HCR.TGE = 1.

ArchMsg.Warning.warning_contiguous_bit_check_abort

TAGS UNPREDICTABLE DISPLAY An abort occurred whilst attempting to check TLB entry is contiguous at %{ENTRY_ADDR}. Fields:

ENTRY_ADDR unsigned int

address of conflicting TLB entry.

ArchMsg.Warning.warning_contiguous_bit_error

TAGS **UNPREDICTABLE** DISPLAY TLB entry at %{ENTRY_ADDR} was not contiguous with entry at %{CONTIG_ADDR} Contents are %{ENTRY_DATA} but expected %{CONTIG_DATA}. Fields:

CONTIG_ADDR unsigned int

address of the TLB first read, with which this entry is expected to be contiguous.

CONTIG_DATA unsigned int

expected contents based on the entry at CONTIG_ADDR.

ENTRY_ADDR unsigned int

address of conflicting TLB entry.

ENTRY_DATA unsigned int

contents of conflicting TLB entry.

ArchMsg.Warning.warning_contiguous_bit_error_D128

TAGS **UNPREDICTABLE** DISPLAY TLB entry at %{ENTRY_ADDR} was not contiguous with entry at %{CONTIG_ADDR} Contents are %{ENTRY_DATA_HI} %{ENTRY_DATA_LO} but expected %{CONTIG_DATA}. Fields:

CONTIG_ADDR unsigned int

address of the TLB first read, with which this entry is expected to be contiguous.

CONTIG_DATA unsigned int

expected contents based on the entry at CONTIG_ADDR.

ENTRY_ADDR unsigned int

address of conflicting TLB entry.

ENTRY_DATA_HI unsigned int

Higher bit contents of conflicting TLB entry.

ENTRY_DATA_LO unsigned int

Lower bit contents of conflicting TLB entry.

ArchMsg.Warning.warning_cp10_cp11_mismatch

TAGS **UNPREDICTABLE** DISPLAY Unpredictable configuration of CP10 and CP11 controls at EL %{EL:d}. Fields:

EL unsigned int

EL owning the control.

ArchMsg.Warning.warning_cp10_cp11_reserved_value

TAGS **UNPREDICTABLE** DISPLAY Unpredictable configuration of CPACR CP10 (%{CP10}) or CP11 (%{CP11}). Fields:

CP10 unsigned int

CP10 access.

CP11 unsigned int

CP11 access.

ArchMsg.Warning.warning_csselr_level_out_of_range

TAGS **UNPREDICTABLE** DISPLAY CSSELR.Level out of range, written value %{LEVEL} with only %{IMPLEMENTED:d} levels implemented. Fields:

IMPLEMENTED unsigned int

number of cache levels implemented.

LEVEL unsigned int

written value of CSSELR.Level.

TYPE enum

The type of cache selected by CSSELR.InD and TnD.

ArchMsg.Warning.warning_debug_flow_control_bits_not_obeyed

TAGS **UNPREDICTABLE** DISPLAY Flow control bits for %{REGISTER} were not obeyed, causing %{ERROR}. Fields:

ERROR enum

type of error resulting.

REGISTER enum

register under consideration.

ArchMsg.Warning.warning_debug_register_access_during_reset

TAGS **UNPREDICTABLE** DISPLAY Unpredictable %{IS_WRITE:(read|write)} access to debug register offset %{OFFSET} during reset. Fields:

IS_WRITE unsigned int

Write Not Read.

OFFSET unsigned int

Register Offset.

ArchMsg.Warning.warning_decode_cps_inconsistent_fields

TAGS **UNPREDICTABLE** DISPLAY The mode setting fields in the CPS instruction are inconsistent.

ArchMsg.Warning.warning_decode_invalid_state

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable in current exception-level/security state.

ArchMsg.Warning.warning_default_cacheable_mmu_on

TAGS **UNPREDICTABLE** DISPLAY Default cacheable is enabled (HCR.DC = 1) while MMU is enabled.

ArchMsg.Warning.warning_default_cacheable_vmmu_off

TAGS **UNPREDICTABLE** DISPLAY Default cacheable is enabled (HCR.DC = 1) while HCR.VM = 0.

ArchMsg.Warning.warning_deprecated_wvr_bit_2_set

TAGS **UNPREDICTABLE** DISPLAY DBGWVR%{N:d}_EL1=%{WVR:h} has bit 2 set, deprecated in ARMv8. Fields:

N unsigned int

index.

WVR unsigned int

DBGWVR.

ArchMsg.Warning.warning_eret_while_software_step_active_pending

Missing ISB between setting MDSCR_EL1.SS and ERET TAGS **UNPREDICTABLE**.

ArchMsg.Warning.warning_exclusive_to_non_normal

TAGS **UNPREDICTABLE** DISPLAY Exclusive Access to Non-Normal Memory %{ADDR}. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_exclusive_to_non_writeback

TAGS **UNPREDICTABLE** DISPLAY Exclusive Access to Non-Writeback-Cacheable Memory %{ADDR}. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_execute_from_device_memory

TAGS **UNPREDICTABLE** DISPLAY An attempt was made to execute from Device memory.

ArchMsg.Warning.warning_illegal_cpsr_mode

TAGS **UNPREDICTABLE** DISPLAY Writing an illegal or unimplement mode (%{NEW_MODE:h}) to CPSR was unpredictable. Fields:

NEW_MODE unsigned int

New value of CPSR.M.

ArchMsg.Warning.warning_illegal_srs_mode

TAGS **UNPREDICTABLE** DISPLAY Illegal or **UNPREDICTABLE** mode (%{MODE:h}) used for SRS instruction. Fields:

MODE unsigned int

mode for Banked SP.

ArchMsg.Warning.warning_implementation_defined_read_debug_register_in_SCS

Read Access to Debug registers from the processor is IMPLEMENTATION DEFINED TAGS IMP_DEF. Fields:

PPB_OFFSET unsigned int
debug register offset.

ArchMsg.Warning.warning_implementation_defined_sequential_security_transitions_supported

The behavior of sequential instruction fetches that cross from non-secure to secure memory and contain an SG instruction is **CONSTRAINED UNPREDICTABLE TAGS UNPREDICTABLE**.

ArchMsg.Warning.warning_implementation_defined_stack_limit_check_supported

It is **IMPLEMENTATION DEFINED** whether stack pointer limit checking is performed for this instruction TAGS IMP_DEF. Fields:

ADDRESS unsigned int
Address of instruction.

OPCODE unsigned int
instruction opcode.

ArchMsg.Warning.warning_implementation_defined_taggednonsharedmemorytype

TAGS IMP_DEF DISPLAY Memory attribute has Tagged, non shared memory type; it is **IMPLEMENTATION DEFINED** whether the memory location is treated as Tagged or non-Tagged.

ArchMsg.Warning.warning_implementation_defined_write_debug_register_in_SCS

Write Access to Debug registers from the processor is **IMPLEMENTATION DEFINED** TAGS IMP_DEF. Fields:

DATA unsigned int
data attempted to be written.

PPB_OFFSET unsigned int
debug register offset.

ArchMsg.Warning.warning_invalid_tcr_granule

TAGS **UNPREDICTABLE** DISPLAY TCR.TG%{TG_ID:(0|1)} bits have been set to a granule size %{{REQUEST:d}K not implemented - using %{{SUBSTITUTE:d}K. Fields:

REQUEST signed int
page size requested (or 0 for reserved).

SUBSTITUTE signed int
best guess available page size.

TG_ID bool
bits TG0 or TG1.

ArchMsg.Warning.warning_load_multiple_user_registers_from_user_mode

TAGS **UNPREDICTABLE** DISPLAY An LDM(user registers) instruction executed from user mode.

ArchMsg.Warning.warning_mte_in_mem_attr_not_supported

TAGS **UNPREDICTABLE** DISPLAY Memory attribute %{{ATTR}} has Tagged memory type but Armv8.5-MTE is not implemented. Fields:

ATTR unsigned int

Memory attribute.

ArchMsg.Warning.warning_reserved_breakpoint_state_match

TAGS **UNPREDICTABLE** DISPLAY **%{IS_BREAKPOINT:(Watchpoint|Breakpoint)}** programmed with a reserved combination of HMC, SSC and **%{IS_BREAKPOINT:(PAC|PMC)}**. Fields:

IS_BREAKPOINT unsigned int

Is a breakpoint, not a watchpoint.

ArchMsg.Warning.warning_shareability

TAGS **UNPREDICTABLE** DISPLAY Unpredictable: combination of the 1st and 2nd stages of translation is Normal Inner Non-Cacheable, Outer Non-Cacheable.

ArchMsg.Warning.warning_software_step_set_while_enabled

TAGS **UNPREDICTABLE** DISPLAY MDSCR_EL1.SS set to 1 while software step debug exceptions are enabled.

ArchMsg.Warning.warning_thumb_instruction_wraps_around_memory

TAGS **UNPREDICTABLE** DISPLAY Thumb instruction wraps around memory.

ArchMsg.Warning.warning_ttbr_sbz_bits_are_not_zero

TAGS **UNPREDICTABLE** DISPLAY **%{TTBR}[%(MSB:d);0]** should be zero, but are not. Fields:

MSB unsigned int

x-1.

TTBR enum

which TTBR register.

ArchMsg.Warning.warning_unaligned_address_dbgdtrrx_write

TAGS **UNPREDICTABLE** DISPLAY Unaligned address **%{ADDR}** in X0/R0 when DBGDTRRX written in memory access mode. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unaligned_address_dbgdtrtx_read

TAGS **UNPREDICTABLE** DISPLAY Unaligned address **%{ADDR}** in X0/R0 when DBGDTRTX read in memory access mode. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unaligned_to_device

TAGS **UNPREDICTABLE** DISPLAY Unaligned Access to Device Memory **%{ADDR}**. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unaligned_to_strongly_ordered

TAGS UNPREDICTABLE DISPLAY Unaligned Access to Strongly Ordered Memory %{ADDR}.

Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unknown_sau_rnr

TAGS UNPREDICTABLE DISPLAY SAU_RNR was set to an unsupported value.

ArchMsg.Warning.warning_unpredictable_AIRCR_PRIS_and_BFHFNMINs

TAGS UNPREDICTABLE DISPLAY The effect of setting both AIRCR.BFHFNMINs and AIRCR.PRIS to 1 is UNPREDICTABLE.

ArchMsg.Warning.warning_unpredictable_AIRCR_VECTCLRACTIVE_when_not_in_debug

TAGS UNPREDICTABLE DISPLAY The effect of writing a 1 to AIRCR.VECTCLRACTIVE if the processor is not halted in Debug state is UNPREDICTABLE.

ArchMsg.Warning.warning_unpredictable_AIRCR_VECTRESET_and_SYSRESETREQ

TAGS UNPREDICTABLE DISPLAY When the processor is halted in Debug state, if a write to the register writes a 1 to both VECTRESET and SYSRESETREQ, the behavior is UNPREDICTABLE.

ArchMsg.Warning.warning_unpredictable_AIRCR_VECTRESET_when_not_in_debug

TAGS UNPREDICTABLE DISPLAY The effect of writing a 1 to AIRCR.VECTRESET if the processor is not halted in Debug state is UNPREDICTABLE.

ArchMsg.Warning.warning_unpredictable_AIRCR_incorrect_VKEY

TAGS UNPREDICTABLE DISPLAY The value 0x05FA must be written to AIRCR.VKEY, otherwise the register write is UNPREDICTABLE.

ArchMsg.Warning.warning_unpredictable_EXC_RETURN_Reserved_Bit

EXC_RETURN[23:7] are reserved with the special condition that all bits should be written as one. Values other than all 1s are UNPREDICTABLE. TAGS UNPREDICTABLE. Fields:

EXC_RETURN unsigned int

Special PC value.

SBO_MASK unsigned int

Which bits Should Be One to not UNPRED.

ArchMsg.Warning.warning_unpredictable_PMEVTYPEPn_ELO_VS_reserved_value

TAGS UNPREDICTABLE DISPLAY Unpredictable reserved value of PMEVTYPEP<n>_ELO.VS.

ArchMsg.Warning.warning_unpredictable_change_to_DEMCR_MON_STEP_at_insufficient_priority

TAGS UNPREDICTABLE DISPLAY The effect of changing DEMCR.MON_STEP at an execution priority that is lower than the priority of the DebugMonitor exception is **CONSTRAINED** UNPREDICTABLE.

ArchMsg.Warning.warning_unpredictable_change_to_priority_of_active_exception

TAGS UNPREDICTABLE DISPLAY Changing the priority of an active exception is UNPREDICTABLE.

ArchMsg.Warning.warning_unpredictable_clear_lspact

TAGS UNPREDICTABLE DISPLAY UNPREDICTABLE state on context switch. Fields:

DRVEXC_TAKEN unsigned int

Derived exception taken, not pended.

REACHED_LAST_STORE unsigned int

Only the last faulted.

ArchMsg.Warning.warning_unpredictable_dbg_exit_unaligned_dlr

TAGS UNPREDICTABLE DISPLAY Debug exit to AArch32 with DLR[0] set to 1. PC[0] can be set to 0 or to DLR[0].

ArchMsg.Warning.warning_unpredictable_exception_catch

TAGS UNPREDICTABLE DISPLAY Generation of an exception catch event is unpredictable.

ArchMsg.Warning.warning_unpredictable_exception_return_inconsistent_state

TAGS UNPREDICTABLE DISPLAY state on exception return is unpredictable.

ArchMsg.Warning.warning_unpredictable_exception_return_instruction

TAGS UNPREDICTABLE DISPLAY exception return instruction is unpredictable.

ArchMsg.Warning.warning_unpredictable_in_debug_state

TAGS UNPREDICTABLE DISPLAY Instruction is UNPREDICTABLE when executed in debug state.

ArchMsg.Warning.warning_unpredictable_mair_encoding

TAGS UNPREDICTABLE DISPLAY Memory attribute %{ATTR} has invalid MAIR encoding; this may lead to unpredictable results. Fields:

ATTR unsigned int

Memory attribute.

ArchMsg.Warning.warning_unpredictable_pmu_counter_access

TAGS UNPREDICTABLE DISPLAY Access to PMU counters from non-secure EL0 or EL1 is UNPREDICTABLE with MDCR_EL2.HPMN set to 0.

ArchMsg.Warning.warning_unpredictable_prioritization_breakpoint_match_vector_catch

TAGS UNPREDICTABLE DISPLAY Two events with the same priority occurred at the same instruction: Address Matching Vector Catch debug event, and Breakpoint debug event. It is constrained unpredictable which is taken.

ArchMsg.Warning.warning_unpredictable_stack_selection

TAGS UNPREDICTABLE DISPLAY UNPREDICTABLE state on context switch.

ArchMsg.Warning.warning_unpredictable_tsize_out_of_range

TAGS UNPREDICTABLE DISPLAY %{TCR}.T%{TTBR:d}SZ (%{TSIZE:d} bits) is out of range. Must be between 25 and %{TSIZE_MAX:d}. Fields:

TCR enum

Which TCR.

TSIZE signed int

Value of TSize.

TSIZE_MAX signed int

The maximum allowed value.

TTBR unsigned int

Which TTBR region.

ArchMsg.Warning.warning_unpredictable_unaligned_pc_as_base_register

TAGS **UNPREDICTABLE** DISPLAY The PC value must be word-aligned, otherwise the behavior of the instruction is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_unaligned_pop_stack

TAGS **UNPREDICTABLE** DISPLAY **CONSTRAINED UNPREDICTABLE** unaligned stack frame
 %{FRAME_ADDR:x} in exception-return. Fields:

FRAME_ADDR unsigned int

unaligned stack frame address.

ArchMsg.Warning.warning_unpredictable_unaligned_to_non_writeback

TAGS **UNPREDICTABLE** DISPLAY Unaligned Access to Non-Writeback-Cacheable Memory.

ArchMsg.Warning.warning_unpredictable_vmsa_memattrib

TAGS **UNPREDICTABLE** DISPLAY Unpredictable vmsa memory attribute with texcb value
 programmed to %{TEXCB} when tex remap is %{TEX_REMAP}. Fields:

TEXCB unsigned int

texcb value.

TEX_REMAP bool

use tex remap.

ArchMsg.Warning.warning_unpredictable_vtcr_t0sz_sl0

TAGS **UNPREDICTABLE** DISPLAY The combination of VTCR.T0SZ=%{T0SZ:d} and VTCR.SL0=
 %{SL0:d} is unpredictable. Fields:

SL0 signed int

VTCR.SL0, starting level for stage 2 translation table walks.

T0SZ signed int

VTCR.T0SZ, required input address range.

ArchMsg.Warning.warning_unpredictable_write_DHCSR

TAGS **UNPREDICTABLE** DISPLAY **UNPREDICTABLE** set of bit-changes in DHCSR. Fields:

NEW unsigned int

data attempted to be written.

OLD unsigned int

previous value.

ArchMsg.Warning.warning_unpredictable_write_SHCSR

TAGS **UNPREDICTABLE** DISPLAY **UNPREDICTABLE** setting a pending bit to 1 for an exception
 with priority >= execution priority.

ArchMsg.Warning.warning_unsupported_access_to_memory_mapped_register

TAGS **UNPREDICTABLE** DISPLAY Access to memory mapped register is unsupported.

ArchMsg.Warning.warning_user_jmcr_access

TAGS UNPREDICTABLE DISPLAY User %{WRITE:(write|read)} access to JMCR is unpredictable.
Fields:

WRITE unsigned int
access is write.

ArchMsg.Warning.warning_wcr_mask_and_bas

TAGS UNPREDICTABLE DISPLAY DBGWCR_EL1.MASK is non-zero and BAS is not 0xff.

ArchMsg.Warning.warning_wcr_mask_reserved

TAGS UNPREDICTABLE DISPLAY DBGWCR_ELn.MASK is set to a reserved value.

ArchMsg.Warning.warning_wcr_non_configuous_bas

TAGS UNPREDICTABLE DISPLAY DBGWCR_EL1.BAS has non-contiguous set of ones.

ArchMsg.Warning.warning_wvr_masked_bit_not_zero

TAGS UNPREDICTABLE DISPLAY DBGWVRn_EL1.VA contains a bit masked by
DBGWCRn_EL1.MASK that is not zero.

BRANCH_MISPREDICT

Simulating branch mispredict. Fields:

PC unsigned int
Origin address (or 0 if unavailable).

BRA_DIR

Direct branches, to immediate address. Fields:

CORE_NUM unsigned int
Core number in a multi processor.

INST_COUNT unsigned int
The core's instruction counter, starting at 1 for the first instruction.

ISSET enum
The instruction set of the branch instruction.

IS_COND bool
Indicates if this is a conditional branch.

IS_HINTED bool
Whether the conditional instruction is hinted.

IS_ISB bool
Whether the instruction is ISB.

IS_LINK bool
Indicates if this is a branch with link.

PC unsigned int
The address of the branch instruction.

TARGET_ISET enum

The instruction set after the branch.

TARGET_PC unsigned int

The address the instruction branches to.

BRA_INDIR

Indirect branches, perhaps to a register. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISET enum

The instruction set of the branch instruction.

IS_COND bool

Indicates if this is a conditional branch.

IS_HINTED bool

Whether the conditional instruction is hinted.

IS_ISB bool

Whether the instruction is ISB.

IS_LINK bool

Indicates if this is a branch with link.

PC unsigned int

The address of the branch instruction.

TARGET_ISET enum

The instruction set after the branch.

TARGET_PC unsigned int

The address the instruction branches to.

BRA_INDIR_CCFAIL

Indirect branches fail, perhaps to a register.

BRA_RET_CCFAIL

Branch return fail, perhaps to a register.

CACHE_MAINTENANCE_OP

Cache Maintenance Operation. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

DATA unsigned int

Specified MVA or Set/way.

FUNCTION enum

Clean / Invalidate.

SCOPE enum

Affected region.

SIDE enum

Inst / Data.

CCFAIL

Conditional instruction condition check fail. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CCFAIL_UNC

Unconditional instruction condition check fail. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CCPASS

Conditional instruction condition check pass. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CCPASS_UNC

Unconditional instruction condition check pass. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CODE_CACHE_MAINT

Code cache maintenance. Fields:

END_INCL unsigned int

Inclusive end address for VA or PA requests.

NS bool

Non-secure world for PA.

START unsigned int

Start address for VA or PA requests.

TYPE enum

Request type.

COMPILE_BLOCK_END

Last instruction of basic block translated. Fields:

VADDR unsigned int

Address of next instruction after this basic block.

COMPILE_BLOCK_START

First instruction of basic block translated. Fields:

VADDR unsigned int

Address of first instruction in basic block.

COMPILE_INST

ARM instruction compiled. Fields:

DISASS string

Disassembly of the instruction.

ISSET enum

The instruction set of this instruction.

ITSTATE unsigned int

The ITSTATE current for the instruction.

OPCODE unsigned int

The opcode of the instruction.

PC unsigned int

The address of the instruction.

SIZE unsigned int

The size of the instruction in bytes.

CONTEXTIDR

Context ID Register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

NS bool

Secure or nonsecure banked register is accessed.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The new value written.

CONTEXT_SYNC

Called for every context synchronization event. Fields:

ARCH_CSE bool

Architectural Context Synchronization Event.

PC unsigned int

The address of the synchronization instruction.

SYSREG_SYNC bool

System register synchronization event.

CORE_ENDIAN

Core BE8 Big-Endian state changed. Fields:

BE8 bool

Core BE8 Big-Endian state.

CORE_INFO

Static processor attributes. Only triggered by a call to DumpState(). Fields:

ARCH_PROFILE enum

The architecture profile of the core.

CLUSTER_ID unsigned int

The cluster ID of this processor.

CORE_NUM unsigned int

The number of this core in an MP processor.

DETAILED_CORE_FEATURES enum

Detailed core features as a bitfield. Enum values are bitmasks to extract the info from the field.

FPU_VERSION enum

The VFP version implemented by the core.

MEM_ARCH enum

The memory architecture of the core.

MIDR_EL1 unsigned int

MIDR_EL1 value.

NUM_CORES unsigned int

The number of cores in this MP processor.

QUANTUM_SIZE unsigned int

The default quantum size of the core (in ticks).

SECURITY_FEATURES bool

Does the core have security features?.

CORE_LOADS

Processor load accesses. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ACQREL enum

Is this an acquire/release.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DATA unsigned int

The data read or written.

ELEMENT_SIZE unsigned int

Width of each element.

ENDIANNESS enum

Byte ordering of this access.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

LOCK enum

Normal, exclusive or locked access.

MECID unsigned int

Memory Encryption Context ID, if applicable.

MECID2 unsigned int

Memory Encryption Context ID of the second page, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PAGE_SIZE unsigned int

Page size as $\log_2(\text{size in bytes})$.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK, 2=Exclusive Failed, 3=Unsupported.

SIZE unsigned int

Width of the access in bytes. Only required if DATA is not traced.

TRANS bool

Is this a translated access.

VADDR unsigned int

The virtual address of the access.

CORE_REGS

Changes of the core registers R0 to R14. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number, 0 to 14.

MODE enum

Bank of the register accessed.

OLD_VALUE unsigned int

The old value overwritten.

PHYS_ID enum

The physical register accessed.

VALUE unsigned int

The new value written to the register.

CORE_STALLED_TRANSITION

Detect when a core is prevented from progressing PChannel states. Fields:

CURRENT_STATE enum

Current power mode of core.

REASON string

The reason a transition was stalled, and what must be done to unblock it.

TARGET_STATE enum

Desired power mode of core after transition.

CORE_STORES

Processor store accesses. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ACQREL enum

Is this an acquire/release.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DATA unsigned int

The data read or written.

ELEMENT_SIZE unsigned int

Width of each element.

ENDIANNESS enum

Byte ordering of this access.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

LOCK enum

Normal, exclusive or locked access.

MECID unsigned int

Memory Encryption Context ID, if applicable.

MECID2 unsigned int

Memory Encryption Context ID of the second page, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PAGE_SIZE unsigned int

Page size as $\log_2(\text{size in bytes})$.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK, 2=Exclusive Failed, 3=Unsupported.

SIZE unsigned int

Width of the access in bytes. Only required if DATA is not traced.

TRANS bool

Is this a translated access.

VADDR unsigned int

The virtual address of the access.

CP14_READ

System Coprocessor register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP14 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc1 unsigned int

opcode 1.

opc2 unsigned int

opcode 2.

CP14_WRITE

System Coprocessor register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP14 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc1 unsigned int

opcode 1.

opc2 unsigned int

opcode 2.

CP15_READ

System Coprocessor register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP15 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc1 unsigned int

opcode 1.

opc2 unsigned int

opcode 2.

CP15_READ64

System Coprocessor register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP15 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc unsigned int

opcode 1.

CP15_WRITE

System Control Coprocessor register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP15 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc1 unsigned int

opcode 1.

opc2 unsigned int
opcode 2.

CP15_WRITE64

System Coprocessor register write. Fields:

CORE_NUM unsigned int
Core number in a multi processor.

CRm unsigned int
CRm.

NS bool
Secure or nonsecure banked register is accessed.

REG_NAME string
Name of the CP15 register accessed.

SECURITY_STATE enum
The current security state.

UNDEF bool
The register accessed is undefined.

UPDATED_VALUE unsigned int
Updated value of the register now it has been written.

VALUE unsigned int
The new value written.

opc unsigned int
opcode 1.

CPSR

CPSR change. (In AArch64 PSTATE represented in SPSR format). Fields:

CORE_NUM unsigned int
Core number in a multi processor.

OLD_VALUE unsigned int
The old CPSR value.

UNKNOWN unsigned int
Bits within the register that have unknown value.

VALUE unsigned int
The new CPSR value.

CRYPTO_SPEC

Every crypto instruction speculatively executed.

CorePowerStateChange

Core power-up state has changed. DISPLAY Core powered %{PU:{up|down}}. Fields:

PU bool

Whether the core powered up.

DATA_CACHE_SET_ALLOC_TAGS

Set allocation tags from DC GVA instruction. Fields:

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

NSDESC unsigned int

The security state of the access.

PADDR unsigned int

The physical address of the access.

PAS enum

The physical address space of the page.

RESPONSE enum

0=Aborted, 1=OK.

SIZE unsigned int

The size of memory in bytes.

TAG unsigned int

The value of the tag being set.

TAGGING_TYPE enum

Type of Memory tagging (Physical or Virtual).

TAG_VADDR unsigned int

The virtual address of the Virtual Tag if Virtual tagging is used; otherwise 0.

VADDR unsigned int

Virtual address used for setting allocation tags.

DATA_CACHE_ZERO

Zero and invalidate cache line from DC ZVA instruction. Fields:

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

NSDESC unsigned int

The security state of the access.

PADDR unsigned int

The physical address of the access.

PADDR_TAG unsigned int

If tag checked, the physical address tag for the access.

PAS enum

The physical address space of the page.

RESPONSE enum

0=Aborted, 1=OK.

SIZE unsigned int

The size of zero'd memory in bytes.

TAG_CHECKED bool

Is this access tag checked.

VADDR unsigned int

Virtual address of the zero'd cache line.

DEBUG_EVENT

Hardware debug support event. Fields:

EVENT enum

Description of event.

VALUE unsigned int

data value.

DEBUG_TARGET_ISET

Hardware debug target ISA. Fields:

TARGET_ISET enum

The target instruction set of the debug state.

DMI_ALLOCATE

DMI allocated. Fields:

ACCESS_DIRECTION enum

READ / WRITE.

PADDR unsigned int

The physical address of the DMI.

PTR unsigned int

DMI range pointer.

SIDE enum

DSIDE / ISIDE.

SIZE unsigned int

The size of DMI.

VADDR unsigned int

The virtual address of the DMI..

DMI_HIT

DMI hit, access used DMI (only fire for small DMI regions i.e. less than a full page). Fields:

ACCESS_DIRECTION enum

READ / WRITE.

PADDR unsigned int

The physical address of the DMI.

SIDE enum

DSIDE / ISIDE.

SIZE unsigned int

The size of the DMI.

VADDR unsigned int

The virtual address of the DMI..

DMI_REVOKE

DMI revoked. Fields:

ACCESS_DIRECTION enum

READ / WRITE.

PADDR unsigned int

The physical address of the DMI.

SIDE enum

DSIDE / ISIDE.

SIZE unsigned int

The size of DMI.

VADDR unsigned int

The virtual address of the DMI..

DTCM_ACCESS

Trace TCM accesses.

DTCM_ACCESS_RAZ_WI

Trace TCM accesses. Fields:

ADDR unsigned int

Address of the **RAZ/WI** access.

END_COMPILE

Compilation end. Fields:

END_OF_PAGE_COUNT unsigned int

Number of basic blocks exited due to page boundary since **START_COMPILE**.

FETCHFAIL_COUNT unsigned int

Number of basic blocks exited due to ifetch failure START_COMPILE.

INST_COUNT unsigned int

Number of instructions compiled since START_COMPILE.

NONSEQ_COUNT unsigned int

Number of basic blocks exited due to non-sequential instructions since START_COMPILE.

PAGE_STRADDLE_COUNT unsigned int

Number of basic blocks exited due to unaligned instructions crossing page since START_COMPILE.

EXCEPTION

Exception taken. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ESR unsigned int

The value assigned to ESR when exception is taken.

ESR_EC unsigned int

The Value assigned to ESR.EC when exception is taken.

IS_PHYSICAL bool

Physical or Virtual exception.

IS_TRAP bool

Exception coming from a trap.

LR unsigned int

The value assigned to the link register.

PC unsigned int

The location where the exception occurred.

PREFERRED_RETURN unsigned int

The preferred return address for the exception.

TAKEN_LOCALLY bool

Whether exception is taken locally(same exception level).

TARGET_ISET enum

The instruction set of the exception handler code.

TARGET_PC unsigned int

The address the exception branches to.

VECTOR enum

The exception vector.

VECTOR_OFFSET unsigned int

The offset of an exception vector from the base address.

EXCEPTION_END

Every exception completed.

EXCEPTION_INFO

Trace human-readable reason of the current exception for data/instruction aborts, floating point exceptions and unknown exceptions. Fields:

DESCRIPTION string

Reason for current exception.

ESR_EC unsigned int

The value of ESR.EC when exception is taken.

EXCEPTION_RAISE

Every exception raised.

EXCEPTION_RETURN

Branches on leaving exception. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISSET enum

The instruction set of the branch instruction.

IS_COND bool

Indicates if this is a conditional branch.

IS_HINTED bool

Whether the conditional instruction is hinted.

IS_ISB bool

Whether the instruction is ISB.

IS_LINK bool

Indicates if this is a branch with link.

PC unsigned int

The address of the branch instruction.

TARGET_ISET enum

The instruction set after the branch.

TARGET_PC unsigned int

The address the instruction branches to.

EXCEPTION_RETURN_PREBRANCH

Exception return event that must occur before PC and processor state have updated.

EXCEPTION_START

Every exception started.

FP_EXCEPTIONS_SIGNALLED

Trace which FP exceptions were signalled as a part of the FP operations performed by an instruction. Fields:

FP_EXCEPTIONS_SIGNALLED_MASK unsigned int

Indicates which FP exceptions were signalled by an instruction. This uses the same encoding as the cumulative floating-point exception bitfields in FPSR.

FP_STATE

Floating point state. Fields:

FPSR unsigned int

FPSR register value.

MASK unsigned int

The mask indicating updated cumulative bits.

FREQ_CHANGED

Core frequency changed. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The instruction count of this core.

NEW_FREQ signed int

The new frequency of this core (expressed in Hz).

OLD_FREQ signed int

The old frequency of this core (expressed in Hz).

HLT

HLT instruction occurred. Fields:

IMM16 unsigned int

The 16-bit immediate value encoded in HLT instruction.

INST

Every instruction executed. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DEBUG_STATE bool

The instruction is executed in debug state.

DISASS string

Disassembly of instruction.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISET enum

The current instruction set.

ITSTATE unsigned int

The current ITSTATE.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

MODE enum

The mode the core is in.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

OPCODE unsigned int

The opcode of the instruction.

PADDR unsigned int

The physical address of the instruction.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the instruction.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the page.

PC unsigned int

The address of the instruction.

SECURITY_STATE enum

The core's security state.

SIZE unsigned int

The size of the instruction in bytes.

INST_END

Every instruction completed.

INST_START

Every instruction started. Fields:

DEBUG_STATE bool

The instruction is executed in debug state.

ISSET enum

The current instruction set.

MODE enum

The mode the core is in.

NS enum

The current Secure State.

PC unsigned int

The address of the conditional instruction.

SECURITY_STATE enum

The current Security State.

INST_STRADDLE

Instruction straddles boundary. Fields:

DEBUG_STATE bool

The instruction is executed in debug state.

ISSET enum

The current instruction set.

MODE enum

The mode the core is in.

NS enum

The current Secure State.

PC unsigned int

The address of the conditional instruction.

SECURITY_STATE enum

The current Security State.

ISB

ISB instruction occurred. Fields:

PC unsigned int

The address of the ISB instruction.

ITCM_ACCESS

Trace TCM accesses.

ITCM_ACCESS_RAZ_WI

Trace TCM accesses. Fields:

ADDR unsigned int

Address of the **RAZ/WI** access.

LOAD_MULTIPLE_REGS_LIST

List of destination registers (not source) of a multiple registers load instruction. Fields:

REG string

Register to add into the list.

LOCAL_MONITOR

Local monitor activity. Fields:

PADDR unsigned int

Local Monitor Address.

State enum

State of the monitor (Open/Exclusive).

MEMMAP_DEBUG_READ

Memory mapped reads to the debug registers. Fields:

ADDR unsigned int

address.

CORE_NUM unsigned int

Core number in a multi processor.

EXT bool

Whether access is from an external device (such as the DAP).

NS enum

Secure state of the access.

REG_NAME string

Name of the debug register accessed.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

MEMMAP_DEBUG_WRITE

Memory mapped writes to the debug registers. Fields:

ADDR unsigned int

address.

CORE_NUM unsigned int

Core number in a multi processor.

EXT bool

Whether access is from an external device (such as the DAP).

NS enum

Secure state of the access.

REG_NAME string

Name of the debug register accessed.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The new value written.

MODE_CHANGE

Mode change. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

MODE enum

The new mode.

OLD_MODE enum

The old mode.

MPU_TRANS

Address translation information. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

EXEC_PERM enum

Execution Permission.

INNERCACHE_RA bool

Is the inner cache allocate on read.

INNERCACHE_TYPE enum

Inner Caching scheme (NC/MB/WA).

INNERCACHE_WA bool

Is the inner cache allocate on write.

MEMTYPE enum

Memory type.

OUTERCACHE_RA bool

Is the outer cache allocate on read.

OUTERCACHE_TYPE enum

Outer Caching scheme (NC/MB/WA).

OUTERCACHE_WA bool

Is the outer cache allocate on write.

PADDR unsigned int

Address of the access.

PAGESIZE unsigned int

Page size as $\log_2(\text{size})$.

READ_PERM enum

Read Permission.

REG_NUM unsigned int

MPU register based region-number 0-15 (else 0xFF=none, 0xBB/0xDD=background/default).

SH enum

Shareability.

SIDE enum

Inst / Data.

WRITE_PERM enum

Write Permission.

PERIODIC

Called for every quantum. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The instruction count of this CPU.

PC unsigned int

The address of the next instruction to be executed on this CPU.

PMU_COUNTER_OVERFLOW

PMU counter overflow. Fields:

COUNTER_GROUP enum

Counter group.

COUNTER_RANGE enum

Counter range.

EVENT_ID unsigned int

Event Identifier, field is valid only for Event counter group.

INDEX unsigned int

Counter index (as selected by PMSELR), field is valid only for Event/Cycle counter groups.

INTERRUPT bool

Is interrupt enabled on overflow for this counter.

POTINDEX_UPDATE

Value of POT index for current pc changed. Fields:

NEW_POTINDEX unsigned int

New POTIndex.

OLD_POTINDEX unsigned int

Old POTIndex.

PRELOAD_DATA

Data preload from PLD instruction. Fields:

VADDR unsigned int

Virtual address of the data that should be preloaded.

PRELOAD_INST

Instruction preload from PLI instruction. Fields:

VADDR unsigned int

Virtual address of the instruction that should be preloaded.

PRE_CORE_LOAD

Trace just before a core load. Fields:

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

PRE_CORE_STORE

Trace just before a core store. Fields:

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

PSTATE

PSTATE fields update. Fields:

ALLINT unsigned int

All interrupt mask bit.

BTYPE unsigned int

Branch target identification bit.

CORE_NUM unsigned int

Core number in a multi processor.

DAIF unsigned int

Exception mask bits.

DIT unsigned int

Data Independent Timing bit.

EL unsigned int

Current Exception level field.

EXLOCK unsigned int

Exception return state lock bit.

IL unsigned int

Illegal Execution state bit.

NZCV unsigned int

Condition flags.

PACM unsigned int

PACM bit.

PAN unsigned int

Privileged Access Never state bit.

PM unsigned int

PMU exception mask bit.

PPEND unsigned int

PMU exception pending bit.

SM unsigned int

StreamingMode status bit.

SP unsigned int

Stack pointer register selection bit.

SS unsigned int

Software Step bit.

SSBS unsigned int

Speculative Store Bypass Safe bit.

TCO unsigned int

Tag Check Override bit.

TOUCHED unsigned int

Mask of the fields that have been touched since this trace has last been emitted. Note that touched does not necessarily mean the value changed. The index of the field is the bit index in the mask plus the index of this field plus one.

UAO unsigned int

User Access Override bit.

UINJ unsigned int

Undefined Instruction exceptions bit.

ZA unsigned int

ZA storage and ZT0 register enable bit.

nRW unsigned int

Current Execution state bit.

PSTATE_PACTIVE_OPMODE_TRANSITION

Trace pactive and Opmode mode transitions on PChannel. Fields:

Copyright © 2023–2025 Arm Limited (or its affiliates). All rights reserved.

Non-Confidential

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Operating mode after the transition.

OLD_STATE enum

Operating mode before the transition.

PSTATE_PACTIVE_OPMODE_TRANSITION_DENIED

Trace the denied pactive and Opmode mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Operating mode after the transition.

OLD_STATE enum

Operating mode before the transition.

REASON string

Reason for the transition being denied.

PSTATE_PACTIVE_POWER_TRANSITION

Trace pactive and power mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Power mode after the transition.

OLD_STATE enum

Power mode before the transition.

PSTATE_PACTIVE_POWER_TRANSITION_DENIED

Trace the denied pactive and power mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Power mode after the transition.

OLD_STATE enum

Power mode before the transition.

REASON string

Reason for the transition being denied.

PSTATE_PACTIVE_THREAD_OPMODE_TRANSITION

Trace pactive and Opmode mode transitions for threads on PChannel. Fields:

DEVPACTIVE **unsigned int**

Integer representation of DEVPACTIVE[] signals.

NEW_STATE **enum**

Operating mode after the transition.

OLD_STATE **enum**

Operating mode before the transition.

QUANTUM_START

Called at the start of every quantum.

RUN_STATE

Run state transition. Fields:

INST_COUNT **unsigned int**

Ticks count at point of transition.

NEW **enum**

New run state.

OLD **enum**

Old run state.

SEMIHOSTING_CALL

Call of a semihost function occurred. Fields:

PC **unsigned int**

The program counter after the semihosting call.

SEMIHOSTING_PRECALL

About to call semihost. Fields:

EL2 **bool**

Use EL2 translation regime.

NS **bool**

Is Non-Secure.

REG_WIDTH **unsigned int**

The current register width in bytes.

SIGNAL

External signal state change. Fields:

SIGNAL **enum**

Signal that changed.

STATE **bool**

Signal asserted state.

SPSR

SPSR change. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

MODE enum

Which of the banked SPSR registers is written.

OLD_VALUE unsigned int

The old SPSR value.

VALUE unsigned int

The new SPSR value.

START_COMPILE

Compilation started. Fields:

VADDR unsigned int

Instruction where compilation begins.

STG_MULTIPLE_MICRO_OPS_ADDRESS_RANGE

Address range for MTE stores with multiple uops. Fields:

ADDR_END unsigned int

Final address.

ADDR_START unsigned int

Starting address.

STORE_MULTIPLE_REGS_ADDRESS_RANGE

Address range for stores with multiple registers. Fields:

ADDR_END unsigned int

Final address.

ADDR_START unsigned int

Starting address.

SUBOPTIMAL_LDST_RETIRED

suboptimal load/store. Fields:

ACCESS_SIZE unsigned int

Log2 of the access width in bytes used for alignment checking (1: half-word, 2: word ...).

OPTIMAL_ALIGN_SELECTOR unsigned int

Selector to decide the optimal alignment size (1: access width, 2: 4-bytes, 3: 8-bytes, ..., 12: 4-kbytes).

VADDR unsigned int

The virtual address of the access.

SYNC

Called for every synchronization, that is every quantum break. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The instruction count of this CPU.

LOCAL_QUANTUM unsigned int

The local quantum of this CPU.

LOCAL_TIME unsigned int

The local time of this CPU.

SYNC_REASON enum

Reason for sync.

SYSCALL

System call instruction executed. Fields:

IMM unsigned int

Immediate value of the system call instruction.

TYPE enum

System call type.

VADDR unsigned int

Instruction that caused the system call.

SYSREG_READ128

System Coprocessor 128 bit register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

REGNUM enum

Internal register number.

REGNUM_PHYS enum

Internal register number physically accessed (e.g. in presense of VHE redirection).

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_READ64

System Coprocessor register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REGNUM enum

Internal register number.

REGNUM_OPERAND enum

Register number corresponding to the instruction operand.

REGNUM_PHYS enum

Internal register number physically accessed (e.g. in presense of VHE redirection).

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_UPDATE128

Triggers when the system updates a register. Fields:

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG enum

Register number.

UNKNOWN unsigned int

Bits of the register which became unknown.

VALUE unsigned int

Value written to the register.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_UPDATE32

Triggers when the system updates a register. Fields:

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG enum

Register number.

UNKNOWN unsigned int

Bits of the register which became unknown.

VALUE unsigned int

Value written to the register.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_UPDATE64

Triggers when the system updates a register. Fields:

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG enum

Register number.

UNKNOWN unsigned int

Bits of the register which became unknown.

VALUE unsigned int

Value written to the register.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_WRITE128

System Coprocessor 128 bit register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

REGNUM enum

Internal register number.

REGNUM_PHYS enum

Internal register number physically accessed (e.g. in presense of VHE redirection).

UNDEF bool

The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_WRITE64

System Coprocessor register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REGNUM enum

Internal register number.

REGNUM_OPERAND enum

Register number corresponding to the instruction operand.

REGNUM_PHYS enum

Internal register number physically accessed (e.g. in presense of VHE redirection).

UNDEF bool

The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

UNALIGNED_LDST_RETIRED

Processor unaligned load/store. Fields:

ACCESS_SIZE unsigned int

Log2 of the access width in bytes used for alignment checking (1=>half-word, 2->word ...).

NUMBER_OF_BEATS unsigned int

Number of accesses (beats) in this burst (Total size in bytes is ACCESS_SIZE * NUMBER_OF_BEATS).

VADDR unsigned int

The virtual address of the access.

VFP_D_REGS

VFP/NEON D 64 bit register write. Fields:

ALIASING enum

Type of register aliasing used.

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a write to the corresponding bit in the register. For example, 0x1 would indicate a write to VALUE[0].

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

VFP_D_REGS_READ

VFP/NEON D 64 bit register read. Fields:

ID unsigned int

The register number.

VALUE unsigned int

The current value of the register.

VFP_Q_REGS

VFP/NEON Q 128 bit register write. Fields:

ALIASING enum

Type of register aliasing used.

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a write to the corresponding byte in the register. For example, 0x1 would indicate a write to VALUE[7:0].

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

VFP_Q_REGS_READ

VFP/NEON Q 128 bit register read. Fields:

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a read to the corresponding byte in the register. For example, 0x1 would indicate a read to VALUE[7:0].

VALUE unsigned int

The current value of the register.

VFP_SYS_REGS

Writes to the VFP/NEON units system registers. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ID enum

Which VFP system register is written.

MASK unsigned int

The mask indicating updated cumulative bits.

OLD_VALUE unsigned int

The register's old value overwritten.

VALUE unsigned int

The new value written to the VFP system register.

VFP_S_REGS

VFP/NEON S 32 bit register write. Fields:

ALIASING enum

Type of register aliasing used.

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a write to the corresponding bit in the register. For example, 0x1 would indicate a write to VALUE[0].

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

VFP_S_REGS_READ

VFP/NEON S 32 bit register read. Fields:

ID unsigned int

The register number.

VALUE unsigned int

The current value of the register.

WAYPOINT

Signals end of basic block execution. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ISET enum

Origin instruction set.

IS_COND bool

Indicates if this is a conditional waypoint.

PC unsigned int

Origin address (or 0 if unavailable).

TAKEN bool

Indicates if this waypoint was taken.

TARGET unsigned int

Destination address.

TARGET_ISET enum

Destination instruction set.

WFE_END

WFE ended. Fields:

INST_COUNT unsigned int

Ticks count when leaving WFE.

WFE_EVENT_REGISTER

WFE event register status: set/clear, reason. Fields:

INST_COUNT unsigned int

Ticks count.

REASON enum

Reason for set/clear. Only REASON==0 (Cleared by WFE) clears the bit, all other reasons set it.

WFE_IGNORED

WFE ignored. Fields:

EVENT bool

This WFE was ignored because the event register was set.

INST_COUNT unsigned int

Ticks count when ignoring WFE.

TRAPPED bool

This WFE was trapped.

WFE_START

WFE entered. Fields:

INST_COUNT unsigned int

Ticks count when entering WFE.

WFI_END

WFI ended. Fields:

INST_COUNT unsigned int

Ticks count when leaving WFI.

WFI_IGNORED

WFI ignored. Fields:

DISABLED bool

This WFI was ignored because WFI is disabled.

INST_COUNT unsigned int

Ticks count when ignoring WFI.

REASON enum

specifies reason why WFI trace was ignored.

TRAPPED bool

This WFI was trapped.

WFI_START

WFI entered. Fields:

INST_COUNT unsigned int

Ticks count when entering WFI.

WFI_WAKEUP

WFI wakeup. Fields:

INST_COUNT **unsigned int**

Ticks count when WFI wakeup occurred.

REASON **enum**

Reason for wakeup.

2.54 ARM_Cortex-R82

This section describes the trace sources.

AA64_ASE_SVE_REGS

Modification of AArch64 Advanced SIMD or SVE Vector Registers (via SIMD or Scalar operations), SVE Predicate Registers and ZTO. Fields:

ID **enum**

The register identifier.

MASK **unsigned int**

Bitmask of the register to signal the modified bits in the VALUE field.

SM **bool**

Whether the PE is in Streaming Mode.

VALUE **unsigned int**

Value of AArch64 vector register.

ASYNC_MEMORY_FAULT

Context ID Register write. Fields:

FAULT **unsigned int**

Fault status present in ESR.ISS encoding for Serror in ARCH64 and present in DFSR encoding for external abort format in ARCH32.

PADDR **unsigned int**

Physical Address (or 0 if unavailable).

ATOMIC_END_ACCESS

Bus trace access for atomics. Fields:

ACCESS_FAIL **bool**

Memory access failed.

ADDR **unsigned int**

The virtual address of the access.

ATTR **unsigned int**

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory

attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

COMPARE_VALUE unsigned int

Compare value for CAS.

LOAD_VALUE unsigned int

Loaded values.

NSDESC unsigned int

The security state of the access.

OPERAND_VALUE unsigned int

Operation's operand.

OPERATION enum

Operation type.

PADDR unsigned int

The physical address of the access.

PRIV bool

Is this a privileged access?.

ATOMIC_START_ACCESS

Bus trace access for atomics. Fields:

ADDR unsigned int

The virtual address of the access.

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

COMPARE_VALUE unsigned int

Compare value for CAS.

NSDESC unsigned int

The security state of the access.

OPERAND_VALUE unsigned int

Operation's operand.

OPERATION enum

Operation type.

PADDR unsigned int

The physical address of the access.

PRIV bool

Is this a privileged access?.

ArchMsg.Error.exit_code

DISPLAY exit code: %{EXITCODE} component kind: %{KIND} reason: %{REASON}. Fields:

EXITCODE signed int

Exit code to be returned.

KIND string

Component kind that invoked the exit code trace.

REASON string

Reason why the exit code trace was invoked (optional).

ArchMsg.Info.invalid_region_programmed

DISPLAY MPU Stage %{STAGE:d}; Region ID %{REGION_ID:d} has base address greater than end address Base Addr=%{BASE_ADDR:x}, End Addr=%{END_ADDR:x}. Fields:

BASE_ADDR unsigned int

Base Address.

END_ADDR unsigned int

End Address.

REGION_ID unsigned int

Region ID.

STAGE unsigned int

MPU Stage.

ArchMsg.Warning.IgnoredWarmResetRequest

Event fires when software attempts to request warm reset through write to deprecated DBGPRCR.CWRR.

ArchMsg.Warning.RAS_ErrselrUnpredictableState

DISPLAY ERRSELR is in unpredictable state due to value greater than number_of_error_records written to it.

ArchMsg.Warning.RAS_UnpredictableStateWrite

DISPLAY %{REG_NAME} write ignored due to unpredictable ERRSELR state. Fields:

REG_NAME string

Name of register being written.

ArchMsg.Warning.ReportedRevisionRegisterRead

DISPLAY Reported revision and patch do not match that of the cpu. Fields:

ORIGINAL_VALUE unsigned int

The value the register would have based on the modelled revision.

REGISTER string

The name of the register that has ben read.

REPORTED_VALUE unsigned int

The value the register has been overridden to.

ArchMsg.Warning.Unpredictably Indexed PM Event Register

TAGS **UNPREDICTABLE** DISPLAY Event %{IsDirect:(Selector value|Register)} %{SEL} is >= %{N}.

Fields:

IsDirect unsigned int

Direct Access rather than indirect.

N unsigned int

Number of Accessible Registers.

SEL unsigned int

Selector.

ArchMsg.Warning.cache_contents_unknown

A cache has been activated without being invalidated and may contain **UNKNOWN** content. DISPLAY %{SIDE:(D|I)}-Cache was enabled in %{REGIME:(SECURE|NON_SECURE|HYP)} regime but was not invalidated since power-on: cache lines could contain **UNKNOWN** content %{SIDE:(Note: D-side caches must invalidate SECURE lines even if the cache is only used by NON_SECURE code in order to ensure that unexpected dirty lines tagged SECURE are not naturally evicted)}. Fields:

REGIME unsigned int

Current translation regime.

SIDE unsigned int

0 d-side cache or unified cache, 1 i-side.

ArchMsg.Warning.dcimvac_matches_watchpoint

TAGS **UNPREDICTABLE** DISPLAY Watchpoints matching an AArch32 DCIMVA are implementation defined.

ArchMsg.Warning.decode_fieldmismatch

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with different fields %{FIELD1} and %{FIELD2}. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_frachitsnegative

TAGS **UNPREDICTABLE** DISPLAY VCVT Instruction is unpredictable with a negative number of fraction bits.

ArchMsg.Warning.decode_initblock

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable in an IT block.

ArchMsg.Warning.decode_initblocknotlast

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable in an IT block when not the last.

ArchMsg.Warning.decode_invalid_condition

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable when the condition is not AL.

ArchMsg.Warning.decode_invalidfieldcombination

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with the values in %{FIELD1} and %{FIELD2}. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_invalidvalue

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with the value in %{FIELD}. Fields:

FIELD enum

Field name.

ArchMsg.Warning.decode_mem_hint_unallocated

TAGS **UNPREDICTABLE** DISPLAY There is no memory hint allocated to this bit pattern.

ArchMsg.Warning.decode_registermatch

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with %{FIELD1} and %{FIELD2} as the same register. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_registermismatch

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with %{FIELD1} and %{FIELD2} as different registers. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_registeroutofrange

TAGS **UNPREDICTABLE** DISPLAY Floating-point instruction is unpredictable indexing beyond the end of the register bank.

ArchMsg.Warning.decode_sbzsbo

TAGS **UNPREDICTABLE** DISPLAY Reserved bits in the instruction are not canonical.

ArchMsg.Warning.decode_transactiontoobig

TAGS UNPREDICTABLE DISPLAY Attempt to perform a memory transaction of more than 128 bytes.

ArchMsg.Warning.decode_unpred_other

TAGS UNPREDICTABLE DISPLAY Instruction is unpredictable.

ArchMsg.Warning.decode_unpreduseofpc

TAGS UNPREDICTABLE DISPLAY Instruction is unpredictable when %{FIELD} is PC. Fields:

FIELD enum

Field name.

ArchMsg.Warning.decode_unpreduseofr13

TAGS UNPREDICTABLE DISPLAY Instruction is unpredictable when %{FIELD} is R13. Fields:

FIELD enum

Field name.

ArchMsg.Warning.decode_unpreduseofr13inreglist

TAGS UNPREDICTABLE DISPLAY Instruction is unpredictable with R13 in register-list.

ArchMsg.Warning.decode_useofd16orhigher

TAGS UNPREDICTABLE DISPLAY Attempt to use register D16 or higher where it is not allowed.

ArchMsg.Warning.decode_writebackandbaseinlist

TAGS UNPREDICTABLE DISPLAY Instruction is unpredictable with write back when the base register is in the transfer list.

ArchMsg.Warning.decode_zeroregistersinlist

TAGS UNPREDICTABLE DISPLAY Instruction is unpredictable when no registers are to be transferred.

ArchMsg.Warning.recursive_branch

An instruction is performing a branch that targets the same instruction. If this was intended then a WFI might be more effective DISPLAY recursive branch detected at PC=%{PC}. PRIMARY KEY PC. Fields:

PC unsigned int

Address of instruction causing the branch.

ArchMsg.Warning.recursive_exception

An instruction has generated an exception that targets the same instruction. DISPLAY recursive %{TYPE} exception detected at PC=%{PC} (CPSR=%{CPSR}). PRIMARY KEY PC. Fields:

CPSR unsigned int

Processor state.

PC unsigned int

Address of instruction causing the exception.

TYPE enum

Type of exception.

ArchMsg.Warning.reentrant_vector_catch

TAGS **UNPREDICTABLE** DISPLAY Vector catch debug event on Prefetch or Data abort vector %{ADDR}. Fields:

ADDR unsigned int

vector address.

ArchMsg.Warning.reserved_it_state

TAGS **UNPREDICTABLE** DISPLAY Execution with reserved IT state %{ITSTATE}. Fields:

ITSTATE unsigned int

reserved IT state.

ArchMsg.Warning.tlb_contents_unknown

Translation has been enabled without invalidating the TLB. DISPLAY Translation was enabled in the %{PAS:(Secure|Non-secure|Root|Realm)} EL%{EL:d} translation regime (vmid=%{VMID:x}) without invalidating %{INVALIDITY:(|d-side |i-side |)}entries since power-on: %{INVALIDITY:(|D|I|)}TLB could contain **UNKNOWN** content. Fields:

EL unsigned int

Exception level of translation regime.

INVALIDITY unsigned int

Which TLB may be invalid (if TLBs are separate).

NON_SECURE bool

Whether the translation regime is non-secure.

PAS unsigned int

Physical address space of translation regime.

VMID unsigned int

Current VMID, if applicable.

ArchMsg.Warning.unknown_DLR_DSPSR

Exiting debug state with unknown DLR or DSPSR.

ArchMsg.Warning.unknown_ELR_SPSR

Returning from debug exception with unknown ELR or SPSR.

ArchMsg.Warning.unpred_FAR_on_LRCPC3_non_atomic_mem_access

TAGS **UNPREDICTABLE** Unpredictable FAR/FSTATUS on LRCPC3 pair instructions resulting in two separate memory accessesDISPLAY Unpredictable FAR on LRCPC3 non atomic memory access at %{ADDR1} with FSTATUS %{FSTATUS1} or %{ADDR2} with FSTATUS %{FSTATUS2}. Fields:

ADDR1 unsigned int

Possible FAR address 1.

ADDR2 unsigned int

Possible FAR address 2.

FSTATUS1 unsigned int

Possible fault status value 1.

FSTATUS2 unsigned int

Possible fault status value 2.

ArchMsg.Warning.unpred_async_tag_check_fail_on_sve_speculative_load_do_update_ffr

Unpredictable set to FALSE of the FFR element associated to SVE speculative load with an async tag check fail TAGS **UNPREDICTABLE**.

ArchMsg.Warning.unpredictable_a32_breakpoint

TAGS **UNPREDICTABLE** DISPLAY Unpredictable breakpoint on a A32 instruction at %{ADDR1} Offset %{OFFSET} BAS %{BAS} for a %{IS_ADDRESS_MISMATCH_BREAKPOINT:(Mismatch|Match)} Breakpoint number %{N:d} Breakpoint address %{ADDR2}. Fields:

ADDR1 unsigned int

Address.

ADDR2 unsigned int

Breakpoint address.

BAS unsigned int

BAS field.

IS_ADDRESS_MISMATCH_BREAKPOINT bool

Is a Mismatch breakpoint, not a Match breakpoint.

N unsigned int

Breakpoint number.

OFFSET unsigned int

Always has a value of zero.

ArchMsg.Warning.unpredictable_banked_register_access

TAGS **UNPREDICTABLE** DISPLAY Access to banked register (R=%{R:d} SYSm=%{SYSM:2x}) is unpredictable. Fields:

R unsigned int

R bit.

SYSM unsigned int

SYSm.

ArchMsg.Warning.unpredictable_ccfail_undef

TAGS **UNPREDICTABLE** DISPLAY An instruction %{OPCODE} at VA=%{VADDR} failed its condition codes check but would otherwise have caused a non-data-dependent trap or undefined exception. Behaviour is implementation-defined choice of exception or nop. Fields:

OPCODE unsigned int

opcode of the instruction.

VADDR unsigned int

Virtual address of the instruction being executed.

ArchMsg.Warning.unpredictable_ici_bits

TAGS **UNPREDICTABLE** DISPLAY ICI bits do not apply or are architecturally unpredictable for instruction.

ArchMsg.Warning.unpredictable_t32_breakpoint

TAGS **UNPREDICTABLE** DISPLAY Unpredictable breakpoint on a T32 instruction at %{ADDR1} Offset %{OFFSET} BAS %{BAS} for a %{IS_ADDRESS_MISMATCH_BREAKPOINT:(Mismatch|Match)} Breakpoint number %{N:d} Breakpoint address %{ADDR2}. Fields:

ADDR1 unsigned int

Address.

ADDR2 unsigned int

Breakpoint address.

BAS unsigned int

BAS field.

IS_ADDRESS_MISMATCH_BREAKPOINT bool

Is a Mismatch breakpoint, not a Match breakpoint.

N unsigned int

Breakpoint number.

OFFSET unsigned int

Indicates which halfword of a 32-bit Thumb instruction the breakpoint is set on. 0 for the first halfword or if the instruction is 16 bits wide, or 1 for the second halfword.

ArchMsg.Warning.unpredictable_t32_vector_catch

TAGS **UNPREDICTABLE** DISPLAY Unpredictable Vector catch exception on a T32 instruction at %{ADDR} Offset %{OFFSET}. Fields:

ADDR unsigned int

Address.

OFFSET unsigned int

Indicates which halfword of a 32-bit Thumb instruction the vector catch exception is set on. 0 for the first halfword or if the instruction is 16 bits wide, or 1 for the second halfword.

ArchMsg.Warning.unpredictable_watchpoint_far

TAGS **UNPREDICTABLE** DISPLAY The value for FAR/EDWAR for hitting this watchpoint may be anywhere in the range %{LOWER_BOUND} to %{UPPER_BOUND}. Fields:

LOWER_BOUND unsigned int

the lowest address accessed by the instruction that triggered the watchpoint.

UPPER_BOUND unsigned int

the highest watchpointed address accessed by that instruction.

ArchMsg.Warning.unpredictable_watchpoint_far_isscmp

DISPLAY The value for FAR/EDWAR for hitting this watchpoint may be anywhere in the range %{LOWER_BOUND} to %{UPPER_BOUND}. Fields:

LOWER_BOUND unsigned int

the lowest address accessed by the instruction that triggered the watchpoint.

UPPER_BOUND unsigned int

the highest watchpointed address accessed by that instruction.

ArchMsg.Warning.warning_AdvSIMDExpandImmUnexpectedZero

TAGS **UNPREDICTABLE** DISPLAY AdvSIMDExpandImm may treat this immediate value as **UNPREDICTABLE**.

ArchMsg.Warning.warning_ConditionalSMC

TAGS **UNPREDICTABLE** DISPLAY SMC instruction has **UNPREDICTABLE** effects when conditional and when combined with traps.

ArchMsg.Warning.warning_TLBID_out_of_range

TLBID out of range in TLBI according to the configuration of NIS, NVIS, NOS and NVOS. Fields:

NIS unsigned int

NIS.

NOS unsigned int

NOS.

NVIS unsigned int

NVIS.

NVOS unsigned int

NVOS.

TLBID unsigned int

The TLBID value used in the TLBI instruction.

ArchMsg.Warning.warning_TTL_or_TG_mismatches_current_translation_granule

TTL or TG mismatches current translation granule in TLBI VA/RVA/IPA/RIPA. Fields:

CURRENT_TG unsigned int

Current TG as indicated by TCR_ELx or VTCR_EL2.

TLBI_TG unsigned int

TG (or TTL[3:2]) used in TLBI instruction.

ArchMsg.Warning.warning_access_crosses_page_boundary

TAGS **UNPREDICTABLE** DISPLAY Access crosses page boundary.

ArchMsg.Warning.warning_access_crosses_page_boundary_has_fault

TAGS **UNPREDICTABLE** DISPLAY Access that crosses page boundary has fault(s).

ArchMsg.Warning.warning_access_crosses_page_boundary_has_fault_on_both_pages

TAGS **UNPREDICTABLE** DISPLAY Access that crosses page boundary has a fault on each page.

Fields:

ADDR1 unsigned int

FAR address on first page.

ADDR2 unsigned int

FAR address on second page.

FSTATUS1 unsigned int

Fault status on first page.

FSTATUS2 unsigned int

Fault status on second page.

ArchMsg.Warning.warning_access_crosses_page_boundary_spanning_different_memory

TAGS **UNPREDICTABLE** DISPLAY Access at address %{ADDR} crosses page from MEM_TYPE %{MEMTYPE_PAGE1} to %{MEMTYPE_PAGE2}. Fields:

ADDR unsigned int

address of access crossing page.

MEMTYPE_PAGE1 signed int

First Page's memory type (0-Normal, 1-Non-Normal).

MEMTYPE_PAGE2 signed int

Second Page's memory type (0-Normal, 1-Non-Normal).

ArchMsg.Warning.warning_access_wraps_around_memory

TAGS **UNPREDICTABLE** DISPLAY Access wraps around memory in %{WIDTH:d}bit mode.

Fields:

WIDTH unsigned int

Address width.

ArchMsg.Warning.warning_bcr_linking_status

TAGS **UNPREDICTABLE** DISPLAY DBGBCR.{BT2,BT} combination gives unpredictable behaviour.

ArchMsg.Warning.warning_bcr_mask_reserved

TAGS **UNPREDICTABLE** DISPLAY DBGBCR_ELn.MASK is set to a reserved value.

ArchMsg.Warning.warning_bvr_masked_bit_not_zero

TAGS **UNPREDICTABLE** DISPLAY DBGBVRn_EL1.VA contains a bit masked by DBGBCRn_EL1.MASK that is not zero.

ArchMsg.Warning.warning_ccsidr_unimplemented_level

TAGS **UNPREDICTABLE** DISPLAY CCSIDR read while CSSELR %{CSSELR:x} points to an unimplemented cache level. Fields:

CSSELR unsigned int

current effective value of CSSELR.

ArchMsg.Warning.warning_change_to_ns_when_tge_set

TAGS **UNPREDICTABLE** DISPLAY Attempting to change NS to 1 when HCR.TGE = 1.

ArchMsg.Warning.warning_contiguous_bit_check_abort

TAGS **UNPREDICTABLE** DISPLAY An abort occurred whilst attempting to check TLB entry is contiguous at %{ENTRY_ADDR}. Fields:

ENTRY_ADDR unsigned int

address of conflicting TLB entry.

ArchMsg.Warning.warning_contiguous_bit_error

TAGS **UNPREDICTABLE** DISPLAY TLB entry at %{ENTRY_ADDR} was not contiguous with entry at %{CONTIG_ADDR} Contents are %{ENTRY_DATA} but expected %{CONTIG_DATA}. Fields:

CONTIG_ADDR unsigned int

address of the TLB first read, with which this entry is expected to be contiguous.

CONTIG_DATA unsigned int

expected contents based on the entry at CONTIG_ADDR.

ENTRY_ADDR unsigned int

address of conflicting TLB entry.

ENTRY_DATA unsigned int

contents of conflicting TLB entry.

ArchMsg.Warning.warning_contiguous_bit_error_D128

TAGS **UNPREDICTABLE** DISPLAY TLB entry at %{ENTRY_ADDR} was not contiguous with entry at %{CONTIG_ADDR} Contents are %{ENTRY_DATA_HI} %{ENTRY_DATA_LO} but expected %{CONTIG_DATA}. Fields:

CONTIG_ADDR unsigned int

address of the TLB first read, with which this entry is expected to be contiguous.

CONTIG_DATA unsigned int

expected contents based on the entry at CONTIG_ADDR.

ENTRY_ADDR unsigned int

address of conflicting TLB entry.

ENTRY_DATA_HI unsigned int

Higher bit contents of conflicting TLB entry.

ENTRY_DATA_LO unsigned int

Lower bit contents of conflicting TLB entry.

ArchMsg.Warning.warning_cp10_cp11_mismatch

TAGS **UNPREDICTABLE** DISPLAY Unpredictable configuration of CP10 and CP11 controls at EL %{EL:d}. Fields:

EL unsigned int

EL owning the control.

ArchMsg.Warning.warning_cp10_cp11_reserved_value

TAGS **UNPREDICTABLE** DISPLAY Unpredictable configuration of CPACR CP10 ({CP10}) or CP11 ({CP11}). Fields:

CP10 unsigned int

CP10 access.

CP11 unsigned int

CP11 access.

ArchMsg.Warning.warning_csselr_level_out_of_range

TAGS **UNPREDICTABLE** DISPLAY CSSELR.Level out of range, written value {LEVEL} with only {IMPLEMENTED:d} levels implemented. Fields:

IMPLEMENTED unsigned int

number of cache levels implemented.

LEVEL unsigned int

written value of CSSELR.Level.

TYPE enum

The type of cache selected by CSSELR.InD and TnD.

ArchMsg.Warning.warning_debug_flow_control_bits_not_obeyed

TAGS **UNPREDICTABLE** DISPLAY Flow control bits for {REGISTER} were not obeyed, causing {ERROR}. Fields:

ERROR enum

type of error resulting.

REGISTER enum

register under consideration.

ArchMsg.Warning.warning_debug_register_access_during_reset

TAGS **UNPREDICTABLE** DISPLAY Unpredictable {IS_WRITE:(read|write)} access to debug register offset {OFFSET} during reset. Fields:

IS_WRITE unsigned int

Write Not Read.

OFFSET unsigned int

Register Offset.

ArchMsg.Warning.warning_decode_cps_inconsistent_fields

TAGS **UNPREDICTABLE** DISPLAY The mode setting fields in the CPS instruction are inconsistent.

ArchMsg.Warning.warning_decode_invalid_state

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable in current exception-level/security state.

ArchMsg.Warning.warning_default_cacheable_mmu_on

TAGS **UNPREDICTABLE** DISPLAY Default cacheable is enabled (HCR.DC = 1) while MMU is enabled.

ArchMsg.Warning.warning_default_cacheable_vmmu_off

TAGS **UNPREDICTABLE** DISPLAY Default cacheable is enabled (HCR.DC = 1) while HCR.VM = 0.

ArchMsg.Warning.warning_deprecated_wvr_bit_2_set

TAGS **UNPREDICTABLE** DISPLAY DBGWVR%{N:d}_EL1=%{WVR:h} has bit 2 set, deprecated in ARMv8. Fields:

N unsigned int

index.

WVR unsigned int

DBGWVR.

ArchMsg.Warning.warning_eret_while_software_step_active_pending

Missing ISB between setting MDSCR_EL1.SS and ERET TAGS **UNPREDICTABLE**.

ArchMsg.Warning.warning_exclusive_to_non_normal

TAGS **UNPREDICTABLE** DISPLAY Exclusive Access to Non-Normal Memory %{ADDR}. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_exclusive_to_non_writeback

TAGS **UNPREDICTABLE** DISPLAY Exclusive Access to Non-Writeback-Cacheable Memory %{ADDR}. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_execute_from_device_memory

TAGS **UNPREDICTABLE** DISPLAY An attempt was made to execute from Device memory.

ArchMsg.Warning.warning_illegal_cpsr_mode

TAGS **UNPREDICTABLE** DISPLAY Writing an illegal or unimplement mode (%{NEW_MODE:h}) to CPSR was unpredictable. Fields:

NEW_MODE unsigned int

New value of CPSR.M.

ArchMsg.Warning.warning_illegal_srs_mode

TAGS **UNPREDICTABLE** DISPLAY Illegal or **UNPREDICTABLE** mode (%{MODE:h}) used for SRS instruction. Fields:

MODE unsigned int

mode for Banked SP.

ArchMsg.Warning.warning_implementation_defined_read_debug_register_in_SCS

Read Access to Debug registers from the processor is IMPLEMENTATION DEFINED TAGS IMP_DEF. Fields:

PPB_OFFSET unsigned int
debug register offset.

ArchMsg.Warning.warning_implementation_defined_sequential_security_transitions_supported

The behavior of sequential instruction fetches that cross from non-secure to secure memory and contain an SG instruction is CONSTRAINED UNPREDICTABLE TAGS UNPREDICTABLE.

ArchMsg.Warning.warning_implementation_defined_stack_limit_check_supported

It is IMPLEMENTATION DEFINED whether stack pointer limit checking is performed for this instruction TAGS IMP_DEF. Fields:

ADDRESS unsigned int
Address of instruction.

OPCODE unsigned int
instruction opcode.

ArchMsg.Warning.warning_implementation_defined_taggednonsharedmemorytype

TAGS IMP_DEF DISPLAY Memory attribute has Tagged, non shared memory type; it is IMPLEMENTATION DEFINED whether the memory location is treated as Tagged or non-Tagged.

ArchMsg.Warning.warning_implementation_defined_write_debug_register_in_SCS

Write Access to Debug registers from the processor is IMPLEMENTATION DEFINED TAGS IMP_DEF. Fields:

DATA unsigned int
data attempted to be written.

PPB_OFFSET unsigned int
debug register offset.

ArchMsg.Warning.warning_invalid_tcr_granule

TAGS UNPREDICTABLE DISPLAY TCR.TG%{TG_ID:(0|1)} bits have been set to a granule size %{{REQUEST:d}}K not implemented - using %{{SUBSTITUTE:d}}K. Fields:

REQUEST signed int
page size requested (or 0 for reserved).

SUBSTITUTE signed int
best guess available page size.

TG_ID bool
bits TG0 or TG1.

ArchMsg.Warning.warning_load_multiple_user_registers_from_user_mode

TAGS UNPREDICTABLE DISPLAY An LDM(user registers) instruction executed from user mode.

ArchMsg.Warning.warning_mte_in_mem_attr_not_supported

TAGS UNPREDICTABLE DISPLAY Memory attribute %{ATTR} has Tagged memory type but Armv8.5-MTE is not implemented. Fields:

ATTR unsigned int

Memory attribute.

ArchMsg.Warning.warning_reserved_breakpoint_state_match

TAGS UNPREDICTABLE DISPLAY %{IS_BREAKPOINT:(Watchpoint|Breakpoint)} programmed with a reserved combination of HMC, SSC and %{IS_BREAKPOINT:(PAC|PMC)}. Fields:

IS_BREAKPOINT unsigned int

Is a breakpoint, not a watchpoint.

ArchMsg.Warning.warning_shareability

TAGS UNPREDICTABLE DISPLAY Unpredictable: combination of the 1st and 2nd stages of translation is Normal Inner Non-Cacheable, Outer Non-Cacheable.

ArchMsg.Warning.warning_software_step_set_while_enabled

TAGS UNPREDICTABLE DISPLAY MDSCR_EL1.SS set to 1 while software step debug exceptions are enabled.

ArchMsg.Warning.warning_thumb_instruction_wraps_around_memory

TAGS UNPREDICTABLE DISPLAY Thumb instruction wraps around memory.

ArchMsg.Warning.warning_ttbr_sbz_bits_are_not_zero

TAGS UNPREDICTABLE DISPLAY %{TTBR}[%{MSB:d};0] should be zero, but are not. Fields:

MSB unsigned int

x-1.

TTBR enum

which TTBR register.

ArchMsg.Warning.warning_unaligned_address_dbgdtrrx_write

TAGS UNPREDICTABLE DISPLAY Unaligned address %{ADDR} in X0/R0 when DBGDTRRX written in memory access mode. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unaligned_address_dbgdtrtx_read

TAGS UNPREDICTABLE DISPLAY Unaligned address %{ADDR} in X0/R0 when DBGDTRTX read in memory access mode. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unaligned_to_device

TAGS UNPREDICTABLE DISPLAY Unaligned Access to Device Memory %{ADDR}. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unaligned_to_strongly_ordered

TAGS **UNPREDICTABLE** DISPLAY Unaligned Access to Strongly Ordered Memory %{ADDR}.

Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unknown_sau_rnr

TAGS **UNPREDICTABLE** DISPLAY SAU_RNR was set to an unsupported value.

ArchMsg.Warning.warning_unpredictable_AIRCR_PRIS_and_BFHFNMINs

TAGS **UNPREDICTABLE** DISPLAY The effect of setting both AIRCR.BFHFNMINs and AIRCR.PRIS to 1 is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_AIRCR_VECTCLRACTIVE_when_not_in_debug

TAGS **UNPREDICTABLE** DISPLAY The effect of writing a 1 to AIRCR.VECTCLRACTIVE if the processor is not halted in Debug state is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_AIRCR_VECTRESET_and_SYSRESETREQ

TAGS **UNPREDICTABLE** DISPLAY When the processor is halted in Debug state, if a write to the register writes a 1 to both VECTRESET and SYSRESETREQ, the behavior is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_AIRCR_VECTRESET_when_not_in_debug

TAGS **UNPREDICTABLE** DISPLAY The effect of writing a 1 to AIRCR.VECTRESET if the processor is not halted in Debug state is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_AIRCR_incorrect_VKEY

TAGS **UNPREDICTABLE** DISPLAY The value 0x05FA must be written to AIRCR.VKEY, otherwise the register write is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_EXC_RETURN_Reserved_Bit

EXC_RETURN[23:7] are reserved with the special condition that all bits should be written as one. Values other than all 1s are **UNPREDICTABLE**. TAGS **UNPREDICTABLE**. Fields:

EXC_RETURN unsigned int

Special PC value.

SBO_MASK unsigned int

Which bits Should Be One to not UNPRED.

ArchMsg.Warning.warning_unpredictable_PMEVTYPEPERn_ELO_VS_reserved_value

TAGS **UNPREDICTABLE** DISPLAY Unpredictable reserved value of PMEVTYPER<n>_ELO.VS.

ArchMsg.Warning.warning_unpredictable_change_to_DEMCR_MON_STEP_at_insufficient_priority

TAGS **UNPREDICTABLE** DISPLAY The effect of changing DEMCR.MON_STEP at an execution priority that is lower than the priority of the DebugMonitor exception is **CONSTRAINED UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_change_to_priority_of_active_exception

TAGS UNPREDICTABLE DISPLAY Changing the priority of an active exception is UNPREDICTABLE.

ArchMsg.Warning.warning_unpredictable_clear_lspact

TAGS UNPREDICTABLE DISPLAY UNPREDICTABLE state on context switch. Fields:

DRVEXC_TAKEN unsigned int

Derived exception taken, not pended.

REACHED_LAST_STORE unsigned int

Only the last faulted.

ArchMsg.Warning.warning_unpredictable_dbg_exit_unaligned_dlr

TAGS UNPREDICTABLE DISPLAY Debug exit to AArch32 with DLR[0] set to 1. PC[0] can be set to 0 or to DLR[0].

ArchMsg.Warning.warning_unpredictable_exception_catch

TAGS UNPREDICTABLE DISPLAY Generation of an exception catch event is unpredictable.

ArchMsg.Warning.warning_unpredictable_exception_return_inconsistent_state

TAGS UNPREDICTABLE DISPLAY state on exception return is unpredictable.

ArchMsg.Warning.warning_unpredictable_exception_return_instruction

TAGS UNPREDICTABLE DISPLAY exception return instruction is unpredictable.

ArchMsg.Warning.warning_unpredictable_in_debug_state

TAGS UNPREDICTABLE DISPLAY Instruction is UNPREDICTABLE when executed in debug state.

ArchMsg.Warning.warning_unpredictable_mair_encoding

TAGS UNPREDICTABLE DISPLAY Memory attribute %{ATTR} has invalid MAIR encoding; this may lead to unpredictable results. Fields:

ATTR unsigned int

Memory attribute.

ArchMsg.Warning.warning_unpredictable_pmu_counter_access

TAGS UNPREDICTABLE DISPLAY Access to PMU counters from non-secure EL0 or EL1 is UNPREDICTABLE with MDCR_EL2.HPMN set to 0.

ArchMsg.Warning.warning_unpredictable_prioritization_breakpoint_match_vector_catch

TAGS UNPREDICTABLE DISPLAY Two events with the same priority occurred at the same instruction: Address Matching Vector Catch debug event, and Breakpoint debug event. It is constrained unpredictable which is taken.

ArchMsg.Warning.warning_unpredictable_stack_selection

TAGS UNPREDICTABLE DISPLAY UNPREDICTABLE state on context switch.

ArchMsg.Warning.warning_unpredictable_tsize_out_of_range

TAGS UNPREDICTABLE DISPLAY %{TCR}.T%{TTBR:d}SZ (%{TSIZE:d} bits) is out of range. Must be between 25 and %{TSIZE_MAX:d}. Fields:

TCR enum

Which TCR.

TSize signed int

Value of TSize.

TSize_MAX signed int

The maximum allowed value.

TTBR unsigned int

Which TTBR region.

ArchMsg.Warning.warning_unpredictable_unaligned_pc_as_base_register

TAGS **UNPREDICTABLE** DISPLAY The PC value must be word-aligned, otherwise the behavior of the instruction is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_unaligned_pop_stack

TAGS **UNPREDICTABLE** DISPLAY **CONSTRAINED UNPREDICTABLE** unaligned stack frame `%{FRAME_ADDR:x}` in exception-return. Fields:

FRAME_ADDR unsigned int

unaligned stack frame address.

ArchMsg.Warning.warning_unpredictable_unaligned_to_non_writeback

TAGS **UNPREDICTABLE** DISPLAY Unaligned Access to Non-Writeback-Cacheable Memory.

ArchMsg.Warning.warning_unpredictable_vmsa_memattrib

TAGS **UNPREDICTABLE** DISPLAY Unpredictable vmsa memory attribute with texcb value programmed to `%{TEXCB}` when tex remap is `%{TEX_REMAP}`. Fields:

TEXCB unsigned int

texcb value.

TEX_REMAP bool

use tex remap.

ArchMsg.Warning.warning_unpredictable_vtcr_t0sz_sl0

TAGS **UNPREDICTABLE** DISPLAY The combination of VTCR.TOSZ=`%{TOSZ:d}` and VTCR.SL0=`%{SL0:d}` is unpredictable. Fields:

SL0 signed int

VTCR.SL0, starting level for stage 2 translation table walks.

TOSZ signed int

VTCR.TOSZ, required input address range.

ArchMsg.Warning.warning_unpredictable_write_DHCSR

TAGS **UNPREDICTABLE** DISPLAY **UNPREDICTABLE** set of bit-changes in DHCSR. Fields:

NEW unsigned int

data attempted to be written.

OLD unsigned int

previous value.

ArchMsg.Warning.warning_unpredictable_write_SHCSR

TAGS UNPREDICTABLE DISPLAY UNPREDICTABLE setting a pending bit to 1 for an exception with priority \geq execution priority.

ArchMsg.Warning.warning_unsupported_access_to_memory_mapped_register

TAGS UNPREDICTABLE DISPLAY Access to memory mapped register is unsupported.

ArchMsg.Warning.warning_user_jmcr_access

TAGS UNPREDICTABLE DISPLAY User %{WRITE:(write|read)} access to JMCR is unpredictable. Fields:

WRITE unsigned int
access is write.

ArchMsg.Warning.warning_wcr_mask_and_bas

TAGS UNPREDICTABLE DISPLAY DBGWCR_EL1.MASK is non-zero and BAS is not 0xff.

ArchMsg.Warning.warning_wcr_mask_reserved

TAGS UNPREDICTABLE DISPLAY DBGWCR_ELn.MASK is set to a reserved value.

ArchMsg.Warning.warning_wcr_non_configuous_bas

TAGS UNPREDICTABLE DISPLAY DBGWCR_EL1.BAS has non-contiguous set of ones.

ArchMsg.Warning.warning_wvr_masked_bit_not_zero

TAGS UNPREDICTABLE DISPLAY DBGWVRn_EL1.VA contains a bit masked by DBGWCRn_EL1.MASK that is not zero.

ArchMsg.Why.why_branch_target_exception

A branch target exception has occurred. DISPLAY branch target exception: target pc=%{PC}, target opcode=%{OPCODE}. PRIMARY KEY PC. Fields:

BTYPE unsigned int
pstate.BTYPE at time branch taken.

OPCODE unsigned int
opcode at branch target.

PC unsigned int
Virtual Address of branch target.

ArchMsg.Why.why_illegal_state

An illegal mode change has occurred. DISPLAY illegal mode change: %{MESSAGE}. Fields:

MESSAGE string
The message.

BRANCH_MISPREDICT

Simulating branch mispredict. Fields:

PC unsigned int
Origin address (or 0 if unavailable).

BRA_DIR

Direct branches, to immediate address. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISSET enum

The instruction set of the branch instruction.

IS_COND bool

Indicates if this is a conditional branch.

IS_HINTED bool

Whether the conditional instruction is hinted.

IS_ISB bool

Whether the instruction is ISB.

IS_LINK bool

Indicates if this is a branch with link.

PC unsigned int

The address of the branch instruction.

TARGET_ISET enum

The instruction set after the branch.

TARGET_PC unsigned int

The address the instruction branches to.

BRA_INDIR

Indirect branches, perhaps to a register. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISSET enum

The instruction set of the branch instruction.

IS_COND bool

Indicates if this is a conditional branch.

IS_HINTED bool

Whether the conditional instruction is hinted.

IS_ISB bool

Whether the instruction is ISB.

IS_LINK bool

Indicates if this is a branch with link.

PC unsigned int

The address of the branch instruction.

TARGET_ISET enum

The instruction set after the branch.

TARGET_PC unsigned int

The address the instruction branches to.

BRA_INDIR_CCFAIL

Indirect branches fail, perhaps to a register.

BRA_RET_CCFAIL

Branch return fail, perhaps to a register.

CACHE_MAINTENANCE_OP

Cache Maintenance Operation. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

DATA unsigned int

Specified MVA or Set/way.

FUNCTION enum

Clean / Invalidate.

SCOPE enum

Affected region.

SIDE enum

Inst / Data.

CCFAIL

Conditional instruction condition check fail. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CCFAIL_UNC

Unconditional instruction condition check fail. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CCPASS

Conditional instruction condition check pass. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CCPASS_UNC

Unconditional instruction condition check pass. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CODE_CACHE_MAINT

Code cache maintenance. Fields:

END_INCL unsigned int

Inclusive end address for VA or PA requests.

NS bool

Non-secure world for PA.

START unsigned int

Start address for VA or PA requests.

TYPE enum

Request type.

COMPILE_BLOCK_END

Last instruction of basic block translated. Fields:

VADDR unsigned int

Address of next instruction after this basic block.

COMPILE_BLOCK_START

First instruction of basic block translated. Fields:

VADDR unsigned int

Address of first instruction in basic block.

COMPILE_INST

ARM instruction compiled. Fields:

DISASS string

Disassembly of the instruction.

ISET enum

The instruction set of this instruction.

ITSTATE unsigned int

The ITSTATE current for the instruction.

OPCODE unsigned int

The opcode of the instruction.

PC unsigned int

The address of the instruction.

SIZE unsigned int

The size of the instruction in bytes.

CONTEXTIDR

Context ID Register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

NS bool

Secure or nonsecure banked register is accessed.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The new value written.

CONTEXT_SYNC

Called for every context synchronization event. Fields:

ARCH_CSE bool

Architectural Context Synchronization Event.

PC unsigned int

The address of the synchronization instruction.

SYSREG_SYNC bool

System register synchronization event.

CORE_ENDIAN

Core BE8 Big-Endian state changed. Fields:

BE8 bool

Core BE8 Big-Endian state.

CORE_INFO

Static processor attributes. Only triggered by a call to DumpState(). Fields:

ARCH_PROFILE enum

The architecture profile of the core.

CLUSTER_ID unsigned int

The cluster ID of this processor.

CORE_NUM unsigned int

The number of this core in an MP processor.

DETAILED_CORE_FEATURES enum

Detailed core features as a bitfield. Enum values are bitmasks to extract the info from the field.

FPU_VERSION enum

The VFP version implemented by the core.

MEM_ARCH enum

The memory architecture of the core.

MIDR_EL1 unsigned int

MIDR_EL1 value.

NUM_CORES unsigned int

The number of cores in this MP processor.

QUANTUM_SIZE unsigned int

The default quantum size of the core (in ticks).

SECURITY_FEATURES bool

Does the core have security features?.

CORE_LOADS

Processor load accesses. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ACQREL enum

Is this an acquire/release.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DATA unsigned int

The data read or written.

ELEMENT_SIZE unsigned int

Width of each element.

ENDIANNESS enum

Byte ordering of this access.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

LOCK enum

Normal, exclusive or locked access.

MECID unsigned int

Memory Encryption Context ID, if applicable.

MECID2 unsigned int

Memory Encryption Context ID of the second page, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PAGE_SIZE unsigned int

Page size as $\log_2(\text{size in bytes})$.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK, 2=Exclusive Failed, 3=Unsupported.

SIZE unsigned int

Width of the access in bytes. Only required if DATA is not traced.

TRANS bool

Is this a translated access.

VADDR unsigned int

The virtual address of the access.

CORE_REGS

Changes of the core registers R0 to R14. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number, 0 to 14.

MODE enum

Bank of the register accessed.

OLD_VALUE unsigned int

The old value overwritten.

PHYS_ID enum

The physical register accessed.

VALUE unsigned int

The new value written to the register.

CORE_REGS64

Changes of the core registers X0..X30, SP_ELn. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ID enum

The register number, 0-30 for X0-X30, >=32 for SP_ELn.

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

CORE_REGS64_READ

General purpose AArch64 register has been read X0..X30, SP_ELn. Fields:

ID enum

The register number, 0-30 for X0-X30, >=32 for SP_ELn.

VALUE unsigned int

The value read from the register.

CORE_STALLED_TRANSITION

Detect when a core is prevented from progressing PChannel states. Fields:

CURRENT_STATE enum

Current power mode of core.

REASON string

The reason a transition was stalled, and what must be done to unblock it.

TARGET_STATE enum

Desired power mode of core after transition.

CORE_STORES

Processor store accesses. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ACQREL enum

Is this an acquire/release.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DATA unsigned int

The data read or written.

ELEMENT_SIZE unsigned int

Width of each element.

ENDIANNESS enum

Byte ordering of this access.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

LOCK enum

Normal, exclusive or locked access.

MECID unsigned int

Memory Encryption Context ID, if applicable.

MECID2 unsigned int

Memory Encryption Context ID of the second page, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PAGE_SIZE unsigned int

Page size as $\log_2(\text{size in bytes})$.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK, 2=Exclusive Failed, 3=Unsupported.

SIZE unsigned int

Width of the access in bytes. Only required if DATA is not traced.

TRANS bool

Is this a translated access.

VADDR unsigned int

The virtual address of the access.

CP14_READ

System Coprocessor register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP14 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc1 unsigned int

opcode 1.

opc2 unsigned int

opcode 2.

CP14_WRITE

System Coprocessor register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP14 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc1 unsigned int

opcode 1.

opc2 unsigned int

opcode 2.

CP15_READ

System Coprocessor register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP15 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc1 unsigned int

opcode 1.

opc2 unsigned int

opcode 2.

CP15_READ64

System Coprocessor register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP15 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc unsigned int

opcode 1.

CP15_WRITE

System Control Coprocessor register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP15 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc1 unsigned int

opcode 1.

opc2 unsigned int
opcode 2.

CP15_WRITE64

System Coprocessor register write. Fields:

CORE_NUM unsigned int
Core number in a multi processor.

CRm unsigned int
CRm.

NS bool
Secure or nonsecure banked register is accessed.

REG_NAME string
Name of the CP15 register accessed.

SECURITY_STATE enum
The current security state.

UNDEF bool
The register accessed is undefined.

UPDATED_VALUE unsigned int
Updated value of the register now it has been written.

VALUE unsigned int
The new value written.

opc unsigned int
opcode 1.

CPSR

CPSR change. (In AArch64 PSTATE represented in SPSR format). Fields:

CORE_NUM unsigned int
Core number in a multi processor.

OLD_VALUE unsigned int
The old CPSR value.

UNKNOWN unsigned int
Bits within the register that have unknown value.

VALUE unsigned int
The new CPSR value.

CRYPTO_SPEC

Every crypto instruction speculatively executed.

CorePowerStateChange

Core power-up state has changed. DISPLAY Core powered %{PU:(up|down)}. Fields:

PU bool

Whether the core powered up.

DATA_CACHE_SET_ALLOC_TAGS

Set allocation tags from DC GVA instruction. Fields:

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

NSDESC unsigned int

The security state of the access.

PADDR unsigned int

The physical address of the access.

PAS enum

The physical address space of the page.

RESPONSE enum

0=Aborted, 1=OK.

SIZE unsigned int

The size of memory in bytes.

TAG unsigned int

The value of the tag being set.

TAGGING_TYPE enum

Type of Memory tagging (Physical or Virtual).

TAG_VADDR unsigned int

The virtual address of the Virtual Tag if Virtual tagging is used; otherwise 0.

VADDR unsigned int

Virtual address used for setting allocation tags.

DATA_CACHE_ZERO

Zero and invalidate cache line from DC ZVA instruction. Fields:

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

NSDESC unsigned int

The security state of the access.

PADDR unsigned int

The physical address of the access.

PADDR_TAG unsigned int

If tag checked, the physical address tag for the access.

PAS enum

The physical address space of the page.

RESPONSE enum

0=Aborted, 1=OK.

SIZE unsigned int

The size of zero'd memory in bytes.

TAG_CHECKED bool

Is this access tag checked.

VADDR unsigned int

Virtual address of the zero'd cache line.

DEBUG_EVENT

Hardware debug support event. Fields:

EVENT enum

Description of event.

VALUE unsigned int

data value.

DEBUG_TARGET_ISET

Hardware debug target ISA. Fields:

TARGET_ISET enum

The target instruction set of the debug state.

DMI_ALLOCATE

DMI allocated. Fields:

ACCESS_DIRECTION enum

READ / WRITE.

PADDR unsigned int

The physical address of the DMI.

PTR unsigned int

DMI range pointer.

SIDE enum

DSIDE / ISIDE.

SIZE unsigned int

The size of DMI.

VADDR unsigned int

The virtual address of the DMI..

DMI_HIT

DMI hit, access used DMI (only fire for small DMI regions i.e. less than a full page). Fields:

ACCESS_DIRECTION enum

READ / WRITE.

PADDR unsigned int

The physical address of the DMI.

SIDE enum

DSIDE / ISIDE.

SIZE unsigned int

The size of the DMI.

VADDR unsigned int

The virtual address of the DMI..

DMI_REVOKE

DMI revoked. Fields:

ACCESS_DIRECTION enum

READ / WRITE.

PADDR unsigned int

The physical address of the DMI.

SIDE enum

DSIDE / ISIDE.

SIZE unsigned int

The size of DMI.

VADDR unsigned int

The virtual address of the DMI..

END_COMPILE

Compilation end. Fields:

END_OF_PAGE_COUNT unsigned int

Number of basic blocks exited due to page boundary since START_COMPILE.

FETCHFAIL_COUNT unsigned int

Number of basic blocks exited due to ifetch failure START_COMPILE.

INST_COUNT unsigned int

Number of instructions compiled since START_COMPILE.

NONSEQ_COUNT unsigned int

Number of basic blocks exited due to non-sequential instructions since START_COMPILE.

PAGE_STRADDLE_COUNT unsigned int

Number of basic blocks exited due to unaligned instructions crossing page since START_COMPILE.

ESB

Error Synchronization Barrier hit. Fields:

ASYNC bool

ESB is async.

IMPLICIT bool

ESB is implicit.

EXCEPTION

Exception taken. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ESR unsigned int

The value assigned to ESR when exception is taken.

ESR_EC unsigned int

The Value assigned to ESR.EC when exception is taken.

IS_PHYSICAL bool

Physical or Virtual exception.

IS_TRAP bool

Exception coming from a trap.

LR unsigned int

The value assigned to the link register.

PC unsigned int

The location where the exception occurred.

PREFERRED_RETURN unsigned int

The preferred return address for the exception.

TAKEN_LOCALLY bool

Whether exception is taken locally(same exception level).

TARGET_ISET enum

The instruction set of the exception handler code.

TARGET_PC unsigned int

The address the exception branches to.

VECTOR enum

The exception vector.

VECTOR_OFFSET unsigned int

The offset of an exception vector from the base address.

EXCEPTION_END

Every exception completed.

EXCEPTION_INFO

Trace human-readable reason of the current exception for data/instruction aborts, floating point exceptions and unknown exceptions. Fields:

DESCRIPTION string

Reason for current exception.

ESR_EC unsigned int

The value of ESR.EC when exception is taken.

EXCEPTION_RAISE

Every exception raised.

EXCEPTION_RETURN

Branches on leaving exception. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISSET enum

The instruction set of the branch instruction.

IS_COND bool

Indicates if this is a conditional branch.

IS_HINTED bool

Whether the conditional instruction is hinted.

IS_ISB bool

Whether the instruction is ISB.

IS_LINK bool

Indicates if this is a branch with link.

PC unsigned int

The address of the branch instruction.

TARGET_ISET enum

The instruction set after the branch.

TARGET_PC unsigned int

The address the instruction branches to.

EXCEPTION_RETURN_PREBRANCH

Exception return event that must occur before PC and processor state have updated.

EXCEPTION_START

Every exception started.

FIQ_TAKEN

FIQ taken exception.

FP_EXCEPTIONS_SIGNALLED

Trace which FP exceptions were signalled as a part of the FP operations performed by an instruction. Fields:

FP_EXCEPTIONS_SIGNALLED_MASK unsigned int

Indicates which FP exceptions were signalled by an instruction. This uses the same encoding as the cumulative floating-point exception bitfields in FPSR.

FP_STATE

Floating point state. Fields:

FPSR unsigned int

FPSR register value.

MASK unsigned int

The mask indicating updated cumulative bits.

FREQ_CHANGED

Core frequency changed. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The instruction count of this core.

NEW_FREQ signed int

The new frequency of this core (expressed in Hz).

OLD_FREQ signed int

The old frequency of this core (expressed in Hz).

HLT

HLT instruction occurred. Fields:

IMM16 unsigned int

The 16-bit immediate value encoded in HLT instruction.

INST

Every instruction executed. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DEBUG_STATE bool

The instruction is executed in debug state.

DISASS string

Disassembly of instruction.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISET enum

The current instruction set.

ITSTATE unsigned int

The current ITSTATE.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

MODE enum

The mode the core is in.

NS unsigned int

The core's non-secure bit.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

OPCODE unsigned int

The opcode of the instruction.

PADDR unsigned int

The physical address of the instruction.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the instruction.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the page.

PC unsigned int

The address of the instruction.

SECURITY_STATE enum

The core's security state.

SIZE unsigned int

The size of the instruction in bytes.

INST_END

Every instruction completed.

INST_START

Every instruction started. Fields:

DEBUG_STATE **bool**

The instruction is executed in debug state.

ISET **enum**

The current instruction set.

MODE **enum**

The mode the core is in.

NS **enum**

The current Secure State.

PC **unsigned int**

The address of the conditional instruction.

SECURITY_STATE **enum**

The current Security State.

INST_STRADDLE

Instruction straddles boundary. Fields:

DEBUG_STATE **bool**

The instruction is executed in debug state.

ISET **enum**

The current instruction set.

MODE **enum**

The mode the core is in.

NS **enum**

The current Secure State.

PC **unsigned int**

The address of the conditional instruction.

SECURITY_STATE **enum**

The current Security State.

IRQ_TAKEN

IRQ taken exception.

ISB

ISB instruction occurred. Fields:

PC **unsigned int**

The address of the ISB instruction.

LOAD_MULTIPLE_REGS_LIST

List of destination registers (not source) of a multiple registers load instruction. Fields:

REG string

Register to add into the list.

LOCAL_MONITOR

Local monitor activity. Fields:

PADDR unsigned int

Local Monitor Address.

State enum

State of the monitor (Open/Exclusive).

MAPPING_CREATE

A memory mapping has been created. This is an implementation detail of the model and so may change without notice. Fields:

ASID unsigned int

The ASID used during translation, or 0 if no ASID was used.

IS_VALID bool

Whether the mapping is valid or not.

PADDR unsigned int

Base physical address of region, or 0 if not known.

SIDE enum

I-side or D-side.

SIZE unsigned int

Size of the page, or 0 if not known.

VADDR unsigned int

Base virtual address of region.

VMID unsigned int

The VMID used during translation, or 0 if no VMID was used.

MAPPING_DESTROY

A memory mapping has been destroyed. This is an implementation detail of the model and so may change without notice. Fields:

ASID unsigned int

The ASID used during translation.

CAUSE enum

The cause of invalidation.

PADDR unsigned int

Base physical address of page.

SIDE enum

I-side or D-side.

SIZE **unsigned int**

Size of the page.

VADDR **unsigned int**

Base virtual address of page.

VMID **unsigned int**

The VMID used during translation.

MEMMAP_DEBUG_READ

Memory mapped reads to the debug registers. Fields:

ADDR **unsigned int**

address.

CORE_NUM **unsigned int**

Core number in a multi processor.

EXT **bool**

Whether access is from an external device (such as the DAP).

NS **enum**

Secure state of the access.

REG_NAME **string**

Name of the debug register accessed.

UNDEF **bool**

The register accessed is undefined.

VALUE **unsigned int**

The value read.

MEMMAP_DEBUG_WRITE

Memory mapped writes to the debug registers. Fields:

ADDR **unsigned int**

address.

CORE_NUM **unsigned int**

Core number in a multi processor.

EXT **bool**

Whether access is from an external device (such as the DAP).

NS **enum**

Secure state of the access.

REG_NAME **string**

Name of the debug register accessed.

UNDEF **bool**

The register accessed is undefined.

VALUE unsigned int

The new value written.

MMU_D128_TTB_READ

This event is triggered by reads caused by a D128 translation table walk. Fields:

ABORTED bool

Was the walk successful.

DATA_HIGH unsigned int

The data upper bits read.

DATA_LOW unsigned int

The data lower bits read.

IPA unsigned int

For stage 1, the IPA of the read.

LEVEL unsigned int

Translation table level.

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

OUTADDR unsigned int

The output physical address.

PADDR unsigned int

The physical address of the read.

PASmem enum

Physical Address Space of the accessed memory.

PASreq enum

Physical Address Space of the requested lookup.

REGIME_EL enum

Entry matches in this translation regime.

REQUESTER enum

The requester of this table walk.

SIDE enum

Inst / Data.

STAGE unsigned int

Translation stage.

MMU_D128_TTB_WRITE

This event is triggered by writes caused by a D128 translation table walk. Fields:

ABORTED bool

Was the walk successful.

DATA_HIGH unsigned int

The data upper bits written.

DATA_LOW unsigned int

The data lower bits written.

IPA unsigned int

For stage 1, the IPA of the write.

IS_HACDBS_ACCESS bool

If the access was due to HACDBS_CLEAN.

LEVEL unsigned int

Translation table level.

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

OUTADDR unsigned int

The output physical address.

PADDR unsigned int

The physical address of the write.

PASmem enum

Physical Address Space of the accessed memory.

PASreq enum

Physical Address Space of the requested lookup.

REGIME_EL enum

Entry matches in this translation regime.

SIDE enum

Inst / Data.

STAGE unsigned int

Translation stage.

MMU_TRANS

Address translation information. Fields:

ASID unsigned int

Address space identifier.

CORE_NUM unsigned int

Core number in a multi processor.

Hyp bool

Entry matches in Hyp state only.

INNERCACHE_RA bool

Is the inner cache allocate on read.

INNERCACHE_TRANSIENT bool

Is the inner write-through transient.

INNERCACHE_TYPE enum

Inner Caching scheme (NC/MB/WA).

INNERCACHE_WA bool

Is the inner cache allocate on write.

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MEMTYPE enum

Memory type.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC enum

Is secure side supposed to access secure or nonsecure memory.

OUTERCACHE_RA bool

Is the outer cache allocate on read.

OUTERCACHE_TRANSIENT bool

Is the outer write-through transient.

OUTERCACHE_TYPE enum

Outer Caching scheme (NC/MB/WA).

OUTERCACHE_WA bool

Is the outer cache allocate on write.

PADDR unsigned int

Physical address of the access.

PAGESIZE unsigned int

Page size as log2(size).

PAS enum

PAS supposed to be accessed is secure or nonsecure memory or Root/Realm if RME is implemented.

SH enum

Shareability.

SIDE enum

Inst / Data.

STAGE1_PERM unsigned int

Stage 1 permissions, [5:3] Privileged access (XWR) [2:0] User access (XWR).

STAGE2_PERM unsigned int

Stage 2 permission mask [5:3] Privileged access (XWR) [2:0] User access (XWR).

VADDR unsigned int

Virtual address of the access.

VMID unsigned int

Virtual machine identifier.

XS bool

XS Attribute.

nG enum

Flag indicating whether ASID will be matched.

MMU_TTB_READ

This event is triggered by reads caused by a translation table walk. Fields:

ABORTED bool

Was the walk successful.

DATA unsigned int

The data read.

DESC_KIND enum

The encoding scheme of the descriptors used for this walk.

IPA unsigned int

For stage 1, the IPA of the read.

LEVEL unsigned int

Translation table level.

LPAE bool

Is this for an LPAE translation (*DEPRECATED*, use DESC_KIND).

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSmem bool

Non secure memory access.

NSreq bool

Non secure request.

OUTADDR unsigned int

The output physical address.

PADDR unsigned int

The physical address of the read.

PASmem enum

Physical Address Space of the accessed memory.

PASreq enum

Physical Address Space of the requested lookup.

REGIME_EL enum

Entry matches in this translation regime.

REQUESTER enum

The requester of this table walk.

SIDE enum

Inst / Data.

STAGE unsigned int

Translation stage.

MMU_TTB_WRITE

This event is triggered by writes caused by a translation table walk. Fields:

ABORTED bool

Was the walk successful.

DATA unsigned int

The data written.

DESC_KIND enum

The encoding scheme of the descriptors used for this walk.

IPA unsigned int

For stage 1, the IPA of the write.

IS_HACDBS_ACCESS bool

If the access was due to HACDBS_CLEAN.

LEVEL unsigned int

Translation table level.

LPAE bool

Is this for an LPAE translation (*DEPRECATED*, use DESC_KIND).

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSmem bool

Non secure memory access.

NSreq bool

Non secure request.

OUTADDR unsigned int

The output physical address.

PADDR unsigned int

The physical address of the write.

PASmem enum

Physical Address Space of the accessed memory.

PASreq enum

Physical Address Space of the requested lookup.

REGIME_EL enum

Entry matches in this translation regime.

SIDE enum

Inst / Data.

STAGE unsigned int

Translation stage.

MODE_CHANGE

Mode change. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

MODE enum

The new mode.

NON_SECURE enum

The core's new non-secure bit.

OLD_MODE enum

The old mode.

SECURITY_STATE enum

The new security state.

MPU_TRANS

Address translation information. Fields:

ACCESS_STATUS enum

Access status(0xAA/0xBB/0xDD/0xEE=overlap/bg hit/default/bg fault).

CORE_NUM unsigned int

Core number in a multi processor.

EXEC_PERM enum

Execution Permission.

INNERCACHE_RA bool

Is the inner cache allocate on read.

INNERCACHE_TYPE enum

Inner Caching scheme (NC/MB/WA).

INNERCACHE_WA bool

Is the inner cache allocate on write.

MEMTYPE enum

Memory type.

NSDESC enum

Is non secure access.

OUTERCACHE_RA bool

Is the outer cache allocate on read.

OUTERCACHE_TYPE enum

Outer Caching scheme (NC/MB/WA).

OUTERCACHE_WA bool

Is the outer cache allocate on write.

PADDR unsigned int

Address of the access.

PAGESIZE unsigned int

Page size as $\log_2(\text{size})$.

PAS enum

PAS accessed is secure or nonsecure memory.

READ_PERM enum

Read Permission.

REG_NUM unsigned int

MPU region-number 0-31.

SH enum

Shareability.

SIDE enum

Inst / Data.

STAGE unsigned int

The translation stage.

WRITE_PERM enum

Write Permission.

NOT_TAG_CHECKED_INST

MTE Tag not checked for current instruction.

PERIODIC

Called for every quantum. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The instruction count of this CPU.

PC unsigned int

The address of the next instruction to be executed on this CPU.

PMU_COUNTER_OVERFLOW

PMU counter overflow. Fields:

COUNTER_GROUP enum

Counter group.

COUNTER_RANGE enum

Counter range.

EVENT_ID unsigned int

Event Identifier, field is valid only for Event counter group.

INDEX unsigned int

Counter index (as selected by PMSELR), field is valid only for Event/Cycle counter groups.

INTERRUPT bool

Is interrupt enabled on overflow for this counter.

POINTER_AUTH

Called when execution hits a pointer authentication check. Fields:

AUTH_STATE bool

Authentication result.

KEY_USED enum

Pointer authentication key.

POTINDEX_UPDATE

Value of POT index for current pc changed. Fields:

NEW_POTINDEX unsigned int

New POTIndex.

OLD_POTINDEX unsigned int

Old POTIndex.

PREFETCH_MEMORY64

Prefetch from PRFM or PRFUM instructions. Fields:

PRFOP enum

Prefetch hint.

VADDR unsigned int

Virtual address of the location that should be prefetched.

PRELOAD64_REQUEST

Preload requests from PRFM or PRFUM instructions. Fields:

PRFOP enum

Prefetch hint.

VADDR unsigned int

Virtual address of the location that should be prefetched.

PRELOAD_DATA

Data preload from PLD instruction. Fields:

VADDR unsigned int

Virtual address of the data that should be preloaded.

PRELOAD_INST

Instruction preload from PLI instruction. Fields:

VADDR unsigned int

Virtual address of the instruction that should be preloaded.

PRE_CORE_LOAD

Trace just before a core load. Fields:

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

PRE_CORE_STORE

Trace just before a core store. Fields:

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

PRE_TTB_READ

This event is triggered before reads caused by a translation table walk. Fields:

ADDR unsigned int

The physical address of the read.

PSTATE

PSTATE fields update. Fields:

ALLINT unsigned int

All interrupt mask bit.

BTYPE unsigned int

Branch target identification bit.

CORE_NUM unsigned int

Core number in a multi processor.

DAIF unsigned int

Exception mask bits.

DIT unsigned int

Data Independent Timing bit.

EL unsigned int

Current Exception level field.

EXLOCK unsigned int

Exception return state lock bit.

IL unsigned int

Illegal Execution state bit.

NZCV unsigned int

Condition flags.

PACM unsigned int

PACM bit.

PAN unsigned int

Privileged Access Never state bit.

PM unsigned int

PMU exception mask bit.

PPEND unsigned int

PMU exception pending bit.

SM unsigned int

StreamingMode status bit.

SP unsigned int

Stack pointer register selection bit.

SS unsigned int

Software Step bit.

SSBS unsigned int

Speculative Store Bypass Safe bit.

TCO unsigned int

Tag Check Override bit.

TOUCHED unsigned int

Mask of the fields that have been touched since this trace has last been emitted. Note that touched does not necessarily mean the value changed. The index of the field is the bit index in the mask plus the index of this field plus one.

UAO unsigned int

User Access Override bit.

UINJ unsigned int

Undefined Instruction exceptions bit.

ZA unsigned int

ZA storage and ZT0 register enable bit.

nRW unsigned int

Current Execution state bit.

PSTATE_PACTIVE_OPMODE_TRANSITION

Trace pactive and Opmode mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Operating mode after the transition.

OLD_STATE enum

Operating mode before the transition.

PSTATE_PACTIVE_OPMODE_TRANSITION_DENIED

Trace the denied pactive and Opmode mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Operating mode after the transition.

OLD_STATE enum

Operating mode before the transition.

REASON string

Reason for the transition being denied.

PSTATE_PACTIVE_POWER_TRANSITION

Trace pactive and power mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Power mode after the transition.

OLD_STATE enum

Power mode before the transition.

PSTATE_PACTIVE_POWER_TRANSITION_DENIED

Trace the denied pactive and power mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Power mode after the transition.

OLD_STATE enum

Power mode before the transition.

REASON string

Reason for the transition being denied.

PSTATE_PACTIVE_THREAD_OPMODE_TRANSITION

Trace pactive and Opmode mode transitions for threads on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Operating mode after the transition.

OLD_STATE enum

Operating mode before the transition.

QUANTUM_START

Called at the start of every quantum.

RANGE_PRELOAD_REQUEST

Preload requests from RPRFM. Fields:

COUNT unsigned int

Number of blocks of data to access.

LENGTH unsigned int

Number of contiguous bytes to be accessed in each block.

OP unsigned int

Range prefetch operation .

REUSE unsigned int

Maximum number of bytes to be accessed before next RPRFM .

STRIDE signed int

Stride to advance after accessing LENGTH bytes.

VADDR unsigned int

Base virtual address of range prefetch request.

RUN_STATE

Run state transition. Fields:

INST_COUNT unsigned int

Ticks count at point of transition.

NEW enum

New run state.

OLD enum

Old run state.

SEMIHOSTING_CALL

Call of a semihost function occurred. Fields:

PC unsigned int

The program counter after the semihosting call.

SEMIHOSTING_PRECALL

About to call semihost. Fields:

EL2 bool

Use EL2 translation regime.

NS bool

Is Non-Secure.

REG_WIDTH unsigned int

The current register width in bytes.

SIGNAL

External signal state change. Fields:

SIGNAL enum

Signal that changed.

STATE bool

Signal asserted state.

SOFTWARE_STEP

Return the Debug Software Step state. Fields:

STATE enum

Software Step state.

SPSR

SPSR change. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

MODE enum

Which of the banked SPSR registers is written.

OLD_VALUE unsigned int

The old SPSR value.

VALUE unsigned int

The new SPSR value.

START_COMPILE

Compilation started. Fields:

VADDR unsigned int

Instruction where compilation begins.

STG_MULTIPLE_MICRO_OPS_ADDRESS_RANGE

Address range for MTE stores with multiple uops. Fields:

ADDR_END unsigned int

Final address.

ADDR_START unsigned int

Starting address.

STORE_MULTIPLE_REGS_ADDRESS_RANGE

Address range for stores with multiple registers. Fields:

ADDR_END unsigned int

Final address.

ADDR_START unsigned int

Starting address.

SUBOPTIMAL_LDST_RETIRED

suboptimal load/store. Fields:

ACCESS_SIZE unsigned int

Log2 of the access width in bytes used for alignment checking (1: half-word, 2: word ...).

OPTIMAL_ALIGN_SELECTOR unsigned int

Selector to decide the optimal alignment size (1: access width, 2: 4-bytes, 3: 8-bytes, ..., 12: 4-kbytes).

VADDR unsigned int

The virtual address of the access.

SYNC

Called for every synchronization, that is every quantum break. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The instruction count of this CPU.

LOCAL_QUANTUM unsigned int

The local quantum of this CPU.

LOCAL_TIME unsigned int

The local time of this CPU.

SYNC_REASON enum

Reason for sync.

SYSCALL

System call instruction executed. Fields:

IMM unsigned int

Immediate value of the system call instruction.

TYPE enum

System call type.

VADDR unsigned int

Instruction that caused the system call.

SYSREG_READ128

System Coprocessor 128 bit register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

REGNUM enum

Internal register number.

REGNUM_PHYS enum

Internal register number physically accessed (e.g. in presense of VHE redirection).

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_READ64

System Coprocessor register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REGNUM enum

Internal register number.

REGNUM_OPERAND enum

Register number corresponding to the instruction operand.

REGNUM_PHYS enum

Internal register number physically accessed (e.g. in presense of VHE redirection).

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_UPDATE128

Triggers when the system updates a register. Fields:

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG enum

Register number.

UNKNOWN unsigned int

Bits of the register which became unknown.

VALUE unsigned int

Value written to the register.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_UPDATE32

Triggers when the system updates a register. Fields:

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG enum

Register number.

UNKNOWN unsigned int

Bits of the register which became unknown.

VALUE unsigned int

Value written to the register.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_UPDATE64

Triggers when the system updates a register. Fields:

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG enum

Register number.

UNKNOWN unsigned int

Bits of the register which became unknown.

VALUE unsigned int

Value written to the register.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_WRITE128

System Coprocessor 128 bit register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

REGNUM enum

Internal register number.

REGNUM_PHYS enum

Internal register number physically accessed (e.g. in presense of VHE redirection).

UNDEF bool

The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_WRITE64

System Coprocessor register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REGNUM enum

Internal register number.

REGNUM_OPERAND enum

Register number corresponding to the instruction operand.

REGNUM_PHYS enum

Internal register number physically accessed (e.g. in presense of VHE redirection).

UNDEF bool

The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

UNALIGNED_LDST_RETIRED

Processor unaligned load/store. Fields:

ACCESS_SIZE unsigned int

Log2 of the access width in bytes used for alignment checking (1=>half-word, 2->word ...).

NUMBER_OF_BEATS unsigned int

Number of accesses (beats) in this burst (Total size in bytes is ACCESS_SIZE * NUMBER_OF_BEATS).

VADDR unsigned int

The virtual address of the access.

VFP_D_REGS

VFP/NEON D 64 bit register write. Fields:

ALIASING enum

Type of register aliasing used.

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a write to the corresponding bit in the register. For example, 0x1 would indicate a write to VALUE[0].

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

VFP_D_REGS_READ

VFP/NEON D 64 bit register read. Fields:

ID unsigned int

The register number.

VALUE unsigned int

The current value of the register.

VFP_Q_REGS

VFP/NEON Q 128 bit register write. Fields:

ALIASING enum

Type of register aliasing used.

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a write to the corresponding byte in the register. For example, 0x1 would indicate a write to VALUE[7:0].

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

VFP_Q_REGS_READ

VFP/NEON Q 128 bit register read. Fields:

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a read to the corresponding byte in the register. For example, 0x1 would indicate a read to VALUE[7:0].

VALUE unsigned int

The current value of the register.

VFP_SYS_REGS

Writes to the VFP/NEON units system registers. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ID enum

Which VFP system register is written.

MASK unsigned int

The mask indicating updated cumulative bits.

OLD_VALUE unsigned int

The register's old value overwritten.

VALUE unsigned int

The new value written to the VFP system register.

VFP_S_REGS

VFP/NEON S 32 bit register write. Fields:

ALIASING enum

Type of register aliasing used.

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a write to the corresponding bit in the register. For example, 0x1 would indicate a write to VALUE[0].

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

VFP_S_REGS_READ

VFP/NEON S 32 bit register read. Fields:

ID unsigned int

The register number.

VALUE unsigned int

The current value of the register.

VIRTUAL_TAG_ALLOCATED

Virtual tag information when allocated. Fields:

TAG_VADDR unsigned int

Virtual tag virtual address.

VADDR unsigned int

Data Virtual address.

VIRTUAL_TAG unsigned int

Virtual tag value.

WAYPOINT

Signals end of basic block execution. Fields:

CORE_NUM *unsigned int*

Core number in a multi processor.

ISET *enum*

Origin instruction set.

IS_COND *bool*

Indicates if this is a conditional waypoint.

PC *unsigned int*

Origin address (or 0 if unavailable).

TAKEN *bool*

Indicates if this waypoint was taken.

TARGET *unsigned int*

Destination address.

TARGET_ISET *enum*

Destination instruction set.

WFE_END

WFE ended. Fields:

INST_COUNT *unsigned int*

Ticks count when leaving WFE.

WFE_EVENT_REGISTER

WFE event register status: set/clear, reason. Fields:

INST_COUNT *unsigned int*

Ticks count.

REASON *enum*

Reason for set/clear. Only REASON==0 (Cleared by WFE) clears the bit, all other reasons set it.

WFE_IGNORED

WFE ignored. Fields:

EVENT *bool*

This WFE was ignored because the event register was set.

INST_COUNT *unsigned int*

Ticks count when ignoring WFE.

TRAPPED *bool*

This WFE was trapped.

WFE_START

WFE entered. Fields:

INST_COUNT **unsigned int**

Ticks count when entering WFE.

WFI_END

WFI ended. Fields:

INST_COUNT **unsigned int**

Ticks count when leaving WFI.

WFI_IGNORED

WFI ignored. Fields:

DISABLED **bool**

This WFI was ignored because WFI is disabled.

INST_COUNT **unsigned int**

Ticks count when ignoring WFI.

REASON **enum**

specifies reason why WFI trace was ignored.

TRAPPED **bool**

This WFI was trapped.

WFI_START

WFI entered. Fields:

INST_COUNT **unsigned int**

Ticks count when entering WFI.

WFI_WAKEUP

WFI wakeup. Fields:

INST_COUNT **unsigned int**

Ticks count when WFI wakeup occurred.

REASON **enum**

Reason for wakeup.

2.55 ARM_Cortex-R82AE

This section describes the trace sources.

AA64_ASE_SVE_REGS

Modification of AArch64 Advanced SIMD or SVE Vector Registers (via SIMD or Scalar operations), SVE Predicate Registers and ZT0. Fields:

ID enum

The register identifier.

MASK unsigned int

Bitmask of the register to signal the modified bits in the VALUE field.

SM bool

Whether the PE is in Streaming Mode.

VALUE unsigned int

Value of AArch64 vector register.

ASYNC_MEMORY_FAULT

Context ID Register write. Fields:

FAULT unsigned int

Fault status present in ESR.ISS encoding for Serror in ARCH64 and present in DFSR encoding for external abort format in ARCH32.

PADDR unsigned int

Physical Address (or 0 if unavailable).

VADDR signed int

Virtual Address (or 0 if unavailable).

ATOMIC_END_ACCESS

Bus trace access for atomics. Fields:

ACCESS_FAIL bool

Memory access failed.

ADDR unsigned int

The virtual address of the access.

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

COMPARE_VALUE unsigned int

Compare value for CAS.

LOAD_VALUE unsigned int

Loaded values.

NSDESC unsigned int

The security state of the access.

OPERAND_VALUE unsigned int

Operation's operand.

OPERATION enum

Operation type.

PADDR unsigned int

The physical address of the access.

PRIV bool

Is this a privileged access?.

ATOMIC_START_ACCESS

Bus trace access for atomics. Fields:

ADDR unsigned int

The virtual address of the access.

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

COMPARE_VALUE unsigned int

Compare value for CAS.

NSDESC unsigned int

The security state of the access.

OPERAND_VALUE unsigned int

Operation's operand.

OPERATION enum

Operation type.

PADDR unsigned int

The physical address of the access.

PRIV bool

Is this a privileged access?.

ArchMsg.Error.exit_code

DISPLAY exit code: %{EXITCODE} component kind: %{KIND} reason: %{REASON}. Fields:

EXITCODE signed int

Exit code to be returned.

KIND string

Component kind that invoked the exit code trace.

REASON string

Reason why the exit code trace was invoked (optional).

ArchMsg.Info.invalid_region_programmed

DISPLAY MPU Stage %{STAGE:d}: Region ID %{REGION_ID:d} has base address greater than end address Base Addr=%{BASE_ADDR:x}, End Addr=%{END_ADDR:x}. Fields:

BASE_ADDR unsigned int

Base Address.

END_ADDR unsigned int

End Address.

REGION_ID unsigned int

Region ID.

STAGE unsigned int

MPU Stage.

ArchMsg.Warning.IgnoredWarmResetRequest

Event fires when software attempts to request warm reset through write to deprecated DBGPRCR.CWRR.

ArchMsg.Warning.RAS_ErrselrUnpredictableState

DISPLAY ERRSELR is in unpredictable state due to value greater than number_of_error_records written to it.

ArchMsg.Warning.RAS_UnpredictableStateWrite

DISPLAY %{REG_NAME} write ignored due to unpredictable ERRSELR state. Fields:

REG_NAME string

Name of register being written.

ArchMsg.Warning.ReportedRevisionRegisterRead

DISPLAY Reported revision and patch do not match that of the cpu. Fields:

ORIGINAL_VALUE unsigned int

The value the register would have based on the modelled revision.

REGISTER string

The name of the register that has ben read.

REPORTED_VALUE unsigned int

The value the register has been overridden to.

ArchMsg.Warning.Unpredictably Indexed PM Event Register

TAGS UNPREDICTABLE DISPLAY Event %{IsDirect:(Selector value|Register)} %{SEL} is >= %{N}. Fields:

IsDirect unsigned int

Direct Access rather than indirect.

N unsigned int

Number of Accessible Registers.

SEL unsigned int

Selector.

ArchMsg.Warning.cache_contents_unknown

A cache has been activated without being invalidated and may contain **UNKNOWN** content. DISPLAY %{SIDE:(D|I)}-Cache was enabled in %{REGIME:(SECURE|NON_SECURE|HYP)} regime but was not invalidated since power-on: cache lines could contain **UNKNOWN** content %{SIDE:(Note: D-side caches must invalidate SECURE lines even if the cache is only used by NON_SECURE code in order to ensure that unexpected dirty lines tagged SECURE are not naturally evicted)}. Fields:

REGIME unsigned int

Current translation regime.

SIDE unsigned int

0 d-side cache or unified cache, 1 i-side.

ArchMsg.Warning.dcimvac_matches_watchpoint

TAGS **UNPREDICTABLE** DISPLAY Watchpoints matching an AArch32 DCIMVA are implementation defined.

ArchMsg.Warning.decode_fieldmismatch

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with different fields %{FIELD1} and %{FIELD2}. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_fracbitsnegative

TAGS **UNPREDICTABLE** DISPLAY VCVT Instruction is unpredictable with a negative number of fraction bits.

ArchMsg.Warning.decode_initblock

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable in an IT block.

ArchMsg.Warning.decode_initblocknotlast

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable in an IT block when not the last.

ArchMsg.Warning.decode_invalid_condition

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable when the condition is not AL.

ArchMsg.Warning.decode_invalidfieldcombination

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with the values in %{FIELD1} and %{FIELD2}. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_invalidvalue

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with the value in %{FIELD}. Fields:

FIELD enum

Field name.

ArchMsg.Warning.decode_mem_hint_unallocated

TAGS **UNPREDICTABLE** DISPLAY There is no memory hint allocated to this bit pattern.

ArchMsg.Warning.decode_registermatch

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with %{FIELD1} and %{FIELD2} as the same register. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_registermismatch

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with %{FIELD1} and %{FIELD2} as different registers. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_registeroutofrange

TAGS **UNPREDICTABLE** DISPLAY Floating-point instruction is unpredictable indexing beyond the end of the register bank.

ArchMsg.Warning.decode_sbzsbo

TAGS **UNPREDICTABLE** DISPLAY Reserved bits in the instruction are not canonical.

ArchMsg.Warning.decode_transactiontoobig

TAGS **UNPREDICTABLE** DISPLAY Attempt to perform a memory transaction of more than 128 bytes.

ArchMsg.Warning.decode_unpred_other

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable.

ArchMsg.Warning.decode_unpreduseofpc

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable when %{FIELD} is PC. Fields:

FIELD enum

Field name.

ArchMsg.Warning.decode_unpreduseofr13

TAGS UNPREDICTABLE DISPLAY Instruction is unpredictable when %{FIELD} is R13. Fields:

FIELD enum

Field name.

ArchMsg.Warning.decode_unpreduseofr13inreglist

TAGS UNPREDICTABLE DISPLAY Instruction is unpredictable with R13 in register-list.

ArchMsg.Warning.decode_useofd16orhigher

TAGS UNPREDICTABLE DISPLAY Attempt to use register D16 or higher where it is not allowed.

ArchMsg.Warning.decode_writebackandbaseinlist

TAGS UNPREDICTABLE DISPLAY Instruction is unpredictable with write back when the base register is in the transfer list.

ArchMsg.Warning.decode_zeroregistersinlist

TAGS UNPREDICTABLE DISPLAY Instruction is unpredictable when no registers are to be transferred.

ArchMsg.Warning.recursive_branch

An instruction is performing a branch that targets the same instruction. If this was intended then a WFI might be more effective DISPLAY recursive branch detected at PC=%{PC}. PRIMARY KEY PC. Fields:

PC unsigned int

Address of instruction causing the branch.

ArchMsg.Warning.recursive_exception

An instruction has generated an exception that targets the same instruction. DISPLAY recursive %{TYPE} exception detected at PC=%{PC} (CPSR=%{CPSR}). PRIMARY KEY PC. Fields:

CPSR unsigned int

Processor state.

PC unsigned int

Address of instruction causing the exception.

TYPE enum

Type of exception.

ArchMsg.Warning.reentrant_vector_catch

TAGS UNPREDICTABLE DISPLAY Vector catch debug event on Prefetch or Data abort vector %{ADDR}. Fields:

ADDR unsigned int

vector address.

ArchMsg.Warning.reserved_it_state

TAGS UNPREDICTABLE DISPLAY Execution with reserved IT state %{ITSTATE}. Fields:

ITSTATE unsigned int

reserved IT state.

ArchMsg.Warning.tlb_contents_unknown

Translation has been enabled without invalidating the TLB. DISPLAY Translation was enabled in the `%{PAS:(Secure|Non-secure|Root|Realm)} EL%{EL:d}` translation regime (`vmid=%{VMID:x}`) without invalidating `%{INVALIDITY:(|d-side |i-side |)}` entries since power-on: `%{INVALIDITY:(|D|I)}` TLB could contain **UNKNOWN** content. Fields:

EL unsigned int

Exception level of translation regime.

INVALIDITY unsigned int

Which TLB may be invalid (if TLBs are separate).

NON_SECURE bool

Whether the translation regime is non-secure.

PAS unsigned int

Physical address space of translation regime.

VMID unsigned int

Current VMID, if applicable.

ArchMsg.Warning.unknown_DLR_DSPSR

Exiting debug state with unknown DLR or DSPSR.

ArchMsg.Warning.unknown_ELR_SPSR

Returning from debug exception with unknown ELR or SPSR.

ArchMsg.Warning.unpred_FAR_on_LRCPC3_non_atomic_mem_access

TAGS **UNPREDICTABLE** Unpredictable FAR/FSTATUS on LRCPC3 pair instructions resulting in two separate memory accesses. DISPLAY Unpredictable FAR on LRCPC3 non atomic memory access at `%{ADDR1}` with FSTATUS `%{FSTATUS1}` or `%{ADDR2}` with FSTATUS `%{FSTATUS2}`. Fields:

ADDR1 unsigned int

Possible FAR address 1.

ADDR2 unsigned int

Possible FAR address 2.

FSTATUS1 unsigned int

Possible fault status value 1.

FSTATUS2 unsigned int

Possible fault status value 2.

ArchMsg.Warning.unpred_async_tag_check_fail_on_sve_speculative_load_do_update_ffr

Unpredictable set to FALSE of the FFR element associated to SVE speculative load with an async tag check fail TAGS **UNPREDICTABLE**.

ArchMsg.Warning.unpredictable_a32_breakpoint

TAGS **UNPREDICTABLE** DISPLAY Unpredictable breakpoint on a A32 instruction at %{ADDR1} Offset %{OFFSET} BAS %{BAS} for a %{IS_ADDRESS_MISMATCH_BREAKPOINT:(Mismatch|Match)} Breakpoint number %{N:d} Breakpoint address %{ADDR2}. Fields:

ADDR1 unsigned int

Address.

ADDR2 unsigned int

Breakpoint address.

BAS unsigned int

BAS field.

IS_ADDRESS_MISMATCH_BREAKPOINT bool

Is a Mismatch breakpoint, not a Match breakpoint.

N unsigned int

Breakpoint number.

OFFSET unsigned int

Always has a value of zero.

ArchMsg.Warning.unpredictable_banked_register_access

TAGS **UNPREDICTABLE** DISPLAY Access to banked register (R=%{R:d} SYSm=%{SYSM:2x}) is unpredictable. Fields:

R unsigned int

R bit.

SYSM unsigned int

SYSm.

ArchMsg.Warning.unpredictable_ccfail_undef

TAGS **UNPREDICTABLE** DISPLAY An instruction %{OPCODE} at VA=%{VADDR} failed its condition codes check but would otherwise have caused a non-data-dependent trap or undefined exception. Behaviour is implementation-defined choice of exception or nop. Fields:

OPCODE unsigned int

opcode of the instruction.

VADDR unsigned int

Virtual address of the instruction being executed.

ArchMsg.Warning.unpredictable_ici_bits

TAGS **UNPREDICTABLE** DISPLAY ICI bits do not apply or are architecturally unpredictable for instruction.

ArchMsg.Warning.unpredictable_t32_breakpoint

TAGS **UNPREDICTABLE** DISPLAY Unpredictable breakpoint on a T32 instruction at %{ADDR1} Offset %{OFFSET} BAS %{BAS} for a %{IS_ADDRESS_MISMATCH_BREAKPOINT:(Mismatch|Match)} Breakpoint number %{N:d} Breakpoint address %{ADDR2}. Fields:

ADDR1 unsigned int

Address.

ADDR2 unsigned int

Breakpoint address.

BAS unsigned int

BAS field.

IS_ADDRESS_MISMATCH_BREAKPOINT bool

Is a Mismatch breakpoint, not a Match breakpoint.

N unsigned int

Breakpoint number.

OFFSET unsigned int

Indicates which halfword of a 32-bit Thumb instruction the breakpoint is set on. 0 for the first halfword or if the instruction is 16 bits wide, or 1 for the second halfword.

ArchMsg.Warning.unpredictable_t32_vector_catch

TAGS **UNPREDICTABLE** DISPLAY Unpredictable Vector catch exception on a T32 instruction at %{ADDR} Offset %{OFFSET}. Fields:

ADDR unsigned int

Address.

OFFSET unsigned int

Indicates which halfword of a 32-bit Thumb instruction the vector catch exception is set on. 0 for the first halfword or if the instruction is 16 bits wide, or 1 for the second halfword.

ArchMsg.Warning.unpredictable_watchpoint_far

TAGS **UNPREDICTABLE** DISPLAY The value for FAR/EDWAR for hitting this watchpoint may be anywhere in the range %{LOWER_BOUND} to %{UPPER_BOUND}. Fields:

LOWER_BOUND unsigned int

the lowest address accessed by the instruction that triggered the watchpoint.

UPPER_BOUND unsigned int

the highest watchpointed address accessed by that instruction.

ArchMsg.Warning.unpredictable_watchpoint_far_isscmp

DISPLAY The value for FAR/EDWAR for hitting this watchpoint may be anywhere in the range %{LOWER_BOUND} to %{UPPER_BOUND}. Fields:

LOWER_BOUND unsigned int

the lowest address accessed by the instruction that triggered the watchpoint.

UPPER_BOUND unsigned int

the highest watchpointed address accessed by that instruction.

ArchMsg.Warning.warning_AdvSIMDExpandImmUnexpectedZero

TAGS **UNPREDICTABLE** DISPLAY AdvSIMDExpandImm may treat this immediate value as **UNPREDICTABLE**.

ArchMsg.Warning.warning_ConditionalSMC

TAGS **UNPREDICTABLE** DISPLAY SMC instruction has **UNPREDICTABLE** effects when conditional and when combined with traps.

ArchMsg.Warning.warning_TLBID_out_of_range

TLBID out of range in TLBI according to the configuration of NIS, NVIS, NOS and NVOS. Fields:

NIS unsigned int

NIS.

NOS unsigned int

NOS.

NVIS unsigned int

NVIS.

NVOS unsigned int

NVOS.

TLBID unsigned int

The TLBID value used in the TLBI instruction.

ArchMsg.Warning.warning_TTL_or_TG_mismatches_current_translation_granule

TTL or TG mismatches current translation granule in TLBI VA/RVA/IPA/RIPA. Fields:

CURRENT_TG unsigned int

Current TG as indicated by TCR_ELx or VTCR_EL2.

TLBI_TG unsigned int

TG (or TTL[3:2]) used in TLBI instruction.

ArchMsg.Warning.warning_access_crosses_page_boundary

TAGS **UNPREDICTABLE** DISPLAY Access crosses page boundary.

ArchMsg.Warning.warning_access_crosses_page_boundary_has_fault

TAGS **UNPREDICTABLE** DISPLAY Access that crosses page boundary has fault(s).

ArchMsg.Warning.warning_access_crosses_page_boundary_has_fault_on_both_pages

TAGS **UNPREDICTABLE** DISPLAY Access that crosses page boundary has a fault on each page. Fields:

ADDR1 unsigned int

FAR address on first page.

ADDR2 unsigned int

FAR address on second page.

FSTATUS1 unsigned int

Fault status on first page.

FSTATUS2 unsigned int

Fault status on second page.

ArchMsg.Warning.warning_access_crosses_page_boundary_spanning_different_memory

TAGS **UNPREDICTABLE** DISPLAY Access at address %{ADDR} crosses page from MEM_TYPE %{MEMTYPE_PAGE1} to %{MEMTYPE_PAGE2}. Fields:

ADDR unsigned int

address of access crossing page.

MEMTYPE_PAGE1 signed int

First Page's memory type (0-Normal, 1-Non-Normal).

MEMTYPE_PAGE2 signed int

Second Page's memory type (0-Normal, 1-Non-Normal).

ArchMsg.Warning.warning_access_wraps_around_memory

TAGS **UNPREDICTABLE** DISPLAY Access wraps around memory in %{WIDTH:d}bit mode. Fields:

WIDTH unsigned int

Address width.

ArchMsg.Warning.warning_bcr_linking_status

TAGS **UNPREDICTABLE** DISPLAY DBGBCR.{BT2,BT} combination gives unpredictable behaviour.

ArchMsg.Warning.warning_bcr_mask_reserved

TAGS **UNPREDICTABLE** DISPLAY DBGBCR_ELn.MASK is set to a reserved value.

ArchMsg.Warning.warning_bvr_masked_bit_not_zero

TAGS **UNPREDICTABLE** DISPLAY DBGBVRn_EL1.VA contains a bit masked by DBGBCRn_EL1.MASK that is not zero.

ArchMsg.Warning.warning_ccsidr_unimplemented_level

TAGS **UNPREDICTABLE** DISPLAY CCSIDR read while CSSELR %{CSSELR:x} points to an unimplemented cache level. Fields:

CSSELR unsigned int

current effective value of CSSELR.

ArchMsg.Warning.warning_change_to_ns_when_tge_set

TAGS **UNPREDICTABLE** DISPLAY Attempting to change NS to 1 when HCR.TGE = 1.

ArchMsg.Warning.warning_contiguous_bit_check_abort

TAGS **UNPREDICTABLE** DISPLAY An abort occurred whilst attempting to check TLB entry is contiguous at %{ENTRY_ADDR}. Fields:

ENTRY_ADDR unsigned int

address of conflicting TLB entry.

ArchMsg.Warning.warning_contiguous_bit_error

TAGS **UNPREDICTABLE** DISPLAY TLB entry at %{ENTRY_ADDR} was not contiguous with entry at %{CONTIG_ADDR} Contents are %{ENTRY_DATA} but expected %{CONTIG_DATA}. Fields:

CONTIG_ADDR unsigned int

address of the TLB first read, with which this entry is expected to be contiguous.

CONTIG_DATA unsigned int

expected contents based on the entry at CONTIG_ADDR.

ENTRY_ADDR unsigned int

address of conflicting TLB entry.

ENTRY_DATA unsigned int

contents of conflicting TLB entry.

ArchMsg.Warning.warning_contiguous_bit_error_D128

TAGS **UNPREDICTABLE** DISPLAY TLB entry at %{ENTRY_ADDR} was not contiguous with entry at %{CONTIG_ADDR} Contents are %{ENTRY_DATA_HI} %{ENTRY_DATA_LO} but expected %{CONTIG_DATA}. Fields:

CONTIG_ADDR unsigned int

address of the TLB first read, with which this entry is expected to be contiguous.

CONTIG_DATA unsigned int

expected contents based on the entry at CONTIG_ADDR.

ENTRY_ADDR unsigned int

address of conflicting TLB entry.

ENTRY_DATA_HI unsigned int

Higher bit contents of conflicting TLB entry.

ENTRY_DATA_LO unsigned int

Lower bit contents of conflicting TLB entry.

ArchMsg.Warning.warning_cp10_cp11_mismatch

TAGS **UNPREDICTABLE** DISPLAY Unpredictable configuration of CP10 and CP11 controls at EL %{EL:d}. Fields:

EL unsigned int

EL owning the control.

ArchMsg.Warning.warning_cp10_cp11_reserved_value

TAGS **UNPREDICTABLE** DISPLAY Unpredictable configuration of CPACR CP10 (%{CP10}) or CP11 (%{CP11}). Fields:

CP10 unsigned int

CP10 access.

CP11 unsigned int

CP11 access.

ArchMsg.Warning.warning_csselr_level_out_of_range

TAGS **UNPREDICTABLE** DISPLAY CSSELR.Level out of range, written value %{LEVEL} with only %{IMPLEMENTED:d} levels implemented. Fields:

IMPLEMENTED unsigned int

number of cache levels implemented.

LEVEL unsigned int

written value of CSSELR.Level.

TYPE enum

The type of cache selected by CSSELR.InD and TnD.

ArchMsg.Warning.warning_debug_flow_control_bits_not_obeyed

TAGS **UNPREDICTABLE** DISPLAY Flow control bits for %{REGISTER} were not obeyed, causing %{ERROR}. Fields:

ERROR enum

type of error resulting.

REGISTER enum

register under consideration.

ArchMsg.Warning.warning_debug_register_access_during_reset

TAGS **UNPREDICTABLE** DISPLAY Unpredictable %{IS_WRITE:(read|write)} access to debug register offset %{OFFSET} during reset. Fields:

IS_WRITE unsigned int

Write Not Read.

OFFSET unsigned int

Register Offset.

ArchMsg.Warning.warning_decode_cps_inconsistent_fields

TAGS **UNPREDICTABLE** DISPLAY The mode setting fields in the CPS instruction are inconsistent.

ArchMsg.Warning.warning_decode_invalid_state

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable in current exception-level/security state.

ArchMsg.Warning.warning_default_cacheable_mmu_on

TAGS **UNPREDICTABLE** DISPLAY Default cacheable is enabled (HCR.DC = 1) while MMU is enabled.

ArchMsg.Warning.warning_default_cacheable_vmmu_off

TAGS **UNPREDICTABLE** DISPLAY Default cacheable is enabled (HCR.DC = 1) while HCR.VM = 0.

ArchMsg.Warning.warning_deprecated_wvr_bit_2_set

TAGS **UNPREDICTABLE** DISPLAY DBGWVR%{N:d}_EL1=%{WVR:h} has bit 2 set, deprecated in ARMv8. Fields:

N unsigned int

index.

WVR unsigned int

DBGWVR.

ArchMsg.Warning.warning_eret_while_software_step_active_pending

Missing ISB between setting MDSCR_EL1.SS and ERET TAGS UNPREDICTABLE.

ArchMsg.Warning.warning_exclusive_to_non_normal

TAGS UNPREDICTABLE DISPLAY Exclusive Access to Non-Normal Memory %{ADDR}. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_exclusive_to_non_writeback

TAGS UNPREDICTABLE DISPLAY Exclusive Access to Non-Writeback-Cacheable Memory %{ADDR}. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_execute_from_device_memory

TAGS UNPREDICTABLE DISPLAY An attempt was made to execute from Device memory.

ArchMsg.Warning.warning_illegal_cpsr_mode

TAGS UNPREDICTABLE DISPLAY Writing an illegal or unimplement mode (%{NEW_MODE:h}) to CPSR was unpredictable. Fields:

NEW_MODE unsigned int

New value of CPSR.M.

ArchMsg.Warning.warning_illegal_srs_mode

TAGS UNPREDICTABLE DISPLAY Illegal or UNPREDICTABLE mode (%{MODE:h}) used for SRS instruction. Fields:

MODE unsigned int

mode for Banked SP.

ArchMsg.Warning.warning_implementation_defined_read_debug_register_in_SCS

Read Access to Debug registers from the processor is IMPLEMENTATION DEFINED TAGS IMP_DEF. Fields:

PPB_OFFSET unsigned int

debug register offset.

ArchMsg.Warning.warning_implementation_defined_sequential_security_transitions_supported

The behavior of sequential instruction fetches that cross from non-secure to secure memory and contain an SG instruction is CONSTRAINED UNPREDICTABLE TAGS UNPREDICTABLE.

ArchMsg.Warning.warning_implementation_defined_stack_limit_check_supported

It is **IMPLEMENTATION DEFINED** whether stack pointer limit checking is performed for this instruction TAGS IMP_DEF. Fields:

ADDRESS unsigned int

Address of instruction.

OPCODE unsigned int

instruction opcode.

ArchMsg.Warning.warning_implementation_defined_taggednonsharedmemorytype

TAGS IMP_DEF DISPLAY Memory attribute has Tagged, non shared memory type; it is **IMPLEMENTATION DEFINED** whether the memory location is treated as Tagged or non-Tagged.

ArchMsg.Warning.warning_implementation_defined_write_debug_register_in_SCS

Write Access to Debug registers from the processor is **IMPLEMENTATION DEFINED** TAGS IMP_DEF. Fields:

DATA unsigned int

data attempted to be written.

PPB_OFFSET unsigned int

debug register offset.

ArchMsg.Warning.warning_invalid_tcr_granule

TAGS **UNPREDICTABLE** DISPLAY TCR.TG%{TG_ID:(0|1)} bits have been set to a granule size %{{REQUEST:d}K} not implemented - using %{{SUBSTITUTE:d}K}. Fields:

REQUEST signed int

page size requested (or 0 for reserved).

SUBSTITUTE signed int

best guess available page size.

TG_ID bool

bits TG0 or TG1.

ArchMsg.Warning.warning_load_multiple_user_registers_from_user_mode

TAGS **UNPREDICTABLE** DISPLAY An LDM(user registers) instruction executed from user mode.

ArchMsg.Warning.warning_mte_in_mem_attr_not_supported

TAGS **UNPREDICTABLE** DISPLAY Memory attribute %{{ATTR}} has Tagged memory type but Armv8.5-MTE is not implemented. Fields:

ATTR unsigned int

Memory attribute.

ArchMsg.Warning.warning_reserved_breakpoint_state_match

TAGS **UNPREDICTABLE** DISPLAY %{{IS_BREAKPOINT:(Watchpoint|Breakpoint)}} programmed with a reserved combination of HMC, SSC and %{{IS_BREAKPOINT:(PAC|PMC)}}. Fields:

IS_BREAKPOINT unsigned int

Is a breakpoint, not a watchpoint.

ArchMsg.Warning.warning_shareability

TAGS **UNPREDICTABLE** DISPLAY Unpredictable: combination of the 1st and 2nd stages of translation is Normal Inner Non-Cacheable, Outer Non-Cacheable.

ArchMsg.Warning.warning_software_step_set_while_enabled

TAGS **UNPREDICTABLE** DISPLAY MDSCR_EL1.SS set to 1 while software step debug exceptions are enabled.

ArchMsg.Warning.warning_thumb_instruction_wraps_around_memory

TAGS **UNPREDICTABLE** DISPLAY Thumb instruction wraps around memory.

ArchMsg.Warning.warning_ttbr_sbz_bits_are_not_zero

TAGS **UNPREDICTABLE** DISPLAY %{TTBR}[%{MSB:d}:0] should be zero, but are not. Fields:

MSB unsigned int

x-1.

TTBR enum

which TTBR register.

ArchMsg.Warning.warning_unaligned_address_dbgdtrrx_write

TAGS **UNPREDICTABLE** DISPLAY Unaligned address %{ADDR} in X0/R0 when DBGDTRRX written in memory access mode. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unaligned_address_dbgdtrtx_read

TAGS **UNPREDICTABLE** DISPLAY Unaligned address %{ADDR} in X0/R0 when DBGDTRTX read in memory access mode. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unaligned_to_device

TAGS **UNPREDICTABLE** DISPLAY Unaligned Access to Device Memory %{ADDR}. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unaligned_to_strongly_ordered

TAGS **UNPREDICTABLE** DISPLAY Unaligned Access to Strongly Ordered Memory %{ADDR}. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unknown_sau_rnr

TAGS UNPREDICTABLE DISPLAY SAU_RNR was set to an unsupported value.

ArchMsg.Warning.warning_unpredictable_AIRCR_PRIS_and_BFHFNMINs

TAGS UNPREDICTABLE DISPLAY The effect of setting both AIRCR.BFHFNMINs and AIRCR.PRIS to 1 is UNPREDICTABLE.

ArchMsg.Warning.warning_unpredictable_AIRCR_VECTCLRACTIVE_when_not_in_debug

TAGS UNPREDICTABLE DISPLAY The effect of writing a 1 to AIRCR.VECTCLRACTIVE if the processor is not halted in Debug state is UNPREDICTABLE.

ArchMsg.Warning.warning_unpredictable_AIRCR_VECTRESET_and_SYSRESETREQ

TAGS UNPREDICTABLE DISPLAY When the processor is halted in Debug state, if a write to the register writes a 1 to both VECTRESET and SYSRESETREQ, the behavior is UNPREDICTABLE.

ArchMsg.Warning.warning_unpredictable_AIRCR_VECTRESET_when_not_in_debug

TAGS UNPREDICTABLE DISPLAY The effect of writing a 1 to AIRCR.VECTRESET if the processor is not halted in Debug state is UNPREDICTABLE.

ArchMsg.Warning.warning_unpredictable_AIRCR_incorrect_VKEY

TAGS UNPREDICTABLE DISPLAY The value 0x05FA must be written to AIRCR.VKEY, otherwise the register write is UNPREDICTABLE.

ArchMsg.Warning.warning_unpredictable_EXC_RETURN_Reserved_Bit

EXC_RETURN[23:7] are reserved with the special condition that all bits should be written as one. Values other than all 1s are UNPREDICTABLE. TAGS UNPREDICTABLE. Fields:

EXC_RETURN unsigned int

Special PC value.

SBO_MASK unsigned int

Which bits Should Be One to not UNPRED.

ArchMsg.Warning.warning_unpredictable_PMEVTYPEPn_ELO_VS_reserved_value

TAGS UNPREDICTABLE DISPLAY Unpredictable reserved value of PMEVTYPEn<n>_ELO.VS.

ArchMsg.Warning.warning_unpredictable_change_to_DEMCR_MON_STEP_at_insufficient_priority

TAGS UNPREDICTABLE DISPLAY The effect of changing DEMCR.MON_STEP at an execution priority that is lower than the priority of the DebugMonitor exception is **CONSTRAINED** UNPREDICTABLE.

ArchMsg.Warning.warning_unpredictable_change_to_priority_of_active_exception

TAGS UNPREDICTABLE DISPLAY Changing the priority of an active exception is UNPREDICTABLE.

ArchMsg.Warning.warning_unpredictable_clear_lspact

TAGS UNPREDICTABLE DISPLAY UNPREDICTABLE state on context switch. Fields:

DRVEXC_TAKEN unsigned int

Derived exception taken, not pended.

REACHED_LAST_STORE unsigned int

Only the last faulted.

ArchMsg.Warning.warning_unpredictable_dbg_exit_unaligned_dlr

TAGS **UNPREDICTABLE** DISPLAY Debug exit to AArch32 with DLR[0] set to 1. PC[0] can be set to 0 or to DLR[0].

ArchMsg.Warning.warning_unpredictable_exception_catch

TAGS **UNPREDICTABLE** DISPLAY Generation of an exception catch event is unpredictable.

ArchMsg.Warning.warning_unpredictable_exception_return_inconsistent_state

TAGS **UNPREDICTABLE** DISPLAY state on exception return is unpredictable.

ArchMsg.Warning.warning_unpredictable_exception_return_instruction

TAGS **UNPREDICTABLE** DISPLAY exception return instruction is unpredictable.

ArchMsg.Warning.warning_unpredictable_in_debug_state

TAGS **UNPREDICTABLE** DISPLAY Instruction is **UNPREDICTABLE** when executed in debug state.

ArchMsg.Warning.warning_unpredictable_mair_encoding

TAGS **UNPREDICTABLE** DISPLAY Memory attribute %{ATTR} has invalid MAIR encoding; this may lead to unpredictable results. Fields:

ATTR unsigned int

Memory attribute.

ArchMsg.Warning.warning_unpredictable_pmu_counter_access

TAGS **UNPREDICTABLE** DISPLAY Access to PMU counters from non-secure EL0 or EL1 is **UNPREDICTABLE** with MDCR_EL2.HPMN set to 0.

ArchMsg.Warning.warning_unpredictable_prioritization_breakpoint_match_vector_catch

TAGS **UNPREDICTABLE** DISPLAY Two events with the same priority occurred at the same instruction: Address Matching Vector Catch debug event, and Breakpoint debug event. It is constrained unpredictable which is taken.

ArchMsg.Warning.warning_unpredictable_stack_selection

TAGS **UNPREDICTABLE** DISPLAY **UNPREDICTABLE** state on context switch.

ArchMsg.Warning.warning_unpredictable_tsize_out_of_range

TAGS **UNPREDICTABLE** DISPLAY %{TCR}.T%{TTBR:d}SZ (%{TSIZE:d} bits) is out of range. Must be between 25 and %{TSIZE_MAX:d}. Fields:

TCR enum

Which TCR.

TSIZE signed int

Value of TSize.

TSIZE_MAX signed int

The maximum allowed value.

TTBR unsigned int

Which TTBR region.

ArchMsg.Warning.warning_unpredictable_unaligned_pc_as_base_register

TAGS **UNPREDICTABLE** DISPLAY The PC value must be word-aligned, otherwise the behavior of the instruction is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_unaligned_pop_stack

TAGS **UNPREDICTABLE** DISPLAY **CONSTRAINED UNPREDICTABLE** unaligned stack frame %{FRAME_ADDR:x} in exception-return. Fields:

FRAME_ADDR unsigned int

unaligned stack frame address.

ArchMsg.Warning.warning_unpredictable_unaligned_to_non_writeback

TAGS **UNPREDICTABLE** DISPLAY Unaligned Access to Non-Writeback-Cacheable Memory.

ArchMsg.Warning.warning_unpredictable_vmsa_memattrib

TAGS **UNPREDICTABLE** DISPLAY Unpredictable vmsa memory attribute with texcb value programmed to %{TEXCB} when tex remap is %{TEX_REMAP}. Fields:

TEXCB unsigned int

texcb value.

TEX_REMAP bool

use tex remap.

ArchMsg.Warning.warning_unpredictable_vtcr_t0sz_sl0

TAGS **UNPREDICTABLE** DISPLAY The combination of VTCR.T0SZ=%{T0SZ:d} and VTCR.SL0=%{SL0:d} is unpredictable. Fields:

SL0 signed int

VTCR.SL0, starting level for stage 2 translation table walks.

T0SZ signed int

VTCR.T0SZ, required input address range.

ArchMsg.Warning.warning_unpredictable_write_DHCSR

TAGS **UNPREDICTABLE** DISPLAY **UNPREDICTABLE** set of bit-changes in DHCSR. Fields:

NEW unsigned int

data attempted to be written.

OLD unsigned int

previous value.

ArchMsg.Warning.warning_unpredictable_write_SHCSR

TAGS **UNPREDICTABLE** DISPLAY **UNPREDICTABLE** setting a pending bit to 1 for an exception with priority >= execution priority.

ArchMsg.Warning.warning_unsupported_access_to_memory_mapped_register

TAGS **UNPREDICTABLE** DISPLAY Access to memory mapped register is unsupported.

ArchMsg.Warning.warning_user_jmcr_access

TAGS UNPREDICTABLE DISPLAY User %{WRITE:(write|read)} access to JMCR is unpredictable.
Fields:

WRITE unsigned int
access is write.

ArchMsg.Warning.warning_wcr_mask_and_bas

TAGS UNPREDICTABLE DISPLAY DBGWCR_EL1.MASK is non-zero and BAS is not 0xff.

ArchMsg.Warning.warning_wcr_mask_reserved

TAGS UNPREDICTABLE DISPLAY DBGWCR_ELn.MASK is set to a reserved value.

ArchMsg.Warning.warning_wcr_non_configuous_bas

TAGS UNPREDICTABLE DISPLAY DBGWCR_EL1.BAS has non-contiguous set of ones.

ArchMsg.Warning.warning_wvr_masked_bit_not_zero

TAGS UNPREDICTABLE DISPLAY DBGWVRn_EL1.VA contains a bit masked by
DBGWCRn_EL1.MASK that is not zero.

ArchMsg.Why.why_branch_target_exception

A branch target exception has occurred. DISPLAY branch target exception: target pc=%{PC},
target opcode=%{OPCODE}. PRIMARY KEY PC. Fields:

BTYPE unsigned int
pstate.BTYPE at time branch taken.

OPCODE unsigned int
opcode at branch target.

PC unsigned int
Virtual Address of branch target.

ArchMsg.Why.why_illegal_state

An illegal mode change has occurred. DISPLAY illegal mode change: %{MESSAGE}. Fields:

MESSAGE string
The message.

BRANCH_MISPREDICT

Simulating branch mispredict. Fields:

PC unsigned int
Origin address (or 0 if unavailable).

BRA_DIR

Direct branches, to immediate address. Fields:

CORE_NUM unsigned int
Core number in a multi processor.

INST_COUNT unsigned int
The core's instruction counter, starting at 1 for the first instruction.

ISSET enum

The instruction set of the branch instruction.

IS_COND bool

Indicates if this is a conditional branch.

IS_HINTED bool

Whether the conditional instruction is hinted.

IS_ISB bool

Whether the instruction is ISB.

IS_LINK bool

Indicates if this is a branch with link.

PC unsigned int

The address of the branch instruction.

TARGET_ISET enum

The instruction set after the branch.

TARGET_PC unsigned int

The address the instruction branches to.

BRA_INDIR

Indirect branches, perhaps to a register. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISSET enum

The instruction set of the branch instruction.

IS_COND bool

Indicates if this is a conditional branch.

IS_HINTED bool

Whether the conditional instruction is hinted.

IS_ISB bool

Whether the instruction is ISB.

IS_LINK bool

Indicates if this is a branch with link.

PC unsigned int

The address of the branch instruction.

TARGET_ISET enum

The instruction set after the branch.

TARGET_PC unsigned int

The address the instruction branches to.

BRA_INDIR_CCFAIL

Indirect branches fail, perhaps to a register.

BRA_RET_CCFAIL

Branch return fail, perhaps to a register.

CACHE_MAINTENANCE_OP

Cache Maintenance Operation. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

DATA unsigned int

Specified MVA or Set/way.

FUNCTION enum

Clean / Invalidate.

SCOPE enum

Affected region.

SIDE enum

Inst / Data.

CCFAIL

Conditional instruction condition check fail. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CCFAIL_UNC

Unconditional instruction condition check fail. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CCPASS

Conditional instruction condition check pass. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CCPASS_UNC

Unconditional instruction condition check pass. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CODE_CACHE_MAINT

Code cache maintenance. Fields:

END_INCL unsigned int

Inclusive end address for VA or PA requests.

NS bool

Non-secure world for PA.

START unsigned int

Start address for VA or PA requests.

TYPE enum

Request type.

COMPILE_BLOCK_END

Last instruction of basic block translated. Fields:

VADDR unsigned int

Address of next instruction after this basic block.

COMPILE_BLOCK_START

First instruction of basic block translated. Fields:

VADDR unsigned int

Address of first instruction in basic block.

COMPILE_INST

ARM instruction compiled. Fields:

DISASS string

Disassembly of the instruction.

ISSET enum

The instruction set of this instruction.

ITSTATE unsigned int

The ITSTATE current for the instruction.

OPCODE unsigned int

The opcode of the instruction.

PC unsigned int

The address of the instruction.

SIZE unsigned int

The size of the instruction in bytes.

CONTEXTIDR

Context ID Register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

NS bool

Secure or nonsecure banked register is accessed.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The new value written.

CONTEXT_SYNC

Called for every context synchronization event. Fields:

ARCH_CSE bool

Architectural Context Synchronization Event.

PC unsigned int

The address of the synchronization instruction.

SYSREG_SYNC bool

System register synchronization event.

CORE_ENDIAN

Core BE8 Big-Endian state changed. Fields:

BE8 bool

Core BE8 Big-Endian state.

CORE_INFO

Static processor attributes. Only triggered by a call to DumpState(). Fields:

ARCH_PROFILE enum

The architecture profile of the core.

CLUSTER_ID unsigned int

The cluster ID of this processor.

CORE_NUM unsigned int

The number of this core in an MP processor.

DETAILED_CORE_FEATURES enum

Detailed core features as a bitfield. Enum values are bitmasks to extract the info from the field.

FPU_VERSION enum

The VFP version implemented by the core.

MEM_ARCH enum

The memory architecture of the core.

MIDR_EL1 unsigned int

MIDR_EL1 value.

NUM_CORES unsigned int

The number of cores in this MP processor.

QUANTUM_SIZE unsigned int

The default quantum size of the core (in ticks).

SECURITY_FEATURES bool

Does the core have security features?.

CORE_LOADS

Processor load accesses. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ACQREL enum

Is this an acquire/release.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DATA unsigned int

The data read or written.

ELEMENT_SIZE unsigned int

Width of each element.

ENDIANNESS enum

Byte ordering of this access.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

LOCK enum

Normal, exclusive or locked access.

MECID unsigned int

Memory Encryption Context ID, if applicable.

MECID2 unsigned int

Memory Encryption Context ID of the second page, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PAGE_SIZE unsigned int

Page size as $\log_2(\text{size in bytes})$.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK, 2=Exclusive Failed, 3=Unsupported.

SIZE unsigned int

Width of the access in bytes. Only required if DATA is not traced.

TRANS bool

Is this a translated access.

VADDR unsigned int

The virtual address of the access.

CORE_REGS

Changes of the core registers R0 to R14. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number, 0 to 14.

MODE enum

Bank of the register accessed.

OLD_VALUE unsigned int

The old value overwritten.

PHYS_ID enum

The physical register accessed.

VALUE unsigned int

The new value written to the register.

CORE_REGS64

Changes of the core registers X0..X30, SP_ELn. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ID enum

The register number, 0-30 for X0-X30, >=32 for SP_ELn.

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

CORE_REGS64_READ

General purpose AArch64 register has been read X0..X30, SP_ELn. Fields:

ID enum

The register number, 0-30 for X0-X30, >=32 for SP_ELn.

VALUE unsigned int

The value read from the register.

CORE_STALLED_TRANSITION

Detect when a core is prevented from progressing PChannel states. Fields:

CURRENT_STATE enum

Current power mode of core.

REASON string

The reason a transition was stalled, and what must be done to unblock it.

TARGET_STATE enum

Desired power mode of core after transition.

CORE_STORES

Processor store accesses. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ACQREL enum

Is this an acquire/release.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DATA unsigned int

The data read or written.

ELEMENT_SIZE unsigned int

Width of each element.

ENDIANNESS enum

Byte ordering of this access.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

LOCK enum

Normal, exclusive or locked access.

MECID unsigned int

Memory Encryption Context ID, if applicable.

MECID2 unsigned int

Memory Encryption Context ID of the second page, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PAGE_SIZE unsigned int

Page size as $\log_2(\text{size in bytes})$.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK, 2=Exclusive Failed, 3=Unsupported.

SIZE unsigned int

Width of the access in bytes. Only required if DATA is not traced.

TRANS bool

Is this a translated access.

VADDR unsigned int

The virtual address of the access.

CP14_READ

System Coprocessor register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP14 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc1 unsigned int

opcode 1.

opc2 unsigned int

opcode 2.

CP14_WRITE

System Coprocessor register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP14 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc1 unsigned int

opcode 1.

opc2 unsigned int
opcode 2.

CP15_READ

System Coprocessor register read. Fields:

CORE_NUM unsigned int
Core number in a multi processor.

CRm unsigned int
CRm.

CRn unsigned int
CRn.

NS bool
Secure or nonsecure banked register is accessed.

REG_NAME string
Name of the CP15 register accessed.

SECURITY_STATE enum
The current security state.

UNDEF bool
The register accessed is undefined.

VALUE unsigned int
The value read.

opc1 unsigned int
opcode 1.

opc2 unsigned int
opcode 2.

CP15_READ64

System Coprocessor register read. Fields:

CORE_NUM unsigned int
Core number in a multi processor.

CRm unsigned int
CRm.

NS bool
Secure or nonsecure banked register is accessed.

REG_NAME string
Name of the CP15 register accessed.

SECURITY_STATE enum
The current security state.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc unsigned int

opcode 1.

CP15_WRITE

System Control Coprocessor register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP15 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc1 unsigned int

opcode 1.

opc2 unsigned int

opcode 2.

CP15_WRITE64

System Coprocessor register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP15 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc unsigned int

opcode 1.

CPSR

CPSR change. (In AArch64 PSTATE represented in SPSR format). Fields:

CORE_NUM unsigned int

Core number in a multi processor.

OLD_VALUE unsigned int

The old CPSR value.

UNKNOWN unsigned int

Bits within the register that have unknown value.

VALUE unsigned int

The new CPSR value.

CRYPTO_SPEC

Every crypto instruction speculatively executed.

CorePowerStateChange

Core power-up state has changed. DISPLAY Core powered %{PU:(up|down)}. Fields:

PU bool

Whether the core powered up.

DATA_CACHE_SET_ALLOC_TAGS

Set allocation tags from DC GVA instruction. Fields:

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate

when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

NSDESC unsigned int

The security state of the access.

PADDR unsigned int

The physical address of the access.

PAS enum

The physical address space of the page.

RESPONSE enum

0=Aborted, 1=OK.

SIZE unsigned int

The size of memory in bytes.

TAG unsigned int

The value of the tag being set.

TAGGING_TYPE enum

Type of Memory tagging (Physical or Virtual).

TAG_VADDR unsigned int

The virtual address of the Virtual Tag if Virtual tagging is used; otherwise 0.

VADDR unsigned int

Virtual address used for setting allocation tags.

DATA_CACHE_ZERO

Zero and invalidate cache line from DC ZVA instruction. Fields:

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

NSDESC unsigned int

The security state of the access.

PADDR unsigned int

The physical address of the access.

PADDR_TAG unsigned int

If tag checked, the physical address tag for the access.

PAS enum

The physical address space of the page.

RESPONSE enum

0=Aborted, 1=OK.

SIZE unsigned int

The size of zero'd memory in bytes.

TAG_CHECKED bool

Is this access tag checked.

VADDR unsigned int

Virtual address of the zero'd cache line.

DEBUG_EVENT

Hardware debug support event. Fields:

EVENT enum

Description of event.

VALUE unsigned int

data value.

DEBUG_TARGET_ISET

Hardware debug target ISA. Fields:

TARGET_ISET enum

The target instruction set of the debug state.

DMI_ALLOCATE

DMI allocated. Fields:

ACCESS_DIRECTION enum

READ / WRITE.

PADDR unsigned int

The physical address of the DMI.

PTR unsigned int

DMI range pointer.

SIDE enum

DSIDE / ISIDE.

SIZE unsigned int

The size of DMI.

VADDR unsigned int

The virtual address of the DMI..

DMI_HIT

DMI hit, access used DMI (only fire for small DMI regions i.e. less than a full page). Fields:

ACCESS_DIRECTION enum

READ / WRITE.

PADDR unsigned int

The physical address of the DMI.

SIDE enum

DSIDE / ISIDE.

SIZE unsigned int

The size of the DMI.

VADDR unsigned int

The virtual address of the DMI..

DMI_REVOKE

DMI revoked. Fields:

ACCESS_DIRECTION enum

READ / WRITE.

PADDR unsigned int

The physical address of the DMI.

SIDE enum

DSIDE / ISIDE.

SIZE unsigned int

The size of DMI.

VADDR unsigned int

The virtual address of the DMI..

END_COMPILE

Compilation end. Fields:

END_OF_PAGE_COUNT unsigned int

Number of basic blocks exited due to page boundary since START_COMPILE.

FETCHFAIL_COUNT unsigned int

Number of basic blocks exited due to ifetch failure START_COMPILE.

INST_COUNT unsigned int

Number of instructions compiled since START_COMPILE.

NONSEQ_COUNT unsigned int

Number of basic blocks exited due to non-sequential instructions since START_COMPILE.

PAGE_STRADDLE_COUNT unsigned int

Number of basic blocks exited due to unaligned instructions crossing page since START_COMPILE.

ESB

Error Synchronization Barrier hit. Fields:

ASYNC bool

ESB is async.

IMPLICIT bool

ESB is implicit.

EXCEPTION

Exception taken. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ESR unsigned int

The value assigned to ESR when exception is taken.

ESR_EC unsigned int

The Value assigned to ESR.EC when exception is taken.

IS_PHYSICAL bool

Physical or Virtual exception.

IS_TRAP bool

Exception coming from a trap.

LR unsigned int

The value assigned to the link register.

PC unsigned int

The location where the exception occurred.

PREFERRED_RETURN unsigned int

The preferred return address for the exception.

TAKEN_LOCALLY bool

Whether exception is taken locally(same exception level).

TARGET_ISET enum

The instruction set of the exception handler code.

TARGET_PC unsigned int

The address the exception branches to.

VECTOR enum

The exception vector.

VECTOR_OFFSET unsigned int

The offset of an exception vector from the base address.

EXCEPTION_END

Every exception completed.

EXCEPTION_INFO

Trace human-readable reason of the current exception for data/instruction aborts, floating point exceptions and unknown exceptions. Fields:

DESCRIPTION string

Reason for current exception.

ESR_EC unsigned int

The value of ESR.EC when exception is taken.

EXCEPTION_RAISE

Every exception raised.

EXCEPTION_RETURN

Branches on leaving exception. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISET enum

The instruction set of the branch instruction.

IS_COND bool

Indicates if this is a conditional branch.

IS_HINTED bool

Whether the conditional instruction is hinted.

IS_ISB bool

Whether the instruction is ISB.

IS_LINK bool

Indicates if this is a branch with link.

PC unsigned int

The address of the branch instruction.

TARGET_ISET enum

The instruction set after the branch.

TARGET_PC unsigned int

The address the instruction branches to.

EXCEPTION_RETURN_PREBRANCH

Exception return event that must occur before PC and processor state have updated.

EXCEPTION_START

Every exception started.

FIQ_TAKEN

FIQ taken exception.

FP_EXCEPTIONS_SIGNALLED

Trace which FP exceptions were signalled as a part of the FP operations performed by an instruction. Fields:

FP_EXCEPTIONS_SIGNALLED_MASK unsigned int

Indicates which FP exceptions were signalled by an instruction. This uses the same encoding as the cumulative floating-point exception bitfields in FPSR.

FP_STATE

Floating point state. Fields:

FPSR unsigned int

FPSR register value.

MASK unsigned int

The mask indicating updated cumulative bits.

FREQ_CHANGED

Core frequency changed. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The instruction count of this core.

NEW_FREQ signed int

The new frequency of this core (expressed in Hz).

OLD_FREQ signed int

The old frequency of this core (expressed in Hz).

HLT

HLT instruction occurred. Fields:

IMM16 unsigned int

The 16-bit immediate value encoded in HLT instruction.

INST

Every instruction executed. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DEBUG_STATE bool

The instruction is executed in debug state.

DISASS string

Disassembly of instruction.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISSET **enum**

The current instruction set.

ITSTATE **unsigned int**

The current ITSTATE.

LOCAL_TIME **unsigned int**

The core's local time, relative to the current quantum.

MODE **enum**

The mode the core is in.

NS **unsigned int**

The core's non-secure bit.

NSDESC **unsigned int**

The physical address non-secure bit.

NSDESC2 **unsigned int**

The second page physical address non-secure bit.

OPCODE **unsigned int**

The opcode of the instruction.

PADDR **unsigned int**

The physical address of the instruction.

PADDR2 **unsigned int**

If different from PADDR, the physical address of the second page of the instruction.

PAS **enum**

The physical address space of the page.

PAS2 **enum**

The physical address space of the page.

PC **unsigned int**

The address of the instruction.

SECURITY_STATE **enum**

The core's security state.

SIZE **unsigned int**

The size of the instruction in bytes.

INST_END

Every instruction completed.

INST_START

Every instruction started. Fields:

DEBUG_STATE **bool**

The instruction is executed in debug state.

ISSET enum

The current instruction set.

MODE enum

The mode the core is in.

NS enum

The current Secure State.

PC unsigned int

The address of the conditional instruction.

SECURITY_STATE enum

The current Security State.

INST_STRADDLE

Instruction straddles boundary. Fields:

DEBUG_STATE bool

The instruction is executed in debug state.

ISSET enum

The current instruction set.

MODE enum

The mode the core is in.

NS enum

The current Secure State.

PC unsigned int

The address of the conditional instruction.

SECURITY_STATE enum

The current Security State.

IRQ_TAKEN

IRQ taken exception.

ISB

ISB instruction occurred. Fields:

PC unsigned int

The address of the ISB instruction.

LOAD_MULTIPLE_REGS_LIST

List of destination registers (not source) of a multiple registers load instruction. Fields:

REG string

Register to add into the list.

LOCAL_MONITOR

Local monitor activity. Fields:

PADDR unsigned int

Local Monitor Address.

State enum

State of the monitor (Open/Exclusive).

MAPPING_CREATE

A memory mapping has been created. This is an implementation detail of the model and so may change without notice. Fields:

ASID unsigned int

The ASID used during translation, or 0 if no ASID was used.

IS_VALID bool

Whether the mapping is valid or not.

PADDR unsigned int

Base physical address of region, or 0 if not known.

SIDE enum

I-side or D-side.

SIZE unsigned int

Size of the page, or 0 if not known.

VADDR unsigned int

Base virtual address of region.

VMID unsigned int

The VMID used during translation, or 0 if no VMID was used.

MAPPING_DESTROY

A memory mapping has been destroyed. This is an implementation detail of the model and so may change without notice. Fields:

ASID unsigned int

The ASID used during translation.

CAUSE enum

The cause of invalidation.

PADDR unsigned int

Base physical address of page.

SIDE enum

I-side or D-side.

SIZE unsigned int

Size of the page.

VADDR unsigned int

Base virtual address of page.

VMID unsigned int

The VMID used during translation.

MEMMAP_DEBUG_READ

Memory mapped reads to the debug registers. Fields:

ADDR unsigned int

address.

CORE_NUM unsigned int

Core number in a multi processor.

EXT bool

Whether access is from an external device (such as the DAP).

NS enum

Secure state of the access.

REG_NAME string

Name of the debug register accessed.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

MEMMAP_DEBUG_WRITE

Memory mapped writes to the debug registers. Fields:

ADDR unsigned int

address.

CORE_NUM unsigned int

Core number in a multi processor.

EXT bool

Whether access is from an external device (such as the DAP).

NS enum

Secure state of the access.

REG_NAME string

Name of the debug register accessed.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The new value written.

MMU_D128_TTB_READ

This event is triggered by reads caused by a D128 translation table walk. Fields:

ABORTED bool

Was the walk successful.

DATA_HIGH unsigned int

The data upper bits read.

DATA_LOW unsigned int

The data lower bits read.

IPA unsigned int

For stage 1, the IPA of the read.

LEVEL unsigned int

Translation table level.

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

OUTADDR unsigned int

The output physical address.

PADDR unsigned int

The physical address of the read.

PASmem enum

Physical Address Space of the accessed memory.

PASreq enum

Physical Address Space of the requested lookup.

REGIME_EL enum

Entry matches in this translation regime.

REQUESTER enum

The requester of this table walk.

SIDE enum

Inst / Data.

STAGE unsigned int

Translation stage.

MMU_D128_TTB_WRITE

This event is triggered by writes caused by a D128 translation table walk. Fields:

ABORTED bool

Was the walk successful.

DATA_HIGH unsigned int

The data upper bits written.

DATA_LOW unsigned int

The data lower bits written.

IPA unsigned int

For stage 1, the IPA of the write.

IS_HACDBS_ACCESS bool

If the access was due to HACDBS_CLEAN.

LEVEL unsigned int

Translation table level.

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

OUTADDR unsigned int

The output physical address.

PADDR unsigned int

The physical address of the write.

PASmem enum

Physical Address Space of the accessed memory.

PASreq enum

Physical Address Space of the requested lookup.

REGIME_EL enum

Entry matches in this translation regime.

SIDE enum

Inst / Data.

STAGE unsigned int

Translation stage.

MMU_TRANS

Address translation information. Fields:

ASID unsigned int

Address space identifier.

CORE_NUM unsigned int

Core number in a multi processor.

Hyp bool

Entry matches in Hyp state only.

INNERCACHE_RA bool

Is the inner cache allocate on read.

INNERCACHE_TRANSIENT bool

Is the inner write-through transient.

INNERCACHE_TYPE enum

Inner Caching scheme (NC/MB/WA).

INNERCACHE_WA bool

Is the inner cache allocate on write.

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MEMTYPE enum

Memory type.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC enum

Is secure side supposed to access secure or nonsecure memory.

OUTERCACHE_RA bool

Is the outer cache allocate on read.

OUTERCACHE_TRANSIENT bool

Is the outer write-through transient.

OUTERCACHE_TYPE enum

Outer Caching scheme (NC/MB/WA).

OUTERCACHE_WA bool

Is the outer cache allocate on write.

PADDR unsigned int

Physical address of the access.

PAGESIZE unsigned int

Page size as log2(size).

PAS enum

PAS supposed to be accessed is secure or nonsecure memory or Root/Realm if RME is implemented.

SH enum

Shareability.

SIDE enum

Inst / Data.

STAGE1_PERM unsigned int

Stage 1 permissions, [5:3] Privileged access (XWR) [2:0] User access (XWR).

STAGE2_PERM unsigned int

Stage 2 permission mask [5:3] Privileged access (XWR) [2:0] User access (XWR).

VADDR unsigned int

Virtual address of the access.

VMID unsigned int

Virtual machine identifier.

XS bool

XS Attribute.

nG enum

Flag indicating whether ASID will be matched.

MMU_TTB_READ

This event is triggered by reads caused by a translation table walk. Fields:

ABORTED bool

Was the walk successful.

DATA unsigned int

The data read.

DESC_KIND enum

The encoding scheme of the descriptors used for this walk.

IPA unsigned int

For stage 1, the IPA of the read.

LEVEL unsigned int

Translation table level.

LPAAE bool

Is this for an LPAAE translation (*DEPRECATED*, use DESC_KIND).

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSmem bool

Non secure memory access.

NSreq bool

Non secure request.

OUTADDR unsigned int

The output physical address.

PADDR unsigned int

The physical address of the read.

PASmem enum

Physical Address Space of the accessed memory.

PASreq enum

Physical Address Space of the requested lookup.

REGIME_EL enum

Entry matches in this translation regime.

REQUESTER enum

The requester of this table walk.

SIDE enum

Inst / Data.

STAGE unsigned int

Translation stage.

MMU_TTB_WRITE

This event is triggered by writes caused by a translation table walk. Fields:

ABORTED bool

Was the walk successful.

DATA unsigned int

The data written.

DESC_KIND enum

The encoding scheme of the descriptors used for this walk.

IPA unsigned int

For stage 1, the IPA of the write.

IS_HACDBS_ACCESS bool

If the access was due to HACDBS_CLEAN.

LEVEL unsigned int

Translation table level.

LPAE bool

Is this for an LPAE translation (*DEPRECATED*, use DESC_KIND).

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSmem bool

Non secure memory access.

NSreq bool

Non secure request.

OUTADDR unsigned int

The output physical address.

PADDR unsigned int

The physical address of the write.

PASmem enum

Physical Address Space of the accessed memory.

PASreq enum

Physical Address Space of the requested lookup.

REGIME_EL enum

Entry matches in this translation regime.

SIDE enum

Inst / Data.

STAGE unsigned int

Translation stage.

MODE_CHANGE

Mode change. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

MODE enum

The new mode.

NON_SECURE enum

The core's new non-secure bit.

OLD_MODE enum

The old mode.

SECURITY_STATE enum

The new security state.

MPU_TRANS

Address translation information. Fields:

ACCESS_STATUS enum

Access status(0xAA/0xBB/0xDD/0xEE=overlap/bg hit/default/bg fault).

CORE_NUM unsigned int

Core number in a multi processor.

EXEC_PERM enum

Execution Permission.

INNERCACHE_RA bool

Is the inner cache allocate on read.

INNERCACHE_TYPE enum

Inner Caching scheme (NC/MB/WA).

INNERCACHE_WA bool

Is the inner cache allocate on write.

MEMTYPE enum

Memory type.

NSDESC enum

Is non secure access.

OUTERCACHE_RA bool

Is the outer cache allocate on read.

OUTERCACHE_TYPE enum

Outer Caching scheme (NC/MB/WA).

OUTERCACHE_WA bool

Is the outer cache allocate on write.

PADDR unsigned int

Address of the access.

PAGESIZE unsigned int

Page size as $\log_2(\text{size})$.

PAS enum

PAS accessed is secure or nonsecure memory.

READ_PERM enum

Read Permission.

REG_NUM unsigned int

MPU region-number 0-31.

SH enum

Shareability.

SIDE enum

Inst / Data.

STAGE unsigned int

The translation stage.

WRITE_PERM enum

Write Permission.

NOT_TAG_CHECKED_INST

MTE Tag not checked for current instruction.

PERIODIC

Called for every quantum. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The instruction count of this CPU.

PC unsigned int

The address of the next instruction to be executed on this CPU.

PMU_COUNTER_OVERFLOW

PMU counter overflow. Fields:

COUNTER_GROUP enum

Counter group.

COUNTER_RANGE enum

Counter range.

EVENT_ID unsigned int

Event Identifier, field is valid only for Event counter group.

INDEX unsigned int

Counter index (as selected by PMSELR), field is valid only for Event/Cycle counter groups.

INTERRUPT bool

Is interrupt enabled on overflow for this counter.

POINTER_AUTH

Called when execution hits a pointer authentication check. Fields:

AUTH_STATE bool

Authentication result.

KEY_USED enum

Pointer authentication key.

POTINDEX_UPDATE

Value of POT index for current pc changed. Fields:

NEW_POTINDEX unsigned int

New POTIndex.

OLD_POTINDEX unsigned int

Old POTIndex.

PREFETCH_MEMORY64

Prefetch from PRFM or PRFUM instructions. Fields:

PRFOP enum

Prefetch hint.

VADDR unsigned int

Virtual address of the location that should be prefetched.

PRELOAD64_REQUEST

Preload requests from PRFM or PRFUM instructions. Fields:

PRFOP enum

Prefetch hint.

VADDR unsigned int

Virtual address of the location that should be prefetched.

PRELOAD_DATA

Data preload from PLD instruction. Fields:

VADDR unsigned int

Virtual address of the data that should be preloaded.

PRELOAD_INST

Instruction preload from PLI instruction. Fields:

VADDR unsigned int

Virtual address of the instruction that should be preloaded.

PRE_CORE_LOAD

Trace just before a core load. Fields:

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

PRE_CORE_STORE

Trace just before a core store. Fields:

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

PRE_TTB_READ

This event is triggered before reads caused by a translation table walk. Fields:

ADDR unsigned int

The physical address of the read.

PSTATE

PSTATE fields update. Fields:

ALLINT unsigned int

All interrupt mask bit.

BTYP unsigned int

Branch target identification bit.

CORE_NUM unsigned int

Core number in a multi processor.

DAIF unsigned int

Exception mask bits.

DIT unsigned int

Data Independent Timing bit.

EL unsigned int

Current Exception level field.

EXLOCK unsigned int

Exception return state lock bit.

IL unsigned int

Illegal Execution state bit.

NZCV unsigned int

Condition flags.

PACM unsigned int

PACM bit.

PAN unsigned int

Privileged Access Never state bit.

PM unsigned int

PMU exception mask bit.

PPEND unsigned int

PMU exception pending bit.

SM unsigned int

StreamingMode status bit.

SP unsigned int

Stack pointer register selection bit.

SS unsigned int

Software Step bit.

SSBS unsigned int

Speculative Store Bypass Safe bit.

TCO unsigned int

Tag Check Override bit.

TOUCHED unsigned int

Mask of the fields that have been touched since this trace has last been emitted. Note that touched does not necessarily mean the value changed. The index of the field is the bit index in the mask plus the index of this field plus one.

UAO unsigned int

User Access Override bit.

UINJ unsigned int

Undefined Instruction exceptions bit.

ZA unsigned int

ZA storage and ZT0 register enable bit.

nRW unsigned int

Current Execution state bit.

PSTATE_PACTIVE_OPMODE_TRANSITION

Trace pactive and Opmode mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Operating mode after the transition.

OLD_STATE enum

Operating mode before the transition.

PSTATE_PACTIVE_OPMODE_TRANSITION_DENIED

Trace the denied pactive and Opmode mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Operating mode after the transition.

OLD_STATE enum

Operating mode before the transition.

REASON string

Reason for the transition being denied.

PSTATE_PACTIVE_POWER_TRANSITION

Trace pactive and power mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Power mode after the transition.

OLD_STATE enum

Power mode before the transition.

PSTATE_PACTIVE_POWER_TRANSITION_DENIED

Trace the denied pactive and power mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Power mode after the transition.

OLD_STATE enum

Power mode before the transition.

REASON string

Reason for the transition being denied.

PSTATE_PACTIVE_THREAD_OPMODE_TRANSITION

Trace pactive and Opmode mode transitions for threads on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Operating mode after the transition.

OLD_STATE enum

Operating mode before the transition.

QUANTUM_START

Called at the start of every quantum.

RANGE_PRELOAD_REQUEST

Preload requests from RPRFM. Fields:

COUNT unsigned int

Number of blocks of data to access.

LENGTH unsigned int

Number of contiguous bytes to be accessed in each block.

OP unsigned int

Range prefetch operation .

REUSE unsigned int

Maximum number of bytes to be accessed before next RPRFM .

STRIDE signed int

Stride to advance after accessing LENGTH bytes.

VADDR unsigned int

Base virtual address of range prefetch request.

RUN_STATE

Run state transition. Fields:

INST_COUNT unsigned int

Ticks count at point of transition.

NEW enum

New run state.

OLD enum

Old run state.

SEMIHOSTING_CALL

Call of a semihost function occurred. Fields:

PC unsigned int

The program counter after the semihosting call.

SEMIHOSTING_PRECALL

About to call semihost. Fields:

EL2 bool

Use EL2 translation regime.

NS bool

Is Non-Secure.

REG_WIDTH unsigned int

The current register width in bytes.

SIGNAL

External signal state change. Fields:

SIGNAL enum

Signal that changed.

STATE bool

Signal asserted state.

SOFTWARE_STEP

Return the Debug Software Step state. Fields:

STATE enum

Software Step state.

SPSR

SPSR change. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

MODE enum

Which of the banked SPSR registers is written.

OLD_VALUE unsigned int

The old SPSR value.

VALUE unsigned int

The new SPSR value.

START_COMPILE

Compilation started. Fields:

VADDR unsigned int

Instruction where compilation begins.

STG_MULTIPLE_MICRO_OPS_ADDRESS_RANGE

Address range for MTE stores with multiple uops. Fields:

ADDR_END unsigned int

Final address.

ADDR_START unsigned int

Starting address.

STORE_MULTIPLE_REGS_ADDRESS_RANGE

Address range for stores with multiple registers. Fields:

ADDR_END unsigned int

Final address.

ADDR_START unsigned int

Starting address.

SUBOPTIMAL_LDST_RETIRED

suboptimal load/store. Fields:

ACCESS_SIZE unsigned int

Log2 of the access width in bytes used for alignment checking (1: half-word, 2: word ...).

OPTIMAL_ALIGN_SELECTOR unsigned int

Selector to decide the optimal alignment size (1: access width, 2: 4-bytes, 3: 8-bytes, ..., 12: 4-kbytes).

VADDR unsigned int

The virtual address of the access.

SYNC

Called for every synchronization, that is every quantum break. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The instruction count of this CPU.

LOCAL_QUANTUM unsigned int

The local quantum of this CPU.

LOCAL_TIME unsigned int

The local time of this CPU.

SYNC_REASON enum

Reason for sync.

SYSCALL

System call instruction executed. Fields:

IMM unsigned int

Immediate value of the system call instruction.

TYPE enum

System call type.

VADDR unsigned int

Instruction that caused the system call.

SYSREG_READ128

System Coprocessor 128 bit register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

REGNUM enum

Internal register number.

REGNUM_PHYS enum

Internal register number physically accessed (e.g. in presense of VHE redirection).

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_READ64

System Coprocessor register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REGNUM enum

Internal register number.

REGNUM_OPERAND enum

Register number corresponding to the instruction operand.

REGNUM_PHYS enum

Internal register number physically accessed (e.g. in presense of VHE redirection).

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_UPDATE128

Triggers when the system updates a register. Fields:

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG enum

Register number.

UNKNOWN unsigned int

Bits of the register which became unknown.

VALUE unsigned int

Value written to the register.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_UPDATE32

Triggers when the system updates a register. Fields:

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG enum

Register number.

UNKNOWN unsigned int

Bits of the register which became unknown.

VALUE unsigned int

Value written to the register.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_UPDATE64

Triggers when the system updates a register. Fields:

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG enum

Register number.

UNKNOWN unsigned int

Bits of the register which became unknown.

VALUE unsigned int

Value written to the register.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_WRITE128

System Coprocessor 128 bit register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

REGNUM enum

Internal register number.

REGNUM_PHYS enum

Internal register number physically accessed (e.g. in presense of VHE redirection).

UNDEF bool

The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_WRITE64

System Coprocessor register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REGNUM enum

Internal register number.

REGNUM_OPERAND enum

Register number corresponding to the instruction operand.

REGNUM_PHYS enum

Internal register number physically accessed (e.g. in presense of VHE redirection).

UNDEF bool

The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

UNALIGNED_LDST_RETIRED

Processor unaligned load/store. Fields:

ACCESS_SIZE unsigned int

Log2 of the access width in bytes used for alignment checking (1=>half-word, 2->word ...).

NUMBER_OF_BEATS unsigned int

Number of accesses (beats) in this burst (Total size in bytes is ACCESS_SIZE * NUMBER_OF_BEATS).

VADDR unsigned int

The virtual address of the access.

VFP_D_REGS

VFP/NEON D 64 bit register write. Fields:

ALIASING enum

Type of register aliasing used.

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a write to the corresponding bit in the register. For example, 0x1 would indicate a write to VALUE[0].

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

VFP_D_REGS_READ

VFP/NEON D 64 bit register read. Fields:

ID unsigned int

The register number.

VALUE unsigned int

The current value of the register.

VFP_Q_REGS

VFP/NEON Q 128 bit register write. Fields:

ALIASING enum

Type of register aliasing used.

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a write to the corresponding byte in the register. For example, 0x1 would indicate a write to VALUE[7:0].

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

VFP_Q_REGS_READ

VFP/NEON Q 128 bit register read. Fields:

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a read to the corresponding byte in the register. For example, 0x1 would indicate a read to VALUE[7:0].

VALUE unsigned int

The current value of the register.

VFP_SYS_REGS

Writes to the VFP/NEON units system registers. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ID enum

Which VFP system register is written.

MASK unsigned int

The mask indicating updated cumulative bits.

OLD_VALUE unsigned int

The register's old value overwritten.

VALUE unsigned int

The new value written to the VFP system register.

VFP_S_REGS

VFP/NEON S 32 bit register write. Fields:

ALIASING enum

Type of register aliasing used.

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a write to the corresponding bit in the register. For example, 0x1 would indicate a write to VALUE[0].

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

VFP_S_REGS_READ

VFP/NEON S 32 bit register read. Fields:

ID unsigned int

The register number.

VALUE unsigned int

The current value of the register.

VIRTUAL_TAG_ALLOCATED

Virtual tag information when allocated. Fields:

TAG_VADDR unsigned int

Virtual tag virtual address.

VADDR unsigned int

Data Virtual address.

VIRTUAL_TAG unsigned int

Virtual tag value.

WAYPOINT

Signals end of basic block execution. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ISET enum

Origin instruction set.

IS_COND bool

Indicates if this is a conditional waypoint.

PC unsigned int

Origin address (or 0 if unavailable).

TAKEN bool

Indicates if this waypoint was taken.

TARGET unsigned int

Destination address.

TARGET_ISET enum

Destination instruction set.

WFE_END

WFE ended. Fields:

INST_COUNT unsigned int

Ticks count when leaving WFE.

WFE_EVENT_REGISTER

WFE event register status: set/clear, reason. Fields:

INST_COUNT unsigned int

Ticks count.

REASON enum

Reason for set/clear. Only REASON==0 (Cleared by WFE) clears the bit, all other reasons set it.

WFE_IGNORED

WFE ignored. Fields:

EVENT bool

This WFE was ignored because the event register was set.

INST_COUNT unsigned int

Ticks count when ignoring WFE.

TRAPPED bool

This WFE was trapped.

WFE_START

WFE entered. Fields:

INST_COUNT unsigned int

Ticks count when entering WFE.

WFI_END

WFI ended. Fields:

INST_COUNT unsigned int

Ticks count when leaving WFI.

WFI_IGNORED

WFI ignored. Fields:

DISABLED bool

This WFI was ignored because WFI is disabled.

INST_COUNT unsigned int

Ticks count when ignoring WFI.

REASON enum

specifies reason why WFI trace was ignored.

TRAPPED bool

This WFI was trapped.

WFI_START

WFI entered. Fields:

INST_COUNT unsigned int

Ticks count when entering WFI.

WFI_WAKEUP

WFI wakeup. Fields:

INST_COUNT unsigned int

Ticks count when WFI wakeup occurred.

REASON enum

Reason for wakeup.

2.56 ARM_Cortex-X1

This section describes the trace sources.

AA64_ASE_SVE_REGS

Modification of AArch64 Advanced SIMD or SVE Vector Registers (via SIMD or Scalar operations), SVE Predicate Registers and ZTO. Fields:

ID enum

The register identifier.

MASK unsigned int

Bitmask of the register to signal the modified bits in the VALUE field.

SM bool

Whether the PE is in Streaming Mode.

VALUE unsigned int

Value of AArch64 vector register.

ASYNC_MEMORY_FAULT

Context ID Register write. Fields:

FAULT unsigned int

Fault status present in ESR.ISS encoding for Serror in ARCH64 and present in DFSR encoding for external abort format in ARCH32.

PADDR unsigned int

Physical Address (or 0 if unavailable).

VADDR signed int

Virtual Address (or 0 if unavailable).

ATOMIC_END_ACCESS

Bus trace access for atomics. Fields:

ACCESS_FAIL bool

Memory access failed.

ADDR unsigned int

The virtual address of the access.

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

COMPARE_VALUE unsigned int

Compare value for CAS.

LOAD_VALUE unsigned int

Loaded values.

NSDESC unsigned int

The security state of the access.

OPERAND_VALUE unsigned int

Operation's operand.

OPERATION enum

Operation type.

PADDR **unsigned int**

The physical address of the access.

PRIV **bool**

Is this a privileged access?.

ATOMIC_START_ACCESS

Bus trace access for atomics. Fields:

ADDR **unsigned int**

The virtual address of the access.

ATTR **unsigned int**

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

COMPARE_VALUE **unsigned int**

Compare value for CAS.

NSDESC **unsigned int**

The security state of the access.

OPERAND_VALUE **unsigned int**

Operation's operand.

OPERATION **enum**

Operation type.

PADDR **unsigned int**

The physical address of the access.

PRIV **bool**

Is this a privileged access?.

ArchMsg.Error.exit_code

DISPLAY exit code: %{EXITCODE} component kind: %{KIND} reason: %{REASON}. Fields:

EXITCODE **signed int**

Exit code to be returned.

KIND **string**

Component kind that invoked the exit code trace.

REASON **string**

Reason why the exit code trace was invoked (optional).

ArchMsg.Warning.IgnoredWarmResetRequest

Event fires when software attempts to request warm reset through write to deprecated DBGPRCR.CWRR.

ArchMsg.Warning.RAS_ErrselrUnpredictableState

DISPLAY ERRSELR is in unpredictable state due to value greater than number_of_error_records written to it.

ArchMsg.Warning.RAS_UnpredictableStateWrite

DISPLAY %{REG_NAME} write ignored due to unpredictable ERRSELR state. Fields:

REG_NAME string

Name of register being written.

ArchMsg.Warning.ReportedRevisionRegisterRead

DISPLAY Reported revision and patch do not match that of the cpu. Fields:

ORIGINAL_VALUE unsigned int

The value the register would have based on the modelled revision.

REGISTER string

The name of the register that has ben read.

REPORTED_VALUE unsigned int

The value the register has been overridden to.

ArchMsg.Warning.Unpredictably Indexed PM Event Register

TAGS **UNPREDICTABLE** DISPLAY Event %{IsDirect:(Selector value|Register)} %{SEL} is >= %{N}. Fields:

IsDirect unsigned int

Direct Access rather than indirect.

N unsigned int

Number of Accessible Registers.

SEL unsigned int

Selector.

ArchMsg.Warning.cache_contents_unknown

A cache has been activated without being invalidated and may contain **UNKNOWN** content. DISPLAY %{SIDE:(D|I)}-Cache was enabled in %{REGIME:(SECURE|NON_SECURE|HYP)} regime but was not invalidated since power-on: cache lines could contain **UNKNOWN** content %{SIDE: (Note: D-side caches must invalidate SECURE lines even if the cache is only used by NON_SECURE code in order to ensure that unexpected dirty lines tagged SECURE are not naturally evicted)}. Fields:

REGIME unsigned int

Current translation regime.

SIDE unsigned int

0 d-side cache or unified cache, 1 i-side.

ArchMsg.Warning.dcimvac_matches_watchpoint

TAGS **UNPREDICTABLE** DISPLAY Watchpoints matching an AArch32 DCIMVA are implementation defined.

ArchMsg.Warning.decode_fieldmismatch

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with different fields %{FIELD1} and %{FIELD2}. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_fracbitsnegative

TAGS **UNPREDICTABLE** DISPLAY VCVT Instruction is unpredictable with a negative number of fraction bits.

ArchMsg.Warning.decode_initblock

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable in an IT block.

ArchMsg.Warning.decode_initblocknotlast

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable in an IT block when not the last.

ArchMsg.Warning.decode_invalid_condition

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable when the condition is not AL.

ArchMsg.Warning.decode_invalidfieldcombination

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with the values in %{FIELD1} and %{FIELD2}. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_invalidvalue

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with the value in %{FIELD}. Fields:

FIELD enum

Field name.

ArchMsg.Warning.decode_mem_hint_unallocated

TAGS **UNPREDICTABLE** DISPLAY There is no memory hint allocated to this bit pattern.

ArchMsg.Warning.decode_registermatch

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with %{FIELD1} and %{FIELD2} as the same register. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_registermismatch

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with %{FIELD1} and %{FIELD2} as different registers. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_registeroutofrange

TAGS **UNPREDICTABLE** DISPLAY Floating-point instruction is unpredictable indexing beyond the end of the register bank.

ArchMsg.Warning.decode_sbzsbo

TAGS **UNPREDICTABLE** DISPLAY Reserved bits in the instruction are not canonical.

ArchMsg.Warning.decode_transactiontoobig

TAGS **UNPREDICTABLE** DISPLAY Attempt to perform a memory transaction of more than 128 bytes.

ArchMsg.Warning.decode_unpred_other

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable.

ArchMsg.Warning.decode_unpreduseofpc

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable when %{FIELD} is PC. Fields:

FIELD enum

Field name.

ArchMsg.Warning.decode_unpreduseofr13

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable when %{FIELD} is R13. Fields:

FIELD enum

Field name.

ArchMsg.Warning.decode_unpreduseofr13inreglist

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with R13 in register-list.

ArchMsg.Warning.decode_useofd16orhigher

TAGS **UNPREDICTABLE** DISPLAY Attempt to use register D16 or higher where it is not allowed.

ArchMsg.Warning.decode_writebackandbaseinlist

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with write back when the base register is in the transfer list.

ArchMsg.Warning.decode_zeroregistersinlist

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable when no registers are to be transferred.

ArchMsg.Warning.recursive_branch

An instruction is performing a branch that targets the same instruction. If this was intended then a WFI might be more effective. DISPLAY recursive branch detected at PC=%{PC}.

PRIMARY KEY PC. Fields:

PC unsigned int

Address of instruction causing the branch.

ArchMsg.Warning.recursive_exception

An instruction has generated an exception that targets the same instruction. DISPLAY recursive %{TYPE} exception detected at PC=%{PC} (CPSR=%{CPSR}). PRIMARY KEY PC. Fields:

CPSR unsigned int

Processor state.

PC unsigned int

Address of instruction causing the exception.

TYPE enum

Type of exception.

ArchMsg.Warning.reentrant_vector_catch

TAGS **UNPREDICTABLE** DISPLAY Vector catch debug event on Prefetch or Data abort vector %{ADDR}. Fields:

ADDR unsigned int

vector address.

ArchMsg.Warning.reserved_it_state

TAGS **UNPREDICTABLE** DISPLAY Execution with reserved IT state %{ITSTATE}. Fields:

ITSTATE unsigned int

reserved IT state.

ArchMsg.Warning.tlb_contents_unknown

Translation has been enabled without invalidating the TLB. DISPLAY Translation was enabled in the %{PAS:(Secure|Non-secure|Root|Realm)} EL%{EL:d} translation regime (vmid=%{VMID:x}) without invalidating %{INVALIDITY:(|d-side |i-side |)}entries since power-on: %{INVALIDITY:(|D||I)}TLB could contain **UNKNOWN** content. Fields:

EL unsigned int

Exception level of translation regime.

INVALIDITY unsigned int

Which TLB may be invalid (if TLBs are separate).

NON_SECURE bool

Whether the translation regime is non-secure.

PAS unsigned int

Physical address space of translation regime.

VMID unsigned int

Current VMID, if applicable.

ArchMsg.Warning.unknown_DLR_DSPSR

Exiting debug state with unknown DLR or DSPSR.

ArchMsg.Warning.unknown_ELR_SPSR

Returning from debug exception with unknown ELR or SPSR.

ArchMsg.Warning.unpred_FAR_on_LRCPC3_non_atomic_mem_access

TAGS **UNPREDICTABLE** Unpredictable FAR/FSTATUS on LRCPC3 pair instructions resulting in two separate memory accesses
DISPLAY Unpredictable FAR on LRCPC3 non atomic memory access at %{ADDR1} with FSTATUS %{FSTATUS1} or %{ADDR2} with FSTATUS %{FSTATUS2}. Fields:

ADDR1 unsigned int

Possible FAR address 1.

ADDR2 unsigned int

Possible FAR address 2.

FSTATUS1 unsigned int

Possible fault status value 1.

FSTATUS2 unsigned int

Possible fault status value 2.

ArchMsg.Warning.unpred_async_tag_check_fail_on_sve_speculative_load_do_update_ffr

Unpredictable set to FALSE of the FFR element associated to SVE speculative load with an async tag check fail TAGS **UNPREDICTABLE**.

ArchMsg.Warning.unpredictable_a32_breakpoint

TAGS **UNPREDICTABLE** DISPLAY Unpredictable breakpoint on a A32 instruction at %{ADDR1} Offset %{OFFSET} BAS %{BAS} for a %{IS_ADDRESS_MISMATCH_BREAKPOINT:(Mismatch|Match)} Breakpoint number %{N:d} Breakpoint address %{ADDR2}. Fields:

ADDR1 unsigned int

Address.

ADDR2 unsigned int

Breakpoint address.

BAS unsigned int

BAS field.

IS_ADDRESS_MISMATCH_BREAKPOINT bool

Is a Mismatch breakpoint, not a Match breakpoint.

N unsigned int

Breakpoint number.

OFFSET unsigned int

Always has a value of zero.

ArchMsg.Warning.unpredictable_banked_register_access

TAGS **UNPREDICTABLE** DISPLAY Access to banked register (R=%{R:d} SYSm=%{SYSM:2x}) is unpredictable. Fields:

R unsigned int

R bit.

SYSM unsigned int

SYSm.

ArchMsg.Warning.unpredictable_ccfail_undef

TAGS **UNPREDICTABLE** DISPLAY An instruction %{OPCODE} at VA=%{VADDR} failed its condition codes check but would otherwise have caused a non-data-dependent trap or undefined exception. Behaviour is implementation-defined choice of exception or nop. Fields:

OPCODE unsigned int

opcode of the instruction.

VADDR unsigned int

Virtual address of the instruction being executed.

ArchMsg.Warning.unpredictable_ici_bits

TAGS **UNPREDICTABLE** DISPLAY ICI bits do not apply or are architecturally unpredictable for instruction.

ArchMsg.Warning.unpredictable_t32_breakpoint

TAGS **UNPREDICTABLE** DISPLAY Unpredictable breakpoint on a T32 instruction at %{ADDR1} Offset %{OFFSET} BAS %{BAS} for a %{IS_ADDRESS_MISMATCH_BREAKPOINT:(Mismatch|Match)} Breakpoint number %{N:d} Breakpoint address %{ADDR2}. Fields:

ADDR1 unsigned int

Address.

ADDR2 unsigned int

Breakpoint address.

BAS unsigned int

BAS field.

IS_ADDRESS_MISMATCH_BREAKPOINT bool

Is a Mismatch breakpoint, not a Match breakpoint.

N unsigned int

Breakpoint number.

OFFSET unsigned int

Indicates which halfword of a 32-bit Thumb instruction the breakpoint is set on. 0 for the first halfword or if the instruction is 16 bits wide, or 1 for the second halfword.

ArchMsg.Warning.unpredictable_t32_vector_catch

TAGS **UNPREDICTABLE** DISPLAY Unpredictable Vector catch exception on a T32 instruction at %{ADDR} Offset %{OFFSET}. Fields:

ADDR unsigned int

Address.

OFFSET unsigned int

Indicates which halfword of a 32-bit Thumb instruction the vector catch exception is set on. 0 for the first halfword or if the instruction is 16 bits wide, or 1 for the second halfword.

ArchMsg.Warning.unpredictable_watchpoint_far

TAGS **UNPREDICTABLE** DISPLAY The value for FAR/EDWAR for hitting this watchpoint may be anywhere in the range %{LOWER_BOUND} to %{UPPER_BOUND}. Fields:

LOWER_BOUND unsigned int

the lowest address accessed by the instruction that triggered the watchpoint.

UPPER_BOUND unsigned int

the highest watchpointed address accessed by that instruction.

ArchMsg.Warning.unpredictable_watchpoint_far_isscmp

DISPLAY The value for FAR/EDWAR for hitting this watchpoint may be anywhere in the range %{LOWER_BOUND} to %{UPPER_BOUND}. Fields:

LOWER_BOUND unsigned int

the lowest address accessed by the instruction that triggered the watchpoint.

UPPER_BOUND unsigned int

the highest watchpointed address accessed by that instruction.

ArchMsg.Warning.warning_AdvSIMDExpandImmUnexpectedZero

TAGS **UNPREDICTABLE** DISPLAY AdvSIMDExpandImm may treat this immediate value as **UNPREDICTABLE**.

ArchMsg.Warning.warning_ConditionalSMC

TAGS **UNPREDICTABLE** DISPLAY SMC instruction has **UNPREDICTABLE** effects when conditional and when combined with traps.

ArchMsg.Warning.warning_TLBID_out_of_range

TLBID out of range in TLBI according to the configuration of NIS, NVIS, NOS and NVOS. Fields:

NIS unsigned int

NIS.

NOS unsigned int

NOS.

NVIS unsigned int

NVIS.

NVOS unsigned int

NVOS.

TLBID unsigned int

The TLBID value used in the TLBI instruction.

ArchMsg.Warning.warning_TTL_or_TG_mismatches_current_translation_granule

TTL or TG mismatches current translation granule in TLBI VA/RVA/IPA/RIPA. Fields:

CURRENT_TG unsigned int

Current TG as indicated by TCR_ELx or VTCR_EL2.

TLBI_TG unsigned int

TG (or TTL[3:2]) used in TLBI instruction.

ArchMsg.Warning.warning_access_crosses_page_boundary

TAGS **UNPREDICTABLE** DISPLAY Access crosses page boundary.

ArchMsg.Warning.warning_access_crosses_page_boundary_has_fault

TAGS **UNPREDICTABLE** DISPLAY Access that crosses page boundary has fault(s).

ArchMsg.Warning.warning_access_crosses_page_boundary_has_fault_on_both_pages

TAGS **UNPREDICTABLE** DISPLAY Access that crosses page boundary has a fault on each page.
Fields:

ADDR1 unsigned int

FAR address on first page.

ADDR2 unsigned int

FAR address on second page.

FSTATUS1 unsigned int

Fault status on first page.

FSTATUS2 unsigned int

Fault status on second page.

ArchMsg.Warning.warning_access_crosses_page_boundary_spanning_different_memory

TAGS **UNPREDICTABLE** DISPLAY Access at address %{ADDR} crosses page from MEM_TYPE %{MEMTYPE_PAGE1} to %{MEMTYPE_PAGE2}. Fields:

ADDR unsigned int

address of access crossing page.

MEMTYPE_PAGE1 signed int

First Page's memory type (0-Normal, 1-Non-Normal).

MEMTYPE_PAGE2 signed int

Second Page's memory type (0-Normal, 1-Non-Normal).

ArchMsg.Warning.warning_access_wraps_around_memory

TAGS **UNPREDICTABLE** DISPLAY Access wraps around memory in %{WIDTH:d}bit mode.
Fields:

WIDTH unsigned int

Address width.

ArchMsg.Warning.warning_bcr_linking_status

TAGS **UNPREDICTABLE** DISPLAY DBGBCR.{BT2,BT} combination gives unpredictable behaviour.

ArchMsg.Warning.warning_bcr_mask_reserved

TAGS **UNPREDICTABLE** DISPLAY DBGBCR_ELn.MASK is set to a reserved value.

ArchMsg.Warning.warning_bvr_masked_bit_not_zero

TAGS **UNPREDICTABLE** DISPLAY DBGBVRn_EL1.VA contains a bit masked by DBGBCRn_EL1.MASK that is not zero.

ArchMsg.Warning.warning_ccsidr_unimplemented_level

TAGS **UNPREDICTABLE** DISPLAY CCSIDR read while CSSELR %{CSSELR:x} points to an unimplemented cache level. Fields:

CSSELR unsigned int

current effective value of CSSELR.

ArchMsg.Warning.warning_change_to_ns_when_tge_set

TAGS **UNPREDICTABLE** DISPLAY Attempting to change NS to 1 when HCR.TGE = 1.

ArchMsg.Warning.warning_contiguous_bit_check_abort

TAGS **UNPREDICTABLE** DISPLAY An abort occurred whilst attempting to check TLB entry is contiguous at %{ENTRY_ADDR}. Fields:

ENTRY_ADDR unsigned int

address of conflicting TLB entry.

ArchMsg.Warning.warning_contiguous_bit_error

TAGS **UNPREDICTABLE** DISPLAY TLB entry at %{ENTRY_ADDR} was not contiguous with entry at %{CONTIG_ADDR} Contents are %{ENTRY_DATA} but expected %{CONTIG_DATA}. Fields:

CONTIG_ADDR unsigned int

address of the TLB first read, with which this entry is expected to be contiguous.

CONTIG_DATA unsigned int

expected contents based on the entry at CONTIG_ADDR.

ENTRY_ADDR unsigned int

address of conflicting TLB entry.

ENTRY_DATA unsigned int

contents of conflicting TLB entry.

ArchMsg.Warning.warning_contiguous_bit_error_D128

TAGS **UNPREDICTABLE** DISPLAY TLB entry at %{ENTRY_ADDR} was not contiguous with entry at %{CONTIG_ADDR} Contents are %{ENTRY_DATA_HI} %{ENTRY_DATA_LO} but expected %{CONTIG_DATA}. Fields:

CONTIG_ADDR unsigned int

address of the TLB first read, with which this entry is expected to be contiguous.

CONTIG_DATA unsigned int

expected contents based on the entry at CONTIG_ADDR.

ENTRY_ADDR unsigned int

address of conflicting TLB entry.

ENTRY_DATA_HI unsigned int

Higher bit contents of conflicting TLB entry.

ENTRY_DATA_LO unsigned int

Lower bit contents of conflicting TLB entry.

ArchMsg.Warning.warning_cp10_cp11_mismatch

TAGS **UNPREDICTABLE** DISPLAY Unpredictable configuration of CP10 and CP11 controls at EL
%{EL:d}. Fields:

EL unsigned int

EL owning the control.

ArchMsg.Warning.warning_cp10_cp11_reserved_value

TAGS **UNPREDICTABLE** DISPLAY Unpredictable configuration of CPACR CP10 (%{CP10}) or
CP11 (%{CP11}). Fields:

CP10 unsigned int

CP10 access.

CP11 unsigned int

CP11 access.

ArchMsg.Warning.warning_csselr_level_out_of_range

TAGS **UNPREDICTABLE** DISPLAY CSSELR.Level out of range, written value %{LEVEL} with only
%{IMPLEMENTED:d} levels implemented. Fields:

IMPLEMENTED unsigned int

number of cache levels implemented.

LEVEL unsigned int

written value of CSSELR.Level.

TYPE enum

The type of cache selected by CSSELR.InD and TnD.

ArchMsg.Warning.warning_debug_flow_control_bits_not_obeyed

TAGS **UNPREDICTABLE** DISPLAY Flow control bits for %{REGISTER} were not obeyed, causing
%{ERROR}. Fields:

ERROR enum

type of error resulting.

REGISTER enum

register under consideration.

ArchMsg.Warning.warning_debug_register_access_during_reset

TAGS **UNPREDICTABLE** DISPLAY Unpredictable %{IS_WRITE:(read|write)} access to debug register offset %{OFFSET} during reset. Fields:

IS_WRITE unsigned int

Write Not Read.

OFFSET unsigned int

Register Offset.

ArchMsg.Warning.warning_decode_cps_inconsistent_fields

TAGS **UNPREDICTABLE** DISPLAY The mode setting fields in the CPS instruction are inconsistent.

ArchMsg.Warning.warning_decode_invalid_state

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable in current exception-level/security state.

ArchMsg.Warning.warning_default_cacheable_mmu_on

TAGS **UNPREDICTABLE** DISPLAY Default cacheable is enabled (HCR.DC = 1) while MMU is enabled.

ArchMsg.Warning.warning_default_cacheable_vmmu_off

TAGS **UNPREDICTABLE** DISPLAY Default cacheable is enabled (HCR.DC = 1) while HCR.VM = 0.

ArchMsg.Warning.warning_deprecated_wvr_bit_2_set

TAGS **UNPREDICTABLE** DISPLAY DBGWVR%{N:d}_EL1=%{WVR:h} has bit 2 set, deprecated in ARMv8. Fields:

N unsigned int

index.

WVR unsigned int

DBGWVR.

ArchMsg.Warning.warning_eret_while_software_step_active_pending

Missing ISB between setting MDSCR_EL1.SS and ERET TAGS **UNPREDICTABLE**.

ArchMsg.Warning.warning_exclusive_to_non_normal

TAGS **UNPREDICTABLE** DISPLAY Exclusive Access to Non-Normal Memory %{ADDR}. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_exclusive_to_non_writeback

TAGS **UNPREDICTABLE** DISPLAY Exclusive Access to Non-Writeback-Cacheable Memory %{ADDR}. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_execute_from_device_memory

TAGS **UNPREDICTABLE** DISPLAY An attempt was made to execute from Device memory.

ArchMsg.Warning.warning_illegal_cpsr_mode

TAGS **UNPREDICTABLE** DISPLAY Writing an illegal or unimplement mode ({NEW_MODE:h}) to CPSR was unpredictable. Fields:

NEW_MODE unsigned int

New value of CPSR.M.

ArchMsg.Warning.warning_illegal_srs_mode

TAGS **UNPREDICTABLE** DISPLAY Illegal or **UNPREDICTABLE** mode ({MODE:h}) used for SRS instruction. Fields:

MODE unsigned int

mode for Banked SP.

ArchMsg.Warning.warning_implementation_defined_read_debug_register_in_SCS

Read Access to Debug registers from the processor is **IMPLEMENTATION DEFINED** TAGS **IMP_DEF**. Fields:

PPB_OFFSET unsigned int

debug register offset.

ArchMsg.Warning.warning_implementation_defined_sequential_security_transitions_supported

The behavior of sequential instruction fetches that cross from non-secure to secure memory and contain an SG instruction is **CONSTRAINED UNPREDICTABLE** TAGS **UNPREDICTABLE**.

ArchMsg.Warning.warning_implementation_defined_stack_limit_check_supported

It is **IMPLEMENTATION DEFINED** whether stack pointer limit checking is performed for this instruction TAGS **IMP_DEF**. Fields:

ADDRESS unsigned int

Address of instruction.

OPCODE unsigned int

instruction opcode.

ArchMsg.Warning.warning_implementation_defined_taggednonsharedmemorytype

TAGS **IMP_DEF** DISPLAY Memory attribute has Tagged, non shared memory type; it is **IMPLEMENTATION DEFINED** whether the memory location is treated as Tagged or non-Tagged.

ArchMsg.Warning.warning_implementation_defined_write_debug_register_in_SCS

Write Access to Debug registers from the processor is **IMPLEMENTATION DEFINED** TAGS **IMP_DEF**. Fields:

DATA unsigned int

data attempted to be written.

PPB_OFFSET unsigned int

debug register offset.

ArchMsg.Warning.warning_invalid_tcr_granule

TAGS UNPREDICTABLE DISPLAY TCR.TG{%TG_ID:(0|1)} bits have been set to a granule size {%REQUEST:d}K not implemented - using {%SUBSTITUTE:d}K. Fields:

REQUEST signed int

page size requested (or 0 for reserved).

SUBSTITUTE signed int

best guess available page size.

TG_ID bool

bits TG0 or TG1.

ArchMsg.Warning.warning_load_multiple_user_registers_from_user_mode

TAGS UNPREDICTABLE DISPLAY An LDM(user registers) instruction executed from user mode.

ArchMsg.Warning.warning_mte_in_mem_attr_not_supported

TAGS UNPREDICTABLE DISPLAY Memory attribute {%ATTR} has Tagged memory type but Armv8.5-MTE is not implemented. Fields:

ATTR unsigned int

Memory attribute.

ArchMsg.Warning.warning_reserved_breakpoint_state_match

TAGS UNPREDICTABLE DISPLAY {%IS_BREAKPOINT:(Watchpoint|Breakpoint)} programmed with a reserved combination of HMC, SSC and {%IS_BREAKPOINT:(PAC|PMC)}. Fields:

IS_BREAKPOINT unsigned int

Is a breakpoint, not a watchpoint.

ArchMsg.Warning.warning_shareability

TAGS UNPREDICTABLE DISPLAY Unpredictable: combination of the 1st and 2nd stages of translation is Normal Inner Non-Cacheable, Outer Non-Cacheable.

ArchMsg.Warning.warning_software_step_set_while_enabled

TAGS UNPREDICTABLE DISPLAY MDSCR_EL1.SS set to 1 while software step debug exceptions are enabled.

ArchMsg.Warning.warning_thumb_instruction_wraps_around_memory

TAGS UNPREDICTABLE DISPLAY Thumb instruction wraps around memory.

ArchMsg.Warning.warning_ttbr_sbz_bits_are_not_zero

TAGS UNPREDICTABLE DISPLAY {%TTBR}[{%MSB:d}:0] should be zero, but are not. Fields:

MSB unsigned int

x-1.

TTBR enum

which TTBR register.

ArchMsg.Warning.warning_unaligned_address_dbgdtrrx_write

TAGS **UNPREDICTABLE** DISPLAY Unaligned address %{ADDR} in X0/R0 when DBGDTRRX written in memory access mode. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unaligned_address_dbgdtrtx_read

TAGS **UNPREDICTABLE** DISPLAY Unaligned address %{ADDR} in X0/R0 when DBGDTRTX read in memory access mode. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unaligned_to_device

TAGS **UNPREDICTABLE** DISPLAY Unaligned Access to Device Memory %{ADDR}. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unaligned_to_strongly_ordered

TAGS **UNPREDICTABLE** DISPLAY Unaligned Access to Strongly Ordered Memory %{ADDR}. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unknown_sau_rnr

TAGS **UNPREDICTABLE** DISPLAY SAU_RNR was set to an unsupported value.

ArchMsg.Warning.warning_unpredictable_AIRCR_PRIS_and_BFHFNMINs

TAGS **UNPREDICTABLE** DISPLAY The effect of setting both AIRCR.BFHFNMINs and AIRCR.PRIS to 1 is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_AIRCR_VECTCLRACTIVE_when_not_in_debug

TAGS **UNPREDICTABLE** DISPLAY The effect of writing a 1 to AIRCR.VECTCLRACTIVE if the processor is not halted in Debug state is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_AIRCR_VECTRESET_and_SYSRESETREQ

TAGS **UNPREDICTABLE** DISPLAY When the processor is halted in Debug state, if a write to the register writes a 1 to both VECTRESET and SYSRESETREQ, the behavior is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_AIRCR_VECTRESET_when_not_in_debug

TAGS **UNPREDICTABLE** DISPLAY The effect of writing a 1 to AIRCR.VECTRESET if the processor is not halted in Debug state is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_AIRCR_incorrect_VKEY

TAGS **UNPREDICTABLE** DISPLAY The value 0x05FA must be written to AIRCR.VKEY, otherwise the register write is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_EXC_RETURN_Reserved_Bit

EXC_RETURN[23:7] are reserved with the special condition that all bits should be written as one. Values other than all 1s are **UNPREDICTABLE**. TAGS **UNPREDICTABLE**. Fields:

EXC_RETURN unsigned int

Special PC value.

SBO_MASK unsigned int

Which bits Should Be One to not UNPRED.

ArchMsg.Warning.warning_unpredictable_PMEVTYPEPn_ELO_VS_reserved_value

TAGS **UNPREDICTABLE** DISPLAY Unpredictable reserved value of PMEVTYPEN<n>_ELO.VS.

ArchMsg.Warning.warning_unpredictable_change_to_DEMCR_MON_STEP_at_insufficient_priority

TAGS **UNPREDICTABLE** DISPLAY The effect of changing DEMCR.MON_STEP at an execution priority that is lower than the priority of the DebugMonitor exception is **CONSTRAINED UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_change_to_priority_of_active_exception

TAGS **UNPREDICTABLE** DISPLAY Changing the priority of an active exception is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_clear_lspact

TAGS **UNPREDICTABLE** DISPLAY **UNPREDICTABLE** state on context switch. Fields:

DRVEXC_TAKEN unsigned int

Derived exception taken, not pended.

REACHED_LAST_STORE unsigned int

Only the last faulted.

ArchMsg.Warning.warning_unpredictable_dbg_exit_unaligned_dlr

TAGS **UNPREDICTABLE** DISPLAY Debug exit to AArch32 with DLR[0] set to 1. PC[0] can be set to 0 or to DLR[0].

ArchMsg.Warning.warning_unpredictable_exception_catch

TAGS **UNPREDICTABLE** DISPLAY Generation of an exception catch event is unpredictable.

ArchMsg.Warning.warning_unpredictable_exception_return_inconsistent_state

TAGS **UNPREDICTABLE** DISPLAY state on exception return is unpredictable.

ArchMsg.Warning.warning_unpredictable_exception_return_instruction

TAGS **UNPREDICTABLE** DISPLAY exception return instruction is unpredictable.

ArchMsg.Warning.warning_unpredictable_in_debug_state

TAGS **UNPREDICTABLE** DISPLAY Instruction is **UNPREDICTABLE** when executed in debug state.

ArchMsg.Warning.warning_unpredictable_mair_encoding

TAGS **UNPREDICTABLE** DISPLAY Memory attribute %{ATTR} has invalid MAIR encoding; this may lead to unpredictable results. Fields:

ATTR unsigned int

Memory attribute.

ArchMsg.Warning.warning_unpredictable_pmu_counter_access

TAGS **UNPREDICTABLE** DISPLAY Access to PMU counters from non-secure EL0 or EL1 is **UNPREDICTABLE** with MDCR_EL2.HPMN set to 0.

ArchMsg.Warning.warning_unpredictable_prioritization_breakpoint_match_vector_catch

TAGS **UNPREDICTABLE** DISPLAY Two events with the same priority occurred at the same instruction: Address Matching Vector Catch debug event, and Breakpoint debug event. It is constrained unpredictable which is taken.

ArchMsg.Warning.warning_unpredictable_stack_selection

TAGS **UNPREDICTABLE** DISPLAY **UNPREDICTABLE** state on context switch.

ArchMsg.Warning.warning_unpredictable_tsize_out_of_range

TAGS **UNPREDICTABLE** DISPLAY %{TCR}.T%{TTBR:d}SZ (%{TSIZE:d} bits) is out of range. Must be between 25 and %{TSIZE_MAX:d}. Fields:

TCR enum

Which TCR.

TSIZE signed int

Value of TSize.

TSIZE_MAX signed int

The maximum allowed value.

TTBR unsigned int

Which TTBR region.

ArchMsg.Warning.warning_unpredictable_unaligned_pc_as_base_register

TAGS **UNPREDICTABLE** DISPLAY The PC value must be word-aligned, otherwise the behavior of the instruction is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_unaligned_pop_stack

TAGS **UNPREDICTABLE** DISPLAY **CONSTRAINED UNPREDICTABLE** unaligned stack frame %{FRAME_ADDR:x} in exception-return. Fields:

FRAME_ADDR unsigned int

unaligned stack frame address.

ArchMsg.Warning.warning_unpredictable_unaligned_to_non_writeback

TAGS **UNPREDICTABLE** DISPLAY Unaligned Access to Non-Writeback-Cacheable Memory.

ArchMsg.Warning.warning_unpredictable_vmsa_memattrib

TAGS **UNPREDICTABLE** DISPLAY Unpredictable vmsa memory attribute with texcb value programmed to %{TEXCB} when tex remap is %{TEX_REMAP}. Fields:

TEXCB unsigned int

texcb value.

TEX_REMAP bool

use tex remap.

ArchMsg.Warning.warning_unpredictable_vtcr_t0sz_sl0

TAGS **UNPREDICTABLE** DISPLAY The combination of VTCR.TOSZ=%{TOSZ:d} and VTCR.SLO=%{SLO:d} is unpredictable. Fields:

SLO signed int

VTCR.SLO, starting level for stage 2 translation table walks.

TOSZ signed int

VTCR.TOSZ, required input address range.

ArchMsg.Warning.warning_unpredictable_write_DHCSR

TAGS **UNPREDICTABLE** DISPLAY **UNPREDICTABLE** set of bit-changes in DHCSR. Fields:

NEW unsigned int

data attempted to be written.

OLD unsigned int

previous value.

ArchMsg.Warning.warning_unpredictable_write_SHCSR

TAGS **UNPREDICTABLE** DISPLAY **UNPREDICTABLE** setting a pending bit to 1 for an exception with priority >= execution priority.

ArchMsg.Warning.warning_unsupported_access_to_memory_mapped_register

TAGS **UNPREDICTABLE** DISPLAY Access to memory mapped register is unsupported.

ArchMsg.Warning.warning_user_jmcr_access

TAGS **UNPREDICTABLE** DISPLAY User %{WRITE:(write|read)} access to JMCR is unpredictable. Fields:

WRITE unsigned int

access is write.

ArchMsg.Warning.warning_wcr_mask_and_bas

TAGS **UNPREDICTABLE** DISPLAY DBGWCR_EL1.MASK is non-zero and BAS is not 0xff.

ArchMsg.Warning.warning_wcr_mask_reserved

TAGS **UNPREDICTABLE** DISPLAY DBGWCR_ELn.MASK is set to a reserved value.

ArchMsg.Warning.warning_wcr_non_configuous_bas

TAGS **UNPREDICTABLE** DISPLAY DBGWCR_EL1.BAS has non-contiguous set of ones.

ArchMsg.Warning.warning_wvr_masked_bit_not_zero

TAGS **UNPREDICTABLE** DISPLAY DBGWVRn_EL1.VA contains a bit masked by DBGWCRn_EL1.MASK that is not zero.

ArchMsg.Why.why_branch_target_exception

A branch target exception has occurred. DISPLAY branch target exception: target pc=%{PC}, target opcode=%{OPCODE}. PRIMARY KEY PC. Fields:

BTYPE unsigned int

pstate.BTYPE at time branch taken.

OPCODE unsigned int

opcode at branch target.

PC unsigned int

Virtual Address of branch target.

ArchMsg.Why.why_illegal_state

An illegal mode change has occurred. DISPLAY illegal mode change: %{MESSAGE}. Fields:

MESSAGE string

The message.

ArchMsg.warning_invalid_interval_counter_reload_value

Invalid interval counter reload value in PMSIRR_EL1.INTERVAL: software should set this to a value greater than or equal to the minimum indicated by PMSIDR_EL1.Interval.

ArchMsg.warning_invalid_spe_buffer_write_pointer

Invalid value programmed in PMBPTR_EL1.PTR or PMBLIMITR_EL1.LIMIT: Current write pointer must be at least one sample record below the write limit pointer and PMBPTR_EL1.PTR[63:56] must be equal to PMBLIMITR_EL1.LIMIT[63:56].

BRANCH_MISPREDICT

Simulating branch mispredict. Fields:

PC unsigned int

Origin address (or 0 if unavailable).

BRA_DIR

Direct branches, to immediate address. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISSET enum

The instruction set of the branch instruction.

IS_COND bool

Indicates if this is a conditional branch.

IS_HINTED bool

Whether the conditional instruction is hinted.

IS_ISB bool

Whether the instruction is ISB.

IS_LINK bool

Indicates if this is a branch with link.

PC unsigned int

The address of the branch instruction.

TARGET_ISET enum

The instruction set after the branch.

TARGET_PC unsigned int

The address the instruction branches to.

BRA_INDIR

Indirect branches, perhaps to a register. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISET enum

The instruction set of the branch instruction.

IS_COND bool

Indicates if this is a conditional branch.

IS_HINTED bool

Whether the conditional instruction is hinted.

IS_ISB bool

Whether the instruction is ISB.

IS_LINK bool

Indicates if this is a branch with link.

PC unsigned int

The address of the branch instruction.

TARGET_ISET enum

The instruction set after the branch.

TARGET_PC unsigned int

The address the instruction branches to.

BRA_INDIR_CCFAIL

Indirect branches fail, perhaps to a register.

BRA_RET_CCFAIL

Branch return fail, perhaps to a register.

CACHE_MAINTENANCE_OP

Cache Maintenance Operation. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

DATA unsigned int

Specified MVA or Set/way.

FUNCTION enum

Clean / Invalidate.

SCOPE enum

Affected region.

SIDE enum

Inst / Data.

CCFAIL

Conditional instruction condition check fail. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CCFAIL_UNC

Unconditional instruction condition check fail. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CCPASS

Conditional instruction condition check pass. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CCPASS_UNC

Unconditional instruction condition check pass. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CODE_CACHE_MAINT

Code cache maintenance. Fields:

END_INCL unsigned int

Inclusive end address for VA or PA requests.

NS bool

Non-secure world for PA.

START unsigned int

Start address for VA or PA requests.

TYPE enum

Request type.

COMPILE_BLOCK_END

Last instruction of basic block translated. Fields:

VADDR unsigned int

Address of next instruction after this basic block.

COMPILE_BLOCK_START

First instruction of basic block translated. Fields:

VADDR unsigned int

Address of first instruction in basic block.

COMPILE_INST

ARM instruction compiled. Fields:

DISASS string

Disassembly of the instruction.

ISSET enum

The instruction set of this instruction.

ITSTATE unsigned int

The ITSTATE current for the instruction.

OPCODE unsigned int

The opcode of the instruction.

PC unsigned int

The address of the instruction.

SIZE unsigned int

The size of the instruction in bytes.

CONTEXTIDR

Context ID Register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

NS bool

Secure or nonsecure banked register is accessed.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The new value written.

CONTEXT_SYNC

Called for every context synchronization event. Fields:

ARCH_CSE bool

Architectural Context Synchronization Event.

PC unsigned int

The address of the synchronization instruction.

SYSREG_SYNC bool

System register synchronization event.

CORE_ENDIAN

Core BE8 Big-Endian state changed. Fields:

BE8 bool

Core BE8 Big-Endian state.

CORE_INFO

Static processor attributes. Only triggered by a call to DumpState(). Fields:

ARCH_PROFILE enum

The architecture profile of the core.

CLUSTER_ID unsigned int

The cluster ID of this processor.

CORE_NUM unsigned int

The number of this core in an MP processor.

DETAILED_CORE_FEATURES enum

Detailed core features as a bitfield. Enum values are bitmasks to extract the info from the field.

FPU_VERSION enum

The VFP version implemented by the core.

MEM_ARCH enum

The memory architecture of the core.

MIDR_EL1 unsigned int

MIDR_EL1 value.

NUM_CORES unsigned int

The number of cores in this MP processor.

QUANTUM_SIZE unsigned int

The default quantum size of the core (in ticks).

SECURITY_FEATURES bool

Does the core have security features?.

CORE_LOADS

Processor load accesses. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ACQREL enum

Is this an acquire/release.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DATA unsigned int

The data read or written.

ELEMENT_SIZE unsigned int

Width of each element.

ENDIANNESS enum

Byte ordering of this access.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

LOCK enum

Normal, exclusive or locked access.

MECID unsigned int

Memory Encryption Context ID, if applicable.

MECID2 unsigned int

Memory Encryption Context ID of the second page, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PAGE_SIZE unsigned int

Page size as $\log_2(\text{size in bytes})$.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK, 2=Exclusive Failed, 3=Unsupported.

SIZE unsigned int

Width of the access in bytes. Only required if DATA is not traced.

TRANS bool

Is this a translated access.

VADDR unsigned int

The virtual address of the access.

CORE_REGS

Changes of the core registers R0 to R14. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number, 0 to 14.

MODE enum

Bank of the register accessed.

OLD_VALUE unsigned int

The old value overwritten.

PHYS_ID enum

The physical register accessed.

VALUE unsigned int

The new value written to the register.

CORE_REGS64

Changes of the core registers X0..X30, SP_ELn. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ID enum

The register number, 0-30 for X0-X30, >=32 for SP_ELn.

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

CORE_REGS64_READ

General purpose AArch64 register has been read X0..X30, SP_ELn. Fields:

ID enum

The register number, 0-30 for X0-X30, >=32 for SP_ELn.

VALUE unsigned int

The value read from the register.

CORE_STALLED_TRANSITION

Detect when a core is prevented from progressing PChannel states. Fields:

CURRENT_STATE enum

Current power mode of core.

REASON string

The reason a transition was stalled, and what must be done to unblock it.

TARGET_STATE enum

Desired power mode of core after transition.

CORE_STORES

Processor store accesses. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ACQREL enum

Is this an acquire/release.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DATA unsigned int

The data read or written.

ELEMENT_SIZE unsigned int

Width of each element.

ENDIANNESS enum

Byte ordering of this access.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

LOCK enum

Normal, exclusive or locked access.

MECID unsigned int

Memory Encryption Context ID, if applicable.

MECID2 unsigned int

Memory Encryption Context ID of the second page, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PAGE_SIZE unsigned int

Page size as $\log_2(\text{size in bytes})$.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK, 2=Exclusive Failed, 3=Unsupported.

SIZE unsigned int

Width of the access in bytes. Only required if DATA is not traced.

TRANS bool

Is this a translated access.

VADDR unsigned int

The virtual address of the access.

CP14_READ

System Coprocessor register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP14 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc1 unsigned int

opcode 1.

opc2 unsigned int
opcode 2.

CP14_WRITE

System Coprocessor register write. Fields:

CORE_NUM unsigned int
Core number in a multi processor.

CRm unsigned int
CRm.

CRn unsigned int
CRn.

NS bool
Secure or nonsecure banked register is accessed.

REG_NAME string
Name of the CP14 register accessed.

SECURITY_STATE enum
The current security state.

UNDEF bool
The register accessed is undefined.

UPDATED_VALUE unsigned int
Updated value of the register now it has been written.

VALUE unsigned int
The new value written.

opc1 unsigned int
opcode 1.

opc2 unsigned int
opcode 2.

CP15_READ

System Coprocessor register read. Fields:

CORE_NUM unsigned int
Core number in a multi processor.

CRm unsigned int
CRm.

CRn unsigned int
CRn.

NS bool
Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP15 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc1 unsigned int

opcode 1.

opc2 unsigned int

opcode 2.

CP15_READ64

System Coprocessor register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP15 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc unsigned int

opcode 1.

CP15_WRITE

System Control Coprocessor register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP15 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc1 unsigned int

opcode 1.

opc2 unsigned int

opcode 2.

CP15_WRITE64

System Coprocessor register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP15 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc unsigned int

opcode 1.

CPSR

CPSR change. (In AArch64 PSTATE represented in SPSR format). Fields:

CORE_NUM unsigned int

Core number in a multi processor.

OLD_VALUE unsigned int

The old CPSR value.

UNKNOWN unsigned int

Bits within the register that have unknown value.

VALUE unsigned int

The new CPSR value.

CRYPTO_SPEC

Every crypto instruction speculatively executed.

CorePowerStateChange

Core power-up state has changed. DISPLAY Core powered %{PU:(up|down)}. Fields:

PU bool

Whether the core powered up.

DATA_CACHE_SET_ALLOC_TAGS

Set allocation tags from DC GVA instruction. Fields:

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

NSDESC unsigned int

The security state of the access.

PADDR unsigned int

The physical address of the access.

PAS enum

The physical address space of the page.

RESPONSE enum

0=Aborted, 1=OK.

SIZE unsigned int

The size of memory in bytes.

TAG unsigned int

The value of the tag being set.

TAGGING_TYPE enum

Type of Memory tagging (Physical or Virtual).

TAG_VADDR unsigned int

The virtual address of the Virtual Tag if Virtual tagging is used; otherwise 0.

VADDR unsigned int

Virtual address used for setting allocation tags.

DATA_CACHE_ZERO

Zero and invalidate cache line from DC ZVA instruction. Fields:

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

NSDESC unsigned int

The security state of the access.

PADDR unsigned int

The physical address of the access.

PADDR_TAG unsigned int

If tag checked, the physical address tag for the access.

PAS enum

The physical address space of the page.

RESPONSE enum

0=Aborted, 1=OK.

SIZE unsigned int

The size of zero'd memory in bytes.

TAG_CHECKED bool

Is this access tag checked.

VADDR unsigned int

Virtual address of the zero'd cache line.

DEBUG_EVENT

Hardware debug support event. Fields:

EVENT enum

Description of event.

VALUE unsigned int

data value.

DEBUG_TARGET_ISET

Hardware debug target ISA. Fields:

TARGET_ISET enum

The target instruction set of the debug state.

DMI_ALLOCATE

DMI allocated. Fields:

ACCESS_DIRECTION enum

READ / WRITE.

PADDR unsigned int

The physical address of the DMI.

PTR unsigned int

DMI range pointer.

SIDE enum

DSIDE / ISIDE.

SIZE unsigned int

The size of DMI.

VADDR unsigned int

The virtual address of the DMI..

DMI_HIT

DMI hit, access used DMI (only fire for small DMI regions i.e. less than a full page). Fields:

ACCESS_DIRECTION enum

READ / WRITE.

PADDR unsigned int

The physical address of the DMI.

SIDE enum

DSIDE / ISIDE.

SIZE unsigned int

The size of the DMI.

VADDR unsigned int

The virtual address of the DMI..

DMI_REVOKE

DMI revoked. Fields:

ACCESS_DIRECTION enum

READ / WRITE.

PADDR unsigned int

The physical address of the DMI.

SIDE enum

DSIDE / ISIDE.

SIZE unsigned int

The size of DMI.

VADDR unsigned int

The virtual address of the DMI..

END_COMPILE

Compilation end. Fields:

END_OF_PAGE_COUNT unsigned int

Number of basic blocks exited due to page boundary since START_COMPILE.

FETCHFAIL_COUNT unsigned int

Number of basic blocks exited due to ifetch failure START_COMPILE.

INST_COUNT unsigned int

Number of instructions compiled since START_COMPILE.

NONSEQ_COUNT unsigned int

Number of basic blocks exited due to non-sequential instructions since START_COMPILE.

PAGE_STRADDLE_COUNT unsigned int

Number of basic blocks exited due to unaligned instructions crossing page since START_COMPILE.

ESB

Error Synchronization Barrier hit. Fields:

ASYNC bool

ESB is async.

IMPLICIT bool

ESB is implicit.

EXCEPTION

Exception taken. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ESR unsigned int

The value assigned to ESR when exception is taken.

ESR_EC unsigned int

The Value assigned to ESR.EC when exception is taken.

IS_PHYSICAL bool

Physical or Virtual exception.

IS_TRAP bool

Exception coming from a trap.

LR unsigned int

The value assigned to the link register.

PC unsigned int

The location where the exception occurred.

PREFERRED_RETURN unsigned int

The preferred return address for the exception.

TAKEN_LOCALLY bool

Whether exception is taken locally(same exception level).

TARGET_ISET enum

The instruction set of the exception handler code.

TARGET_PC unsigned int

The address the exception branches to.

VECTOR enum

The exception vector.

VECTOR_OFFSET unsigned int

The offset of an exception vector from the base address.

EXCEPTION_END

Every exception completed.

EXCEPTION_INFO

Trace human-readable reason of the current exception for data/instruction aborts, floating point exceptions and unknown exceptions. Fields:

DESCRIPTION string

Reason for current exception.

ESR_EC unsigned int

The value of ESR.EC when exception is taken.

EXCEPTION_RAISE

Every exception raised.

EXCEPTION_RETURN

Branches on leaving exception. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISET enum

The instruction set of the branch instruction.

IS_COND bool

Indicates if this is a conditional branch.

IS_HINTED bool

Whether the conditional instruction is hinted.

IS_ISB bool

Whether the instruction is ISB.

IS_LINK bool

Indicates if this is a branch with link.

PC unsigned int

The address of the branch instruction.

TARGET_ISET enum

The instruction set after the branch.

TARGET_PC unsigned int

The address the instruction branches to.

EXCEPTION_RETURN_PREBRANCH

Exception return event that must occur before PC and processor state have updated.

EXCEPTION_START

Every exception started.

FIQ_TAKEN

FIQ taken exception.

FP_EXCEPTIONS_SIGNALLED

Trace which FP exceptions were signalled as a part of the FP operations performed by an instruction. Fields:

FP_EXCEPTIONS_SIGNALLED_MASK unsigned int

Indicates which FP exceptions were signalled by an instruction. This uses the same encoding as the cumulative floating-point exception bitfields in FPSR.

FP_STATE

Floating point state. Fields:

FPSR unsigned int

FPSR register value.

MASK unsigned int

The mask indicating updated cumulative bits.

FREQ_CHANGED

Core frequency changed. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The instruction count of this core.

NEW_FREQ signed int

The new frequency of this core (expressed in Hz).

OLD_FREQ signed int

The old frequency of this core (expressed in Hz).

HLT

HLT instruction occurred. Fields:

IMM16 unsigned int

The 16-bit immediate value encoded in HLT instruction.

INST

Every instruction executed. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DEBUG_STATE bool

The instruction is executed in debug state.

DISASS string

Disassembly of instruction.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISSET enum

The current instruction set.

ITSTATE unsigned int

The current ITSTATE.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

MODE enum

The mode the core is in.

NS unsigned int

The core's non-secure bit.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

OPCODE unsigned int

The opcode of the instruction.

PADDR unsigned int

The physical address of the instruction.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the instruction.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the page.

PC unsigned int

The address of the instruction.

SECURITY_STATE enum

The core's security state.

SIZE unsigned int

The size of the instruction in bytes.

INST_END

Every instruction completed.

INST_START

Every instruction started. Fields:

DEBUG_STATE bool

The instruction is executed in debug state.

ISSET enum

The current instruction set.

MODE enum

The mode the core is in.

NS enum

The current Secure State.

PC unsigned int

The address of the conditional instruction.

SECURITY_STATE enum

The current Security State.

INST_STRADDLE

Instruction straddles boundary. Fields:

DEBUG_STATE bool

The instruction is executed in debug state.

ISSET enum

The current instruction set.

MODE enum

The mode the core is in.

NS enum

The current Secure State.

PC unsigned int

The address of the conditional instruction.

SECURITY_STATE enum

The current Security State.

IRQ_TAKEN

IRQ taken exception.

ISB

ISB instruction occurred. Fields:

PC unsigned int

The address of the ISB instruction.

LOAD_MULTIPLE_REGS_LIST

List of destination registers (not source) of a multiple registers load instruction. Fields:

REG string

Register to add into the list.

LOCAL_MONITOR

Local monitor activity. Fields:

PADDR unsigned int

Local Monitor Address.

State enum

State of the monitor (Open/Exclusive).

MAPPING_CREATE

A memory mapping has been created. This is an implementation detail of the model and so may change without notice. Fields:

ASID unsigned int

The ASID used during translation, or 0 if no ASID was used.

IS_VALID bool

Whether the mapping is valid or not.

PADDR unsigned int

Base physical address of region, or 0 if not known.

SIDE enum

I-side or D-side.

SIZE unsigned int

Size of the page, or 0 if not known.

VADDR unsigned int

Base virtual address of region.

VMID unsigned int

The VMID used during translation, or 0 if no VMID was used.

MAPPING_DESTROY

A memory mapping has been destroyed. This is an implementation detail of the model and so may change without notice. Fields:

ASID unsigned int

The ASID used during translation.

CAUSE enum

The cause of invalidation.

PADDR unsigned int

Base physical address of page.

SIDE enum

I-side or D-side.

SIZE unsigned int

Size of the page.

VADDR unsigned int

Base virtual address of page.

VMID unsigned int

The VMID used during translation.

MEMMAP_DEBUG_READ

Memory mapped reads to the debug registers. Fields:

ADDR unsigned int

address.

CORE_NUM unsigned int

Core number in a multi processor.

EXT bool

Whether access is from an external device (such as the DAP).

NS enum

Secure state of the access.

REG_NAME string

Name of the debug register accessed.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

MEMMAP_DEBUG_WRITE

Memory mapped writes to the debug registers. Fields:

ADDR unsigned int

address.

CORE_NUM unsigned int

Core number in a multi processor.

EXT bool

Whether access is from an external device (such as the DAP).

NS enum

Secure state of the access.

REG_NAME string

Name of the debug register accessed.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The new value written.

MMU_D128_TTB_READ

This event is triggered by reads caused by a D128 translation table walk. Fields:

ABORTED bool

Was the walk successful.

DATA_HIGH unsigned int

The data upper bits read.

DATA_LOW unsigned int

The data lower bits read.

IPA unsigned int

For stage 1, the IPA of the read.

LEVEL unsigned int

Translation table level.

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

OUTADDR unsigned int

The output physical address.

PADDR unsigned int

The physical address of the read.

PASmem enum

Physical Address Space of the accessed memory.

PASreq enum

Physical Address Space of the requested lookup.

REGIME_EL enum

Entry matches in this translation regime.

REQUESTER enum

The requester of this table walk.

SIDE enum

Inst / Data.

STAGE unsigned int

Translation stage.

MMU_D128_TTB_WRITE

This event is triggered by writes caused by a D128 translation table walk. Fields:

ABORTED bool

Was the walk successful.

DATA_HIGH unsigned int

The data upper bits written.

DATA_LOW unsigned int

The data lower bits written.

IPA unsigned int

For stage 1, the IPA of the write.

IS_HACDBS_ACCESS bool

If the access was due to HACDBS_CLEAN.

LEVEL unsigned int

Translation table level.

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

OUTADDR unsigned int

The output physical address.

PADDR unsigned int

The physical address of the write.

PASmem enum

Physical Address Space of the accessed memory.

PASreq enum

Physical Address Space of the requested lookup.

REGIME_EL enum

Entry matches in this translation regime.

SIDE enum

Inst / Data.

STAGE unsigned int

Translation stage.

MMU_TRANS

Address translation information. Fields:

ASID unsigned int

Address space identifier.

CORE_NUM unsigned int

Core number in a multi processor.

Hyp bool

Entry matches in Hyp state only.

INNERCACHE_RA bool

Is the inner cache allocate on read.

INNERCACHE_TRANSIENT bool

Is the inner write-through transient.

INNERCACHE_TYPE enum

Inner Caching scheme (NC/MB/WA).

INNERCACHE_WA bool

Is the inner cache allocate on write.

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MEMTYPE enum

Memory type.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC enum

Is secure side supposed to access secure or nonsecure memory.

OUTERCACHE_RA bool

Is the outer cache allocate on read.

OUTERCACHE_TRANSIENT bool

Is the outer write-through transient.

OUTERCACHE_TYPE enum

Outer Caching scheme (NC/MB/WA).

OUTERCACHE_WA bool

Is the outer cache allocate on write.

PADDR unsigned int

Physical address of the access.

PAGESIZE unsigned int

Page size as log2(size).

PAS enum

PAS supposed to be accessed is secure or nonsecure memory or Root/Realm if RME is implemented.

SH enum

Shareability.

SIDE enum

Inst / Data.

STAGE1_PERM unsigned int

Stage 1 permissions, [5:3] Privileged access (XWR) [2:0] User access (XWR).

STAGE2_PERM unsigned int

Stage 2 permission mask [5:3] Privileged access (XWR) [2:0] User access (XWR).

VADDR unsigned int

Virtual address of the access.

VMID unsigned int

Virtual machine identifier.

XS bool

XS Attribute.

nG enum

Flag indicating whether ASID will be matched.

MMU_TTB_READ

This event is triggered by reads caused by a translation table walk. Fields:

ABORTED bool

Was the walk successful.

DATA unsigned int

The data read.

DESC_KIND enum

The encoding scheme of the descriptors used for this walk.

IPA unsigned int

For stage 1, the IPA of the read.

LEVEL unsigned int

Translation table level.

LPAAE bool

Is this for an LPAE translation (*DEPRECATED*, use DESC_KIND).

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSmem bool

Non secure memory access.

NSreq bool

Non secure request.

OUTADDR unsigned int

The output physical address.

PADDR unsigned int

The physical address of the read.

PASmem enum

Physical Address Space of the accessed memory.

PASreq enum

Physical Address Space of the requested lookup.

REGIME_EL enum

Entry matches in this translation regime.

REQUESTER enum

The requester of this table walk.

SIDE enum

Inst / Data.

STAGE unsigned int

Translation stage.

MMU_TTB_WRITE

This event is triggered by writes caused by a translation table walk. Fields:

ABORTED bool

Was the walk successful.

DATA unsigned int

The data written.

DESC_KIND enum

The encoding scheme of the descriptors used for this walk.

IPA unsigned int

For stage 1, the IPA of the write.

IS_HACDBS_ACCESS bool

If the access was due to HACDBS_CLEAN.

LEVEL unsigned int

Translation table level.

LPAE bool

Is this for an LPAE translation (*DEPRECATED*, use DESC_KIND).

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSmem bool

Non secure memory access.

NSreq bool

Non secure request.

OUTADDR unsigned int

The output physical address.

PADDR unsigned int

The physical address of the write.

PASmem enum

Physical Address Space of the accessed memory.

PASreq enum

Physical Address Space of the requested lookup.

REGIME_EL enum

Entry matches in this translation regime.

SIDE enum

Inst / Data.

STAGE unsigned int

Translation stage.

MODE_CHANGE

Mode change. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

MODE enum

The new mode.

NON_SECURE enum

The core's new non-secure bit.

OLD_MODE enum

The old mode.

SECURITY_STATE enum

The new security state.

NOT_TAG_CHECKED_INST

MTE Tag not checked for current instruction.

PERIODIC

Called for every quantum. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The instruction count of this CPU.

PC unsigned int

The address of the next instruction to be executed on this CPU.

PMU_COUNTER_OVERFLOW

PMU counter overflow. Fields:

COUNTER_GROUP enum

Counter group.

COUNTER_RANGE enum

Counter range.

EVENT_ID unsigned int

Event Identifier, field is valid only for Event counter group.

INDEX unsigned int

Counter index (as selected by PMSELR), field is valid only for Event/Cycle counter groups.

INTERRUPT bool

Is interrupt enabled on overflow for this counter.

POTINDEX_UPDATE

Value of POT index for current pc changed. Fields:

NEW_POTINDEX unsigned int

New POTIndex.

OLD_POTINDEX unsigned int

Old POTIndex.

PREFETCH_MEMORY64

Prefetch from PRFM or PRFUM instructions. Fields:

PRFOP enum

Prefetch hint.

VADDR unsigned int

Virtual address of the location that should be prefetched.

PRELOAD64_REQUEST

Preload requests from PRFM or PRFUM instructions. Fields:

PRFOP enum

Prefetch hint.

VADDR unsigned int

Virtual address of the location that should be prefetched.

PRELOAD_DATA

Data preload from PLD instruction. Fields:

VADDR unsigned int

Virtual address of the data that should be preloaded.

PRELOAD_INST

Instruction preload from PLI instruction. Fields:

VADDR unsigned int

Virtual address of the instruction that should be preloaded.

PRE_CORE_LOAD

Trace just before a core load. Fields:

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

PRE_CORE_STORE

Trace just before a core store. Fields:

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

PRE_TTB_READ

This event is triggered before reads caused by a translation table walk. Fields:

ADDR unsigned int

The physical address of the read.

PSTATE

PSTATE fields update. Fields:

ALLINT unsigned int

All interrupt mask bit.

BTYPE unsigned int

Branch target identification bit.

CORE_NUM unsigned int

Core number in a multi processor.

DAIF unsigned int

Exception mask bits.

DIT unsigned int

Data Independent Timing bit.

EL unsigned int

Current Exception level field.

EXLOCK unsigned int

Exception return state lock bit.

IL unsigned int

Illegal Execution state bit.

NZCV unsigned int

Condition flags.

PACM unsigned int

PACM bit.

PAN unsigned int

Privileged Access Never state bit.

PM unsigned int

PMU exception mask bit.

PPEND unsigned int

PMU exception pending bit.

SM unsigned int

StreamingMode status bit.

SP unsigned int

Stack pointer register selection bit.

SS unsigned int

Software Step bit.

SSBS unsigned int

Speculative Store Bypass Safe bit.

TCO unsigned int

Tag Check Override bit.

TOUCHED unsigned int

Mask of the fields that have been touched since this trace has last been emitted. Note that touched does not necessarily mean the value changed. The index of the field is the bit index in the mask plus the index of this field plus one.

UAO unsigned int

User Access Override bit.

UINJ unsigned int

Undefined Instruction exceptions bit.

ZA unsigned int

ZA storage and ZT0 register enable bit.

nRW unsigned int

Current Execution state bit.

PSTATE_PACTIVE_OPMODE_TRANSITION

Trace pactive and Opmode mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Operating mode after the transition.

OLD_STATE enum

Operating mode before the transition.

PSTATE_PACTIVE_OPMODE_TRANSITION_DENIED

Trace the denied pactive and Opmode mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Operating mode after the transition.

OLD_STATE enum

Operating mode before the transition.

REASON string

Reason for the transition being denied.

PSTATE_PACTIVE_POWER_TRANSITION

Trace pactive and power mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Power mode after the transition.

OLD_STATE enum

Power mode before the transition.

PSTATE_PACTIVE_POWER_TRANSITION_DENIED

Trace the denied pactive and power mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Power mode after the transition.

OLD_STATE enum

Power mode before the transition.

REASON string

Reason for the transition being denied.

PSTATE_PACTIVE_THREAD_OPMODE_TRANSITION

Trace pactive and Opmode mode transitions for threads on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Operating mode after the transition.

OLD_STATE enum

Operating mode before the transition.

QUANTUM_START

Called at the start of every quantum.

RANGE_PRELOAD_REQUEST

Preload requests from RPRFM. Fields:

COUNT unsigned int

Number of blocks of data to access.

LENGTH unsigned int

Number of contiguous bytes to be accessed in each block.

OP unsigned int

Range prefetch operation .

REUSE unsigned int

Maximum number of bytes to be accessed before next RPRFM .

STRIDE signed int

Stride to advance after accessing LENGTH bytes.

VADDR unsigned int

Base virtual address of range prefetch request.

RUN_STATE

Run state transition. Fields:

INST_COUNT unsigned int

Ticks count at point of transition.

NEW enum

New run state.

OLD enum

Old run state.

SBP_SAMPLE_COLLISION

DISPLAY Statistical profiling samples collided .

SBP_SAMPLE_DISCARDED

DISPLAY Operation discarded by statistical profiling unit when configured in discard mode.
Fields:

BUFFER_POINTER unsigned int

Value of buffer pointer.

DATA unsigned int

Data discarded.

SIZE unsigned int

Size of sample, in bytes.

SBP_SAMPLE_END

DISPLAY Statistical profiling of an operation ended.

SBP_SAMPLE_POP

DISPLAY Operation which might be sampled by statistical profiling.

SBP_SAMPLE_RECORDED

DISPLAY Operation sampled by statistical profiling has been written to the buffer at %{BUFFER_ADDRESS} with size %{SIZE} and data %{DATA}. Fields:

BUFFER_ADDRESS unsigned int

Virtual Address of start of buffer write.

DATA unsigned int

Data written to the buffer.

SIZE unsigned int

Size of sample written to buffer, in bytes.

SBP_SAMPLE_START

DISPLAY Statistical profiling of an operation started.

SBP_SAMPLE_TAKEN

DISPLAY Operation sampled by statistical profiling after filtering.

SEMIHOSTING_CALL

Call of a semihost function occurred. Fields:

PC unsigned int

The program counter after the semihosting call.

SEMIHOSTING_PRECALL

About to call semihost. Fields:

EL2 bool

Use EL2 translation regime.

NS bool

Is Non-Secure.

REG_WIDTH unsigned int

The current register width in bytes.

SIGNAL

External signal state change. Fields:

SIGNAL *enum*

Signal that changed.

STATE *bool*

Signal asserted state.

SOFTWARE_STEP

Return the Debug Software Step state. Fields:

STATE *enum*

Software Step state.

SPSR

SPSR change. Fields:

CORE_NUM *unsigned int*

Core number in a multi processor.

MODE *enum*

Which of the banked SPSR registers is written.

OLD_VALUE *unsigned int*

The old SPSR value.

VALUE *unsigned int*

The new SPSR value.

START_COMPILE

Compilation started. Fields:

VADDR *unsigned int*

Instruction where compilation begins.

STG_MULTIPLE_MICRO_OPS_ADDRESS_RANGE

Address range for MTE stores with multiple uops. Fields:

ADDR_END *unsigned int*

Final address.

ADDR_START *unsigned int*

Starting address.

STORE_MULTIPLE_REGS_ADDRESS_RANGE

Address range for stores with multiple registers. Fields:

ADDR_END *unsigned int*

Final address.

ADDR_START *unsigned int*

Starting address.

SUBOPTIMAL_LDST_RETIRED

suboptimal load/store. Fields:

ACCESS_SIZE unsigned int

Log2 of the access width in bytes used for alignment checking (1: half-word, 2: word ...).

OPTIMAL_ALIGN_SELECTOR unsigned int

Selector to decide the optimal alignment size (1: access width, 2: 4-bytes, 3: 8-bytes, ..., 12: 4-kbytes).

VADDR unsigned int

The virtual address of the access.

SYNC

Called for every synchronization, that is every quantum break. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The instruction count of this CPU.

LOCAL_QUANTUM unsigned int

The local quantum of this CPU.

LOCAL_TIME unsigned int

The local time of this CPU.

SYNC_REASON enum

Reason for sync.

SYSCALL

System call instruction executed. Fields:

IMM unsigned int

Immediate value of the system call instruction.

TYPE enum

System call type.

VADDR unsigned int

Instruction that caused the system call.

SYSREG_READ128

System Coprocessor 128 bit register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

REGNUM enum

Internal register number.

REGNUM_PHYS enum

Internal register number physically accessed (e.g. in presense of VHE redirection).

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_READ64

System Coprocessor register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REGNUM enum

Internal register number.

REGNUM_OPERAND enum

Register number corresponding to the instruction operand.

REGNUM_PHYS enum

Internal register number physically accessed (e.g. in presense of VHE redirection).

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_UPDATE128

Triggers when the system updates a register. Fields:

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG enum

Register number.

UNKNOWN unsigned int

Bits of the register which became unknown.

VALUE unsigned int

Value written to the register.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_UPDATE32

Triggers when the system updates a register. Fields:

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG enum

Register number.

UNKNOWN unsigned int

Bits of the register which became unknown.

VALUE unsigned int

Value written to the register.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_UPDATE64

Triggers when the system updates a register. Fields:

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG enum

Register number.

UNKNOWN unsigned int

Bits of the register which became unknown.

VALUE unsigned int

Value written to the register.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_WRITE128

System Coprocessor 128 bit register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

REGNUM enum

Internal register number.

REGNUM_PHYS enum

Internal register number physically accessed (e.g. in presense of VHE redirection).

UNDEF bool

The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_WRITE64

System Coprocessor register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REGNUM enum

Internal register number.

REGNUM_OPERAND enum

Register number corresponding to the instruction operand.

REGNUM_PHYS enum

Internal register number physically accessed (e.g. in presense of VHE redirection).

UNDEF bool

The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

UNALIGNED_LDST_RETIRE

Processor unaligned load/store. Fields:

ACCESS_SIZE unsigned int

Log2 of the access width in bytes used for alignment checking (1=>half-word, 2->word ...).

NUMBER_OF_BEATS unsigned int

Number of accesses (beats) in this burst (Total size in bytes is ACCESS_SIZE * NUMBER_OF_BEATS).

VADDR unsigned int

The virtual address of the access.

VFP_D_REGS

VFP/NEON D 64 bit register write. Fields:

ALIASING enum

Type of register aliasing used.

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a write to the corresponding bit in the register. For example, 0x1 would indicate a write to VALUE[0].

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

VFP_D_REGS_READ

VFP/NEON D 64 bit register read. Fields:

ID unsigned int

The register number.

VALUE unsigned int

The current value of the register.

VFP_Q_REGS

VFP/NEON Q 128 bit register write. Fields:

ALIASING enum

Type of register aliasing used.

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a write to the corresponding byte in the register. For example, 0x1 would indicate a write to VALUE[7:0].

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

VFP_Q_REGS_READ

VFP/NEON Q 128 bit register read. Fields:

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a read to the corresponding byte in the register. For example, 0x1 would indicate a read to VALUE[7:0].

VALUE unsigned int

The current value of the register.

VFP_SYS_REGS

Writes to the VFP/NEON units system registers. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ID enum

Which VFP system register is written.

MASK unsigned int

The mask indicating updated cumulative bits.

OLD_VALUE unsigned int

The register's old value overwritten.

VALUE unsigned int

The new value written to the VFP system register.

VFP_S_REGS

VFP/NEON S 32 bit register write. Fields:

ALIASING enum

Type of register aliasing used.

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a write to the corresponding bit in the register. For example, 0x1 would indicate a write to VALUE[0].

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

VFP_S_REGS_READ

VFP/NEON S 32 bit register read. Fields:

ID unsigned int

The register number.

VALUE unsigned int

The current value of the register.

VIRTUAL_TAG_ALLOCATED

Virtual tag information when allocated. Fields:

TAG_VADDR unsigned int

Virtual tag virtual address.

VADDR unsigned int

Data Virtual address.

VIRTUAL_TAG unsigned int

Virtual tag value.

WAYPOINT

Signals end of basic block execution. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ISET enum

Origin instruction set.

IS_COND bool

Indicates if this is a conditional waypoint.

PC unsigned int

Origin address (or 0 if unavailable).

TAKEN bool

Indicates if this waypoint was taken.

TARGET unsigned int

Destination address.

TARGET_ISET enum

Destination instruction set.

WFE_END

WFE ended. Fields:

INST_COUNT unsigned int

Ticks count when leaving WFE.

WFE_EVENT_REGISTER

WFE event register status: set/clear, reason. Fields:

INST_COUNT unsigned int

Ticks count.

REASON enum

Reason for set/clear. Only REASON==0 (Cleared by WFE) clears the bit, all other reasons set it.

WFE_IGNORED

WFE ignored. Fields:

EVENT bool

This WFE was ignored because the event register was set.

INST_COUNT unsigned int

Ticks count when ignoring WFE.

TRAPPED bool

This WFE was trapped.

WFE_START

WFE entered. Fields:

INST_COUNT unsigned int

Ticks count when entering WFE.

WFI_END

WFI ended. Fields:

INST_COUNT unsigned int

Ticks count when leaving WFI.

WFI_IGNORED

WFI ignored. Fields:

DISABLED bool

This WFI was ignored because WFI is disabled.

INST_COUNT unsigned int

Ticks count when ignoring WFI.

REASON enum

specifies reason why WFI trace was ignored.

TRAPPED bool

This WFI was trapped.

WFI_START

WFI entered. Fields:

INST_COUNT unsigned int

Ticks count when entering WFI.

WFI_WAKEUP

WFI wakeup. Fields:

INST_COUNT unsigned int

Ticks count when WFI wakeup occurred.

REASON enum

Reason for wakeup.

2.57 ARM_Cortex-X1C

This section describes the trace sources.

AA64_ASE_SVE_REGS

Modification of AArch64 Advanced SIMD or SVE Vector Registers (via SIMD or Scalar operations), SVE Predicate Registers and ZTO. Fields:

ID enum

The register identifier.

MASK unsigned int

Bitmask of the register to signal the modified bits in the VALUE field.

SM bool

Whether the PE is in Streaming Mode.

VALUE unsigned int

Value of AArch64 vector register.

ASYNC_MEMORY_FAULT

Context ID Register write. Fields:

FAULT unsigned int

Fault status present in ESR.ISS encoding for Serror in ARCH64 and present in DFSR encoding for external abort format in ARCH32.

PADDR unsigned int

Physical Address (or 0 if unavailable).

VADDR signed int

Virtual Address (or 0 if unavailable).

ATOMIC_END_ACCESS

Bus trace access for atomics. Fields:

ACCESS_FAIL bool

Memory access failed.

ADDR unsigned int

The virtual address of the access.

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

COMPARE_VALUE unsigned int

Compare value for CAS.

LOAD_VALUE unsigned int

Loaded values.

NSDESC unsigned int

The security state of the access.

OPERAND_VALUE unsigned int

Operation's operand.

OPERATION enum

Operation type.

PADDR **unsigned int**

The physical address of the access.

PRIV **bool**

Is this a privileged access?.

ATOMIC_START_ACCESS

Bus trace access for atomics. Fields:

ADDR **unsigned int**

The virtual address of the access.

ATTR **unsigned int**

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

COMPARE_VALUE **unsigned int**

Compare value for CAS.

NSDESC **unsigned int**

The security state of the access.

OPERAND_VALUE **unsigned int**

Operation's operand.

OPERATION **enum**

Operation type.

PADDR **unsigned int**

The physical address of the access.

PRIV **bool**

Is this a privileged access?.

ArchMsg.Error.exit_code

DISPLAY exit code: %{EXITCODE} component kind: %{KIND} reason: %{REASON}. Fields:

EXITCODE **signed int**

Exit code to be returned.

KIND **string**

Component kind that invoked the exit code trace.

REASON **string**

Reason why the exit code trace was invoked (optional).

ArchMsg.Warning.IgnoredWarmResetRequest

Event fires when software attempts to request warm reset through write to deprecated DBGPRCR.CWRR.

ArchMsg.Warning.RAS_ErrselrUnpredictableState

DISPLAY ERRSELR is in unpredictable state due to value greater than number_of_error_records written to it.

ArchMsg.Warning.RAS_UnpredictableStateWrite

DISPLAY %{REG_NAME} write ignored due to unpredictable ERRSELR state. Fields:

REG_NAME string

Name of register being written.

ArchMsg.Warning.ReportedRevisionRegisterRead

DISPLAY Reported revision and patch do not match that of the cpu. Fields:

ORIGINAL_VALUE unsigned int

The value the register would have based on the modelled revision.

REGISTER string

The name of the register that has ben read.

REPORTED_VALUE unsigned int

The value the register has been overridden to.

ArchMsg.Warning.Unpredictably Indexed PM Event Register

TAGS **UNPREDICTABLE** DISPLAY Event %{IsDirect:(Selector value|Register)} %{SEL} is >= %{N}. Fields:

IsDirect unsigned int

Direct Access rather than indirect.

N unsigned int

Number of Accessible Registers.

SEL unsigned int

Selector.

ArchMsg.Warning.cache_contents_unknown

A cache has been activated without being invalidated and may contain **UNKNOWN** content. DISPLAY %{SIDE:(D|I)}-Cache was enabled in %{REGIME:(SECURE|NON_SECURE|HYP)} regime but was not invalidated since power-on: cache lines could contain **UNKNOWN** content %{SIDE: (Note: D-side caches must invalidate SECURE lines even if the cache is only used by NON_SECURE code in order to ensure that unexpected dirty lines tagged SECURE are not naturally evicted)}. Fields:

REGIME unsigned int

Current translation regime.

SIDE unsigned int

0 d-side cache or unified cache, 1 i-side.

ArchMsg.Warning.dcimvac_matches_watchpoint

TAGS **UNPREDICTABLE** DISPLAY Watchpoints matching an AArch32 DCIMVA are implementation defined.

ArchMsg.Warning.decode_fieldmismatch

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with different fields %{FIELD1} and %{FIELD2}. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_fracbitsnegative

TAGS **UNPREDICTABLE** DISPLAY VCVT Instruction is unpredictable with a negative number of fraction bits.

ArchMsg.Warning.decode_initblock

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable in an IT block.

ArchMsg.Warning.decode_initblocknotlast

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable in an IT block when not the last.

ArchMsg.Warning.decode_invalid_condition

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable when the condition is not AL.

ArchMsg.Warning.decode_invalidfieldcombination

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with the values in %{FIELD1} and %{FIELD2}. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_invalidvalue

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with the value in %{FIELD}. Fields:

FIELD enum

Field name.

ArchMsg.Warning.decode_mem_hint_unallocated

TAGS **UNPREDICTABLE** DISPLAY There is no memory hint allocated to this bit pattern.

ArchMsg.Warning.decode_registermatch

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with %{FIELD1} and %{FIELD2} as the same register. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_registermismatch

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with %{FIELD1} and %{FIELD2} as different registers. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_registeroutofrange

TAGS **UNPREDICTABLE** DISPLAY Floating-point instruction is unpredictable indexing beyond the end of the register bank.

ArchMsg.Warning.decode_sbzsbo

TAGS **UNPREDICTABLE** DISPLAY Reserved bits in the instruction are not canonical.

ArchMsg.Warning.decode_transactiontoobig

TAGS **UNPREDICTABLE** DISPLAY Attempt to perform a memory transaction of more than 128 bytes.

ArchMsg.Warning.decode_unpred_other

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable.

ArchMsg.Warning.decode_unpreduseofpc

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable when %{FIELD} is PC. Fields:

FIELD enum

Field name.

ArchMsg.Warning.decode_unpreduseofr13

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable when %{FIELD} is R13. Fields:

FIELD enum

Field name.

ArchMsg.Warning.decode_unpreduseofr13inreglist

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with R13 in register-list.

ArchMsg.Warning.decode_useofd16orhigher

TAGS **UNPREDICTABLE** DISPLAY Attempt to use register D16 or higher where it is not allowed.

ArchMsg.Warning.decode_writebackandbaseinlist

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with write back when the base register is in the transfer list.

ArchMsg.Warning.decode_zeroregistersinlist

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable when no registers are to be transferred.

ArchMsg.Warning.recursive_branch

An instruction is performing a branch that targets the same instruction. If this was intended then a WFI might be more effective. DISPLAY recursive branch detected at PC=%{PC}.

PRIMARY KEY PC. Fields:

PC unsigned int

Address of instruction causing the branch.

ArchMsg.Warning.recursive_exception

An instruction has generated an exception that targets the same instruction. DISPLAY recursive %{TYPE} exception detected at PC=%{PC} (CPSR=%{CPSR}). PRIMARY KEY PC. Fields:

CPSR unsigned int

Processor state.

PC unsigned int

Address of instruction causing the exception.

TYPE enum

Type of exception.

ArchMsg.Warning.reentrant_vector_catch

TAGS **UNPREDICTABLE** DISPLAY Vector catch debug event on Prefetch or Data abort vector %{ADDR}. Fields:

ADDR unsigned int

vector address.

ArchMsg.Warning.reserved_it_state

TAGS **UNPREDICTABLE** DISPLAY Execution with reserved IT state %{ITSTATE}. Fields:

ITSTATE unsigned int

reserved IT state.

ArchMsg.Warning.tlb_contents_unknown

Translation has been enabled without invalidating the TLB. DISPLAY Translation was enabled in the %{PAS:(Secure|Non-secure|Root|Realm)} EL%{EL:d} translation regime (vmid=%{VMID:x}) without invalidating %{INVALIDITY:(|d-side |i-side |)}entries since power-on: %{INVALIDITY:(|D||I)}TLB could contain **UNKNOWN** content. Fields:

EL unsigned int

Exception level of translation regime.

INVALIDITY unsigned int

Which TLB may be invalid (if TLBs are separate).

NON_SECURE bool

Whether the translation regime is non-secure.

PAS unsigned int

Physical address space of translation regime.

VMID unsigned int

Current VMID, if applicable.

ArchMsg.Warning.unknown_DLR_DSPSR

Exiting debug state with unknown DLR or DSPSR.

ArchMsg.Warning.unknown_ELR_SPSR

Returning from debug exception with unknown ELR or SPSR.

ArchMsg.Warning.unpred_FAR_on_LRCPC3_non_atomic_mem_access

TAGS **UNPREDICTABLE** Unpredictable FAR/FSTATUS on LRCPC3 pair instructions resulting in two separate memory accesses
DISPLAY Unpredictable FAR on LRCPC3 non atomic memory access at %{ADDR1} with FSTATUS %{FSTATUS1} or %{ADDR2} with FSTATUS %{FSTATUS2}. Fields:

ADDR1 unsigned int

Possible FAR address 1.

ADDR2 unsigned int

Possible FAR address 2.

FSTATUS1 unsigned int

Possible fault status value 1.

FSTATUS2 unsigned int

Possible fault status value 2.

ArchMsg.Warning.unpred_async_tag_check_fail_on_sve_speculative_load_do_update_ffr

Unpredictable set to FALSE of the FFR element associated to SVE speculative load with an async tag check fail TAGS **UNPREDICTABLE**.

ArchMsg.Warning.unpredictable_a32_breakpoint

TAGS **UNPREDICTABLE** DISPLAY Unpredictable breakpoint on a A32 instruction at %{ADDR1} Offset %{OFFSET} BAS %{BAS} for a %{IS_ADDRESS_MISMATCH_BREAKPOINT:(Mismatch|Match)} Breakpoint number %{N:d} Breakpoint address %{ADDR2}. Fields:

ADDR1 unsigned int

Address.

ADDR2 unsigned int

Breakpoint address.

BAS unsigned int

BAS field.

IS_ADDRESS_MISMATCH_BREAKPOINT bool

Is a Mismatch breakpoint, not a Match breakpoint.

N unsigned int

Breakpoint number.

OFFSET unsigned int

Always has a value of zero.

ArchMsg.Warning.unpredictable_banked_register_access

TAGS **UNPREDICTABLE** DISPLAY Access to banked register (R=%{R:d} SYSm=%{SYSM:2x}) is unpredictable. Fields:

R unsigned int

R bit.

SYSM unsigned int

SYSm.

ArchMsg.Warning.unpredictable_ccfail_undef

TAGS **UNPREDICTABLE** DISPLAY An instruction %{OPCODE} at VA=%{VADDR} failed its condition codes check but would otherwise have caused a non-data-dependent trap or undefined exception. Behaviour is implementation-defined choice of exception or nop. Fields:

OPCODE unsigned int

opcode of the instruction.

VADDR unsigned int

Virtual address of the instruction being executed.

ArchMsg.Warning.unpredictable_ici_bits

TAGS **UNPREDICTABLE** DISPLAY ICI bits do not apply or are architecturally unpredictable for instruction.

ArchMsg.Warning.unpredictable_t32_breakpoint

TAGS **UNPREDICTABLE** DISPLAY Unpredictable breakpoint on a T32 instruction at %{ADDR1} Offset %{OFFSET} BAS %{BAS} for a %{IS_ADDRESS_MISMATCH_BREAKPOINT:(Mismatch|Match)} Breakpoint number %{N:d} Breakpoint address %{ADDR2}. Fields:

ADDR1 unsigned int

Address.

ADDR2 unsigned int

Breakpoint address.

BAS unsigned int

BAS field.

IS_ADDRESS_MISMATCH_BREAKPOINT bool

Is a Mismatch breakpoint, not a Match breakpoint.

N unsigned int

Breakpoint number.

OFFSET unsigned int

Indicates which halfword of a 32-bit Thumb instruction the breakpoint is set on. 0 for the first halfword or if the instruction is 16 bits wide, or 1 for the second halfword.

ArchMsg.Warning.unpredictable_t32_vector_catch

TAGS **UNPREDICTABLE** DISPLAY Unpredictable Vector catch exception on a T32 instruction at %{ADDR} Offset %{OFFSET}. Fields:

ADDR unsigned int

Address.

OFFSET unsigned int

Indicates which halfword of a 32-bit Thumb instruction the vector catch exception is set on. 0 for the first halfword or if the instruction is 16 bits wide, or 1 for the second halfword.

ArchMsg.Warning.unpredictable_watchpoint_far

TAGS **UNPREDICTABLE** DISPLAY The value for FAR/EDWAR for hitting this watchpoint may be anywhere in the range %{LOWER_BOUND} to %{UPPER_BOUND}. Fields:

LOWER_BOUND unsigned int

the lowest address accessed by the instruction that triggered the watchpoint.

UPPER_BOUND unsigned int

the highest watchpointed address accessed by that instruction.

ArchMsg.Warning.unpredictable_watchpoint_far_isscmp

DISPLAY The value for FAR/EDWAR for hitting this watchpoint may be anywhere in the range %{LOWER_BOUND} to %{UPPER_BOUND}. Fields:

LOWER_BOUND unsigned int

the lowest address accessed by the instruction that triggered the watchpoint.

UPPER_BOUND unsigned int

the highest watchpointed address accessed by that instruction.

ArchMsg.Warning.warning_AdvSIMDExpandImmUnexpectedZero

TAGS **UNPREDICTABLE** DISPLAY AdvSIMDExpandImm may treat this immediate value as **UNPREDICTABLE**.

ArchMsg.Warning.warning_ConditionalSMC

TAGS **UNPREDICTABLE** DISPLAY SMC instruction has **UNPREDICTABLE** effects when conditional and when combined with traps.

ArchMsg.Warning.warning_TLBID_out_of_range

TLBID out of range in TLBI according to the configuration of NIS, NVIS, NOS and NVOS. Fields:

NIS unsigned int

NIS.

NOS unsigned int

NOS.

NVIS unsigned int

NVIS.

NVOS unsigned int

NVOS.

TLBID unsigned int

The TLBID value used in the TLBI instruction.

ArchMsg.Warning.warning_TTL_or_TG_mismatches_current_translation_granule

TTL or TG mismatches current translation granule in TLBI VA/RVA/IPA/RIPA. Fields:

CURRENT_TG unsigned int

Current TG as indicated by TCR_ELx or VTCR_EL2.

TLBI_TG unsigned int

TG (or TTL[3:2]) used in TLBI instruction.

ArchMsg.Warning.warning_access_crosses_page_boundary

TAGS **UNPREDICTABLE** DISPLAY Access crosses page boundary.

ArchMsg.Warning.warning_access_crosses_page_boundary_has_fault

TAGS **UNPREDICTABLE** DISPLAY Access that crosses page boundary has fault(s).

ArchMsg.Warning.warning_access_crosses_page_boundary_has_fault_on_both_pages

TAGS **UNPREDICTABLE** DISPLAY Access that crosses page boundary has a fault on each page.
Fields:

ADDR1 unsigned int

FAR address on first page.

ADDR2 unsigned int

FAR address on second page.

FSTATUS1 unsigned int

Fault status on first page.

FSTATUS2 unsigned int

Fault status on second page.

ArchMsg.Warning.warning_access_crosses_page_boundary_spanning_different_memory

TAGS **UNPREDICTABLE** DISPLAY Access at address %{ADDR} crosses page from MEM_TYPE %{MEMTYPE_PAGE1} to %{MEMTYPE_PAGE2}. Fields:

ADDR unsigned int

address of access crossing page.

MEMTYPE_PAGE1 signed int

First Page's memory type (0-Normal, 1-Non-Normal).

MEMTYPE_PAGE2 signed int

Second Page's memory type (0-Normal, 1-Non-Normal).

ArchMsg.Warning.warning_access_wraps_around_memory

TAGS **UNPREDICTABLE** DISPLAY Access wraps around memory in %{WIDTH:d}bit mode.
Fields:

WIDTH unsigned int

Address width.

ArchMsg.Warning.warning_bcr_linking_status

TAGS **UNPREDICTABLE** DISPLAY DBGBCR.{BT2,BT} combination gives unpredictable behaviour.

ArchMsg.Warning.warning_bcr_mask_reserved

TAGS **UNPREDICTABLE** DISPLAY DBGBCR_ELn.MASK is set to a reserved value.

ArchMsg.Warning.warning_bvr_masked_bit_not_zero

TAGS **UNPREDICTABLE** DISPLAY DBGBVRn_EL1.VA contains a bit masked by DBGBCRn_EL1.MASK that is not zero.

ArchMsg.Warning.warning_ccsidr_unimplemented_level

TAGS **UNPREDICTABLE** DISPLAY CCSIDR read while CSSELR %{CSSELR:x} points to an unimplemented cache level. Fields:

CSSELR unsigned int

current effective value of CSSELR.

ArchMsg.Warning.warning_change_to_ns_when_tge_set

TAGS **UNPREDICTABLE** DISPLAY Attempting to change NS to 1 when HCR.TGE = 1.

ArchMsg.Warning.warning_contiguous_bit_check_abort

TAGS **UNPREDICTABLE** DISPLAY An abort occurred whilst attempting to check TLB entry is contiguous at %{ENTRY_ADDR}. Fields:

ENTRY_ADDR unsigned int

address of conflicting TLB entry.

ArchMsg.Warning.warning_contiguous_bit_error

TAGS **UNPREDICTABLE** DISPLAY TLB entry at %{ENTRY_ADDR} was not contiguous with entry at %{CONTIG_ADDR} Contents are %{ENTRY_DATA} but expected %{CONTIG_DATA}. Fields:

CONTIG_ADDR unsigned int

address of the TLB first read, with which this entry is expected to be contiguous.

CONTIG_DATA unsigned int

expected contents based on the entry at CONTIG_ADDR.

ENTRY_ADDR unsigned int

address of conflicting TLB entry.

ENTRY_DATA unsigned int

contents of conflicting TLB entry.

ArchMsg.Warning.warning_contiguous_bit_error_D128

TAGS **UNPREDICTABLE** DISPLAY TLB entry at %{ENTRY_ADDR} was not contiguous with entry at %{CONTIG_ADDR} Contents are %{ENTRY_DATA_HI} %{ENTRY_DATA_LO} but expected %{CONTIG_DATA}. Fields:

CONTIG_ADDR unsigned int

address of the TLB first read, with which this entry is expected to be contiguous.

CONTIG_DATA unsigned int

expected contents based on the entry at CONTIG_ADDR.

ENTRY_ADDR unsigned int

address of conflicting TLB entry.

ENTRY_DATA_HI unsigned int

Higher bit contents of conflicting TLB entry.

ENTRY_DATA_LO unsigned int

Lower bit contents of conflicting TLB entry.

ArchMsg.Warning.warning_cp10_cp11_mismatch

TAGS **UNPREDICTABLE** DISPLAY Unpredictable configuration of CP10 and CP11 controls at EL
%{EL:d}. Fields:

EL unsigned int

EL owning the control.

ArchMsg.Warning.warning_cp10_cp11_reserved_value

TAGS **UNPREDICTABLE** DISPLAY Unpredictable configuration of CPACR CP10 (%{CP10}) or
CP11 (%{CP11}). Fields:

CP10 unsigned int

CP10 access.

CP11 unsigned int

CP11 access.

ArchMsg.Warning.warning_csselr_level_out_of_range

TAGS **UNPREDICTABLE** DISPLAY CSSELR.Level out of range, written value %{LEVEL} with only
%{IMPLEMENTED:d} levels implemented. Fields:

IMPLEMENTED unsigned int

number of cache levels implemented.

LEVEL unsigned int

written value of CSSELR.Level.

TYPE enum

The type of cache selected by CSSELR.InD and TnD.

ArchMsg.Warning.warning_debug_flow_control_bits_not_obeyed

TAGS **UNPREDICTABLE** DISPLAY Flow control bits for %{REGISTER} were not obeyed, causing
%{ERROR}. Fields:

ERROR enum

type of error resulting.

REGISTER enum

register under consideration.

ArchMsg.Warning.warning_debug_register_access_during_reset

TAGS **UNPREDICTABLE** DISPLAY Unpredictable %{IS_WRITE:(read|write)} access to debug register offset %{OFFSET} during reset. Fields:

IS_WRITE unsigned int

Write Not Read.

OFFSET unsigned int

Register Offset.

ArchMsg.Warning.warning_decode_cps_inconsistent_fields

TAGS **UNPREDICTABLE** DISPLAY The mode setting fields in the CPS instruction are inconsistent.

ArchMsg.Warning.warning_decode_invalid_state

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable in current exception-level/security state.

ArchMsg.Warning.warning_default_cacheable_mmu_on

TAGS **UNPREDICTABLE** DISPLAY Default cacheable is enabled (HCR.DC = 1) while MMU is enabled.

ArchMsg.Warning.warning_default_cacheable_vmmu_off

TAGS **UNPREDICTABLE** DISPLAY Default cacheable is enabled (HCR.DC = 1) while HCR.VM = 0.

ArchMsg.Warning.warning_deprecated_wvr_bit_2_set

TAGS **UNPREDICTABLE** DISPLAY DBGWVR%{N:d}_EL1=%{WVR:h} has bit 2 set, deprecated in ARMv8. Fields:

N unsigned int

index.

WVR unsigned int

DBGWVR.

ArchMsg.Warning.warning_eret_while_software_step_active_pending

Missing ISB between setting MDSCR_EL1.SS and ERET TAGS **UNPREDICTABLE**.

ArchMsg.Warning.warning_exclusive_to_non_normal

TAGS **UNPREDICTABLE** DISPLAY Exclusive Access to Non-Normal Memory %{ADDR}. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_exclusive_to_non_writeback

TAGS **UNPREDICTABLE** DISPLAY Exclusive Access to Non-Writeback-Cacheable Memory %{ADDR}. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_execute_from_device_memory

TAGS **UNPREDICTABLE** DISPLAY An attempt was made to execute from Device memory.

ArchMsg.Warning.warning_illegal_cpsr_mode

TAGS **UNPREDICTABLE** DISPLAY Writing an illegal or unimplement mode ({NEW_MODE:h}) to CPSR was unpredictable. Fields:

NEW_MODE unsigned int

New value of CPSR.M.

ArchMsg.Warning.warning_illegal_srs_mode

TAGS **UNPREDICTABLE** DISPLAY Illegal or **UNPREDICTABLE** mode ({MODE:h}) used for SRS instruction. Fields:

MODE unsigned int

mode for Banked SP.

ArchMsg.Warning.warning_implementation_defined_read_debug_register_in_SCS

Read Access to Debug registers from the processor is **IMPLEMENTATION DEFINED** TAGS **IMP_DEF**. Fields:

PPB_OFFSET unsigned int

debug register offset.

ArchMsg.Warning.warning_implementation_defined_sequential_security_transitions_supported

The behavior of sequential instruction fetches that cross from non-secure to secure memory and contain an SG instruction is **CONSTRAINED UNPREDICTABLE** TAGS **UNPREDICTABLE**.

ArchMsg.Warning.warning_implementation_defined_stack_limit_check_supported

It is **IMPLEMENTATION DEFINED** whether stack pointer limit checking is performed for this instruction TAGS **IMP_DEF**. Fields:

ADDRESS unsigned int

Address of instruction.

OPCODE unsigned int

instruction opcode.

ArchMsg.Warning.warning_implementation_defined_taggednonsharedmemorytype

TAGS **IMP_DEF** DISPLAY Memory attribute has Tagged, non shared memory type; it is **IMPLEMENTATION DEFINED** whether the memory location is treated as Tagged or non-Tagged.

ArchMsg.Warning.warning_implementation_defined_write_debug_register_in_SCS

Write Access to Debug registers from the processor is **IMPLEMENTATION DEFINED** TAGS **IMP_DEF**. Fields:

DATA unsigned int

data attempted to be written.

PPB_OFFSET unsigned int

debug register offset.

ArchMsg.Warning.warning_invalid_tcr_granule

TAGS UNPREDICTABLE DISPLAY TCR.TG{%TG_ID:(0|1)} bits have been set to a granule size {%REQUEST:d}K not implemented - using {%SUBSTITUTE:d}K. Fields:

REQUEST signed int

page size requested (or 0 for reserved).

SUBSTITUTE signed int

best guess available page size.

TG_ID bool

bits TG0 or TG1.

ArchMsg.Warning.warning_load_multiple_user_registers_from_user_mode

TAGS UNPREDICTABLE DISPLAY An LDM(user registers) instruction executed from user mode.

ArchMsg.Warning.warning_mte_in_mem_attr_not_supported

TAGS UNPREDICTABLE DISPLAY Memory attribute {%ATTR} has Tagged memory type but Armv8.5-MTE is not implemented. Fields:

ATTR unsigned int

Memory attribute.

ArchMsg.Warning.warning_reserved_breakpoint_state_match

TAGS UNPREDICTABLE DISPLAY {%IS_BREAKPOINT:(Watchpoint|Breakpoint)} programmed with a reserved combination of HMC, SSC and {%IS_BREAKPOINT:(PAC|PMC)}. Fields:

IS_BREAKPOINT unsigned int

Is a breakpoint, not a watchpoint.

ArchMsg.Warning.warning_shareability

TAGS UNPREDICTABLE DISPLAY Unpredictable: combination of the 1st and 2nd stages of translation is Normal Inner Non-Cacheable, Outer Non-Cacheable.

ArchMsg.Warning.warning_software_step_set_while_enabled

TAGS UNPREDICTABLE DISPLAY MDSCR_EL1.SS set to 1 while software step debug exceptions are enabled.

ArchMsg.Warning.warning_thumb_instruction_wraps_around_memory

TAGS UNPREDICTABLE DISPLAY Thumb instruction wraps around memory.

ArchMsg.Warning.warning_ttbr_sbz_bits_are_not_zero

TAGS UNPREDICTABLE DISPLAY {%TTBR}[{%MSB:d}:0] should be zero, but are not. Fields:

MSB unsigned int

x-1.

TTBR enum

which TTBR register.

ArchMsg.Warning.warning_unaligned_address_dbgdtrrx_write

TAGS **UNPREDICTABLE** DISPLAY Unaligned address %{ADDR} in X0/R0 when DBGDTRRX written in memory access mode. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unaligned_address_dbgdtrtx_read

TAGS **UNPREDICTABLE** DISPLAY Unaligned address %{ADDR} in X0/R0 when DBGDTRTX read in memory access mode. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unaligned_to_device

TAGS **UNPREDICTABLE** DISPLAY Unaligned Access to Device Memory %{ADDR}. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unaligned_to_strongly_ordered

TAGS **UNPREDICTABLE** DISPLAY Unaligned Access to Strongly Ordered Memory %{ADDR}. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unknown_sau_rnr

TAGS **UNPREDICTABLE** DISPLAY SAU_RNR was set to an unsupported value.

ArchMsg.Warning.warning_unpredictable_AIRCR_PRIS_and_BFHFNMINs

TAGS **UNPREDICTABLE** DISPLAY The effect of setting both AIRCR.BFHFNMINs and AIRCR.PRIS to 1 is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_AIRCR_VECTCLRACTIVE_when_not_in_debug

TAGS **UNPREDICTABLE** DISPLAY The effect of writing a 1 to AIRCR.VECTCLRACTIVE if the processor is not halted in Debug state is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_AIRCR_VECTRESET_and_SYSRESETREQ

TAGS **UNPREDICTABLE** DISPLAY When the processor is halted in Debug state, if a write to the register writes a 1 to both VECTRESET and SYSRESETREQ, the behavior is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_AIRCR_VECTRESET_when_not_in_debug

TAGS **UNPREDICTABLE** DISPLAY The effect of writing a 1 to AIRCR.VECTRESET if the processor is not halted in Debug state is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_AIRCR_incorrect_VKEY

TAGS **UNPREDICTABLE** DISPLAY The value 0x05FA must be written to AIRCR.VKEY, otherwise the register write is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_EXC_RETURN_Reserved_Bit

EXC_RETURN[23:7] are reserved with the special condition that all bits should be written as one. Values other than all 1s are **UNPREDICTABLE**. TAGS **UNPREDICTABLE**. Fields:

EXC_RETURN unsigned int

Special PC value.

SBO_MASK unsigned int

Which bits Should Be One to not UNPRED.

ArchMsg.Warning.warning_unpredictable_PMEVTYPEPn_ELO_VS_reserved_value

TAGS **UNPREDICTABLE** DISPLAY Unpredictable reserved value of PMEVTYPEn<n>_ELO.VS.

ArchMsg.Warning.warning_unpredictable_change_to_DEMCR_MON_STEP_at_insufficient_priority

TAGS **UNPREDICTABLE** DISPLAY The effect of changing DEMCR.MON_STEP at an execution priority that is lower than the priority of the DebugMonitor exception is **CONSTRAINED UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_change_to_priority_of_active_exception

TAGS **UNPREDICTABLE** DISPLAY Changing the priority of an active exception is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_clear_lspact

TAGS **UNPREDICTABLE** DISPLAY **UNPREDICTABLE** state on context switch. Fields:

DRVEXC_TAKEN unsigned int

Derived exception taken, not pended.

REACHED_LAST_STORE unsigned int

Only the last faulted.

ArchMsg.Warning.warning_unpredictable_dbg_exit_unaligned_dlr

TAGS **UNPREDICTABLE** DISPLAY Debug exit to AArch32 with DLR[0] set to 1. PC[0] can be set to 0 or to DLR[0].

ArchMsg.Warning.warning_unpredictable_exception_catch

TAGS **UNPREDICTABLE** DISPLAY Generation of an exception catch event is unpredictable.

ArchMsg.Warning.warning_unpredictable_exception_return_inconsistent_state

TAGS **UNPREDICTABLE** DISPLAY state on exception return is unpredictable.

ArchMsg.Warning.warning_unpredictable_exception_return_instruction

TAGS **UNPREDICTABLE** DISPLAY exception return instruction is unpredictable.

ArchMsg.Warning.warning_unpredictable_in_debug_state

TAGS **UNPREDICTABLE** DISPLAY Instruction is **UNPREDICTABLE** when executed in debug state.

ArchMsg.Warning.warning_unpredictable_mair_encoding

TAGS **UNPREDICTABLE** DISPLAY Memory attribute %{ATTR} has invalid MAIR encoding; this may lead to unpredictable results. Fields:

ATTR unsigned int

Memory attribute.

ArchMsg.Warning.warning_unpredictable_pmu_counter_access

TAGS **UNPREDICTABLE** DISPLAY Access to PMU counters from non-secure EL0 or EL1 is **UNPREDICTABLE** with MDCR_EL2.HPMN set to 0.

ArchMsg.Warning.warning_unpredictable_prioritization_breakpoint_match_vector_catch

TAGS **UNPREDICTABLE** DISPLAY Two events with the same priority occurred at the same instruction: Address Matching Vector Catch debug event, and Breakpoint debug event. It is constrained unpredictable which is taken.

ArchMsg.Warning.warning_unpredictable_stack_selection

TAGS **UNPREDICTABLE** DISPLAY **UNPREDICTABLE** state on context switch.

ArchMsg.Warning.warning_unpredictable_tsize_out_of_range

TAGS **UNPREDICTABLE** DISPLAY %{TCR}.T%{TTBR:d}SZ (%{TSIZE:d} bits) is out of range. Must be between 25 and %{TSIZE_MAX:d}. Fields:

TCR enum

Which TCR.

TSIZE signed int

Value of TSize.

TSIZE_MAX signed int

The maximum allowed value.

TTBR unsigned int

Which TTBR region.

ArchMsg.Warning.warning_unpredictable_unaligned_pc_as_base_register

TAGS **UNPREDICTABLE** DISPLAY The PC value must be word-aligned, otherwise the behavior of the instruction is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_unaligned_pop_stack

TAGS **UNPREDICTABLE** DISPLAY **CONSTRAINED UNPREDICTABLE** unaligned stack frame %{FRAME_ADDR:x} in exception-return. Fields:

FRAME_ADDR unsigned int

unaligned stack frame address.

ArchMsg.Warning.warning_unpredictable_unaligned_to_non_writeback

TAGS **UNPREDICTABLE** DISPLAY Unaligned Access to Non-Writeback-Cacheable Memory.

ArchMsg.Warning.warning_unpredictable_vmsa_memattrib

TAGS **UNPREDICTABLE** DISPLAY Unpredictable vmsa memory attribute with texcb value programmed to %{TEXCB} when tex remap is %{TEX_REMAP}. Fields:

TEXCB unsigned int

texcb value.

TEX_REMAP bool

use tex remap.

ArchMsg.Warning.warning_unpredictable_vtcr_t0sz_sl0

TAGS **UNPREDICTABLE** DISPLAY The combination of VTCR.TOSZ=%{TOSZ:d} and VTCR.SLO=%{SLO:d} is unpredictable. Fields:

SLO signed int

VTCR.SLO, starting level for stage 2 translation table walks.

TOSZ signed int

VTCR.TOSZ, required input address range.

ArchMsg.Warning.warning_unpredictable_write_DHCSR

TAGS **UNPREDICTABLE** DISPLAY **UNPREDICTABLE** set of bit-changes in DHCSR. Fields:

NEW unsigned int

data attempted to be written.

OLD unsigned int

previous value.

ArchMsg.Warning.warning_unpredictable_write_SHCSR

TAGS **UNPREDICTABLE** DISPLAY **UNPREDICTABLE** setting a pending bit to 1 for an exception with priority >= execution priority.

ArchMsg.Warning.warning_unsupported_access_to_memory_mapped_register

TAGS **UNPREDICTABLE** DISPLAY Access to memory mapped register is unsupported.

ArchMsg.Warning.warning_user_jmcr_access

TAGS **UNPREDICTABLE** DISPLAY User %{WRITE:(write|read)} access to JMCR is unpredictable. Fields:

WRITE unsigned int

access is write.

ArchMsg.Warning.warning_wcr_mask_and_bas

TAGS **UNPREDICTABLE** DISPLAY DBGWCR_EL1.MASK is non-zero and BAS is not 0xff.

ArchMsg.Warning.warning_wcr_mask_reserved

TAGS **UNPREDICTABLE** DISPLAY DBGWCR_ELn.MASK is set to a reserved value.

ArchMsg.Warning.warning_wcr_non_configuous_bas

TAGS **UNPREDICTABLE** DISPLAY DBGWCR_EL1.BAS has non-contiguous set of ones.

ArchMsg.Warning.warning_wvr_masked_bit_not_zero

TAGS **UNPREDICTABLE** DISPLAY DBGWVRn_EL1.VA contains a bit masked by DBGWCRn_EL1.MASK that is not zero.

ArchMsg.Why.why_branch_target_exception

A branch target exception has occurred. DISPLAY branch target exception: target pc=%{PC}, target opcode=%{OPCODE}. PRIMARY KEY PC. Fields:

BTYPE unsigned int

pstate.BTYPE at time branch taken.

OPCODE unsigned int

opcode at branch target.

PC unsigned int

Virtual Address of branch target.

ArchMsg.Why.why_illegal_state

An illegal mode change has occurred. DISPLAY illegal mode change: %{MESSAGE}. Fields:

MESSAGE string

The message.

ArchMsg.warning_invalid_interval_counter_reload_value

Invalid interval counter reload value in PMSIRR_EL1.INTERVAL: software should set this to a value greater than or equal to the minimum indicated by PMSIDR_EL1.Interval.

ArchMsg.warning_invalid_spe_buffer_write_pointer

Invalid value programmed in PMBPTR_EL1.PTR or PMBLIMITR_EL1.LIMIT: Current write pointer must be at least one sample record below the write limit pointer and PMBPTR_EL1.PTR[63:56] must be equal to PMBLIMITR_EL1.LIMIT[63:56].

BRANCH_MISPREDICT

Simulating branch mispredict. Fields:

PC unsigned int

Origin address (or 0 if unavailable).

BRA_DIR

Direct branches, to immediate address. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISSET enum

The instruction set of the branch instruction.

IS_COND bool

Indicates if this is a conditional branch.

IS_HINTED bool

Whether the conditional instruction is hinted.

IS_ISB bool

Whether the instruction is ISB.

IS_LINK bool

Indicates if this is a branch with link.

PC unsigned int

The address of the branch instruction.

TARGET_ISET enum

The instruction set after the branch.

TARGET_PC unsigned int

The address the instruction branches to.

BRA_INDIR

Indirect branches, perhaps to a register. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISET enum

The instruction set of the branch instruction.

IS_COND bool

Indicates if this is a conditional branch.

IS_HINTED bool

Whether the conditional instruction is hinted.

IS_ISB bool

Whether the instruction is ISB.

IS_LINK bool

Indicates if this is a branch with link.

PC unsigned int

The address of the branch instruction.

TARGET_ISET enum

The instruction set after the branch.

TARGET_PC unsigned int

The address the instruction branches to.

BRA_INDIR_CCFAIL

Indirect branches fail, perhaps to a register.

BRA_RET_CCFAIL

Branch return fail, perhaps to a register.

CACHE_MAINTENANCE_OP

Cache Maintenance Operation. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

DATA unsigned int

Specified MVA or Set/way.

FUNCTION enum

Clean / Invalidate.

SCOPE enum

Affected region.

SIDE enum

Inst / Data.

CCFAIL

Conditional instruction condition check fail. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CCFAIL_UNC

Unconditional instruction condition check fail. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CCPASS

Conditional instruction condition check pass. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CCPASS_UNC

Unconditional instruction condition check pass. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CODE_CACHE_MAINT

Code cache maintenance. Fields:

END_INCL unsigned int

Inclusive end address for VA or PA requests.

NS bool

Non-secure world for PA.

START unsigned int

Start address for VA or PA requests.

TYPE enum

Request type.

COMPILE_BLOCK_END

Last instruction of basic block translated. Fields:

VADDR unsigned int

Address of next instruction after this basic block.

COMPILE_BLOCK_START

First instruction of basic block translated. Fields:

VADDR unsigned int

Address of first instruction in basic block.

COMPILE_INST

ARM instruction compiled. Fields:

DISASS string

Disassembly of the instruction.

ISSET enum

The instruction set of this instruction.

ITSTATE unsigned int

The ITSTATE current for the instruction.

OPCODE unsigned int

The opcode of the instruction.

PC unsigned int

The address of the instruction.

SIZE unsigned int

The size of the instruction in bytes.

CONTEXTIDR

Context ID Register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

NS bool

Secure or nonsecure banked register is accessed.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The new value written.

CONTEXT_SYNC

Called for every context synchronization event. Fields:

ARCH_CSE bool

Architectural Context Synchronization Event.

PC unsigned int

The address of the synchronization instruction.

SYSREG_SYNC bool

System register synchronization event.

CORE_ENDIAN

Core BE8 Big-Endian state changed. Fields:

BE8 bool

Core BE8 Big-Endian state.

CORE_INFO

Static processor attributes. Only triggered by a call to DumpState(). Fields:

ARCH_PROFILE enum

The architecture profile of the core.

CLUSTER_ID unsigned int

The cluster ID of this processor.

CORE_NUM unsigned int

The number of this core in an MP processor.

DETAILED_CORE_FEATURES enum

Detailed core features as a bitfield. Enum values are bitmasks to extract the info from the field.

FPU_VERSION enum

The VFP version implemented by the core.

MEM_ARCH enum

The memory architecture of the core.

MIDR_EL1 unsigned int

MIDR_EL1 value.

NUM_CORES unsigned int

The number of cores in this MP processor.

QUANTUM_SIZE unsigned int

The default quantum size of the core (in ticks).

SECURITY_FEATURES bool

Does the core have security features?.

CORE_LOADS

Processor load accesses. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ACQREL enum

Is this an acquire/release.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DATA unsigned int

The data read or written.

ELEMENT_SIZE unsigned int

Width of each element.

ENDIANNESS enum

Byte ordering of this access.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

LOCK enum

Normal, exclusive or locked access.

MECID unsigned int

Memory Encryption Context ID, if applicable.

MECID2 unsigned int

Memory Encryption Context ID of the second page, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PAGE_SIZE unsigned int

Page size as $\log_2(\text{size in bytes})$.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK, 2=Exclusive Failed, 3=Unsupported.

SIZE unsigned int

Width of the access in bytes. Only required if DATA is not traced.

TRANS bool

Is this a translated access.

VADDR unsigned int

The virtual address of the access.

CORE_REGS

Changes of the core registers R0 to R14. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number, 0 to 14.

MODE enum

Bank of the register accessed.

OLD_VALUE unsigned int

The old value overwritten.

PHYS_ID enum

The physical register accessed.

VALUE unsigned int

The new value written to the register.

CORE_REGS64

Changes of the core registers X0..X30, SP_ELn. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ID enum

The register number, 0-30 for X0-X30, >=32 for SP_ELn.

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

CORE_REGS64_READ

General purpose AArch64 register has been read X0..X30, SP_ELn. Fields:

ID enum

The register number, 0-30 for X0-X30, >=32 for SP_ELn.

VALUE unsigned int

The value read from the register.

CORE_STALLED_TRANSITION

Detect when a core is prevented from progressing PChannel states. Fields:

CURRENT_STATE enum

Current power mode of core.

REASON string

The reason a transition was stalled, and what must be done to unblock it.

TARGET_STATE enum

Desired power mode of core after transition.

CORE_STORES

Processor store accesses. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ACQREL enum

Is this an acquire/release.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DATA unsigned int

The data read or written.

ELEMENT_SIZE unsigned int

Width of each element.

ENDIANNESS enum

Byte ordering of this access.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

LOCK enum

Normal, exclusive or locked access.

MECID unsigned int

Memory Encryption Context ID, if applicable.

MECID2 unsigned int

Memory Encryption Context ID of the second page, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PAGE_SIZE unsigned int

Page size as $\log_2(\text{size in bytes})$.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK, 2=Exclusive Failed, 3=Unsupported.

SIZE unsigned int

Width of the access in bytes. Only required if DATA is not traced.

TRANS bool

Is this a translated access.

VADDR unsigned int

The virtual address of the access.

CP14_READ

System Coprocessor register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP14 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc1 unsigned int

opcode 1.

opc2 unsigned int
opcode 2.

CP14_WRITE

System Coprocessor register write. Fields:

CORE_NUM unsigned int
Core number in a multi processor.

CRm unsigned int
CRm.

CRn unsigned int
CRn.

NS bool
Secure or nonsecure banked register is accessed.

REG_NAME string
Name of the CP14 register accessed.

SECURITY_STATE enum
The current security state.

UNDEF bool
The register accessed is undefined.

UPDATED_VALUE unsigned int
Updated value of the register now it has been written.

VALUE unsigned int
The new value written.

opc1 unsigned int
opcode 1.

opc2 unsigned int
opcode 2.

CP15_READ

System Coprocessor register read. Fields:

CORE_NUM unsigned int
Core number in a multi processor.

CRm unsigned int
CRm.

CRn unsigned int
CRn.

NS bool
Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP15 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc1 unsigned int

opcode 1.

opc2 unsigned int

opcode 2.

CP15_READ64

System Coprocessor register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP15 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc unsigned int

opcode 1.

CP15_WRITE

System Control Coprocessor register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP15 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc1 unsigned int

opcode 1.

opc2 unsigned int

opcode 2.

CP15_WRITE64

System Coprocessor register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP15 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc unsigned int

opcode 1.

CPSR

CPSR change. (In AArch64 PSTATE represented in SPSR format). Fields:

CORE_NUM unsigned int

Core number in a multi processor.

OLD_VALUE unsigned int

The old CPSR value.

UNKNOWN unsigned int

Bits within the register that have unknown value.

VALUE unsigned int

The new CPSR value.

CRYPTO_SPEC

Every crypto instruction speculatively executed.

CorePowerStateChange

Core power-up state has changed. DISPLAY Core powered %{PU:(up|down)}. Fields:

PU bool

Whether the core powered up.

DATA_CACHE_SET_ALLOC_TAGS

Set allocation tags from DC GVA instruction. Fields:

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

NSDESC unsigned int

The security state of the access.

PADDR unsigned int

The physical address of the access.

PAS enum

The physical address space of the page.

RESPONSE enum

0=Aborted, 1=OK.

SIZE unsigned int

The size of memory in bytes.

TAG unsigned int

The value of the tag being set.

TAGGING_TYPE enum

Type of Memory tagging (Physical or Virtual).

TAG_VADDR unsigned int

The virtual address of the Virtual Tag if Virtual tagging is used; otherwise 0.

VADDR unsigned int

Virtual address used for setting allocation tags.

DATA_CACHE_ZERO

Zero and invalidate cache line from DC ZVA instruction. Fields:

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

NSDESC unsigned int

The security state of the access.

PADDR unsigned int

The physical address of the access.

PADDR_TAG unsigned int

If tag checked, the physical address tag for the access.

PAS enum

The physical address space of the page.

RESPONSE enum

0=Aborted, 1=OK.

SIZE unsigned int

The size of zero'd memory in bytes.

TAG_CHECKED bool

Is this access tag checked.

VADDR unsigned int

Virtual address of the zero'd cache line.

DEBUG_EVENT

Hardware debug support event. Fields:

EVENT enum

Description of event.

VALUE unsigned int

data value.

DEBUG_TARGET_ISET

Hardware debug target ISA. Fields:

TARGET_ISET enum

The target instruction set of the debug state.

DMI_ALLOCATE

DMI allocated. Fields:

ACCESS_DIRECTION enum

READ / WRITE.

PADDR unsigned int

The physical address of the DMI.

PTR unsigned int

DMI range pointer.

SIDE enum

DSIDE / ISIDE.

SIZE unsigned int

The size of DMI.

VADDR unsigned int

The virtual address of the DMI..

DMI_HIT

DMI hit, access used DMI (only fire for small DMI regions i.e. less than a full page). Fields:

ACCESS_DIRECTION enum

READ / WRITE.

PADDR unsigned int

The physical address of the DMI.

SIDE enum

DSIDE / ISIDE.

SIZE unsigned int

The size of the DMI.

VADDR unsigned int

The virtual address of the DMI..

DMI_REVOKE

DMI revoked. Fields:

ACCESS_DIRECTION enum

READ / WRITE.

PADDR unsigned int

The physical address of the DMI.

SIDE enum

DSIDE / ISIDE.

SIZE unsigned int

The size of DMI.

VADDR unsigned int

The virtual address of the DMI..

END_COMPILE

Compilation end. Fields:

END_OF_PAGE_COUNT unsigned int

Number of basic blocks exited due to page boundary since START_COMPILE.

FETCHFAIL_COUNT unsigned int

Number of basic blocks exited due to ifetch failure START_COMPILE.

INST_COUNT unsigned int

Number of instructions compiled since START_COMPILE.

NONSEQ_COUNT unsigned int

Number of basic blocks exited due to non-sequential instructions since START_COMPILE.

PAGE_STRADDLE_COUNT unsigned int

Number of basic blocks exited due to unaligned instructions crossing page since START_COMPILE.

ESB

Error Synchronization Barrier hit. Fields:

ASYNC bool

ESB is async.

IMPLICIT bool

ESB is implicit.

EXCEPTION

Exception taken. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ESR unsigned int

The value assigned to ESR when exception is taken.

ESR_EC unsigned int

The Value assigned to ESR.EC when exception is taken.

IS_PHYSICAL bool

Physical or Virtual exception.

IS_TRAP bool

Exception coming from a trap.

LR unsigned int

The value assigned to the link register.

PC unsigned int

The location where the exception occurred.

PREFERRED_RETURN unsigned int

The preferred return address for the exception.

TAKEN_LOCALLY bool

Whether exception is taken locally(same exception level).

TARGET_ISET enum

The instruction set of the exception handler code.

TARGET_PC unsigned int

The address the exception branches to.

VECTOR enum

The exception vector.

VECTOR_OFFSET unsigned int

The offset of an exception vector from the base address.

EXCEPTION_END

Every exception completed.

EXCEPTION_INFO

Trace human-readable reason of the current exception for data/instruction aborts, floating point exceptions and unknown exceptions. Fields:

DESCRIPTION string

Reason for current exception.

ESR_EC unsigned int

The value of ESR.EC when exception is taken.

EXCEPTION_RAISE

Every exception raised.

EXCEPTION_RETURN

Branches on leaving exception. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISSET enum

The instruction set of the branch instruction.

IS_COND bool

Indicates if this is a conditional branch.

IS_HINTED bool

Whether the conditional instruction is hinted.

IS_ISB bool

Whether the instruction is ISB.

IS_LINK bool

Indicates if this is a branch with link.

PC unsigned int

The address of the branch instruction.

TARGET_ISET enum

The instruction set after the branch.

TARGET_PC unsigned int

The address the instruction branches to.

EXCEPTION_RETURN_PREBRANCH

Exception return event that must occur before PC and processor state have updated.

EXCEPTION_START

Every exception started.

FIQ_TAKEN

FIQ taken exception.

FP_EXCEPTIONS_SIGNALLED

Trace which FP exceptions were signalled as a part of the FP operations performed by an instruction. Fields:

FP_EXCEPTIONS_SIGNALLED_MASK unsigned int

Indicates which FP exceptions were signalled by an instruction. This uses the same encoding as the cumulative floating-point exception bitfields in FPSR.

FP_STATE

Floating point state. Fields:

FPSR unsigned int

FPSR register value.

MASK unsigned int

The mask indicating updated cumulative bits.

FREQ_CHANGED

Core frequency changed. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The instruction count of this core.

NEW_FREQ signed int

The new frequency of this core (expressed in Hz).

OLD_FREQ signed int

The old frequency of this core (expressed in Hz).

HLT

HLT instruction occurred. Fields:

IMM16 unsigned int

The 16-bit immediate value encoded in HLT instruction.

INST

Every instruction executed. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DEBUG_STATE bool

The instruction is executed in debug state.

DISASS string

Disassembly of instruction.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISSET enum

The current instruction set.

ITSTATE unsigned int

The current ITSTATE.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

MODE enum

The mode the core is in.

NS unsigned int

The core's non-secure bit.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

OPCODE unsigned int

The opcode of the instruction.

PADDR unsigned int

The physical address of the instruction.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the instruction.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the page.

PC unsigned int

The address of the instruction.

SECURITY_STATE enum

The core's security state.

SIZE unsigned int

The size of the instruction in bytes.

INST_END

Every instruction completed.

INST_START

Every instruction started. Fields:

DEBUG_STATE bool

The instruction is executed in debug state.

ISSET enum

The current instruction set.

MODE enum

The mode the core is in.

NS enum

The current Secure State.

PC unsigned int

The address of the conditional instruction.

SECURITY_STATE enum

The current Security State.

INST_STRADDLE

Instruction straddles boundary. Fields:

DEBUG_STATE bool

The instruction is executed in debug state.

ISSET enum

The current instruction set.

MODE enum

The mode the core is in.

NS enum

The current Secure State.

PC unsigned int

The address of the conditional instruction.

SECURITY_STATE enum

The current Security State.

IRQ_TAKEN

IRQ taken exception.

ISB

ISB instruction occurred. Fields:

PC unsigned int

The address of the ISB instruction.

LOAD_MULTIPLE_REGS_LIST

List of destination registers (not source) of a multiple registers load instruction. Fields:

REG string

Register to add into the list.

LOCAL_MONITOR

Local monitor activity. Fields:

PADDR unsigned int

Local Monitor Address.

State enum

State of the monitor (Open/Exclusive).

MAPPING_CREATE

A memory mapping has been created. This is an implementation detail of the model and so may change without notice. Fields:

ASID unsigned int

The ASID used during translation, or 0 if no ASID was used.

IS_VALID bool

Whether the mapping is valid or not.

PADDR unsigned int

Base physical address of region, or 0 if not known.

SIDE enum

I-side or D-side.

SIZE unsigned int

Size of the page, or 0 if not known.

VADDR unsigned int

Base virtual address of region.

VMID unsigned int

The VMID used during translation, or 0 if no VMID was used.

MAPPING_DESTROY

A memory mapping has been destroyed. This is an implementation detail of the model and so may change without notice. Fields:

ASID unsigned int

The ASID used during translation.

CAUSE enum

The cause of invalidation.

PADDR unsigned int

Base physical address of page.

SIDE enum

I-side or D-side.

SIZE unsigned int

Size of the page.

VADDR unsigned int

Base virtual address of page.

VMID unsigned int

The VMID used during translation.

MEMMAP_DEBUG_READ

Memory mapped reads to the debug registers. Fields:

ADDR unsigned int

address.

CORE_NUM unsigned int

Core number in a multi processor.

EXT bool

Whether access is from an external device (such as the DAP).

NS enum

Secure state of the access.

REG_NAME string

Name of the debug register accessed.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

MEMMAP_DEBUG_WRITE

Memory mapped writes to the debug registers. Fields:

ADDR unsigned int

address.

CORE_NUM unsigned int

Core number in a multi processor.

EXT bool

Whether access is from an external device (such as the DAP).

NS enum

Secure state of the access.

REG_NAME string

Name of the debug register accessed.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The new value written.

MMU_D128_TTB_READ

This event is triggered by reads caused by a D128 translation table walk. Fields:

ABORTED bool

Was the walk successful.

DATA_HIGH unsigned int

The data upper bits read.

DATA_LOW unsigned int

The data lower bits read.

IPA unsigned int

For stage 1, the IPA of the read.

LEVEL unsigned int

Translation table level.

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

OUTADDR unsigned int

The output physical address.

PADDR unsigned int

The physical address of the read.

PASmem enum

Physical Address Space of the accessed memory.

PASreq enum

Physical Address Space of the requested lookup.

REGIME_EL enum

Entry matches in this translation regime.

REQUESTER enum

The requester of this table walk.

SIDE enum

Inst / Data.

STAGE unsigned int

Translation stage.

MMU_D128_TTB_WRITE

This event is triggered by writes caused by a D128 translation table walk. Fields:

ABORTED bool

Was the walk successful.

DATA_HIGH unsigned int

The data upper bits written.

DATA_LOW unsigned int

The data lower bits written.

IPA unsigned int

For stage 1, the IPA of the write.

IS_HACDBS_ACCESS bool

If the access was due to HACDBS_CLEAN.

LEVEL unsigned int

Translation table level.

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

OUTADDR unsigned int

The output physical address.

PADDR unsigned int

The physical address of the write.

PASmem enum

Physical Address Space of the accessed memory.

PASreq enum

Physical Address Space of the requested lookup.

REGIME_EL enum

Entry matches in this translation regime.

SIDE enum

Inst / Data.

STAGE unsigned int

Translation stage.

MMU_TRANS

Address translation information. Fields:

ASID unsigned int

Address space identifier.

CORE_NUM unsigned int

Core number in a multi processor.

Hyp bool

Entry matches in Hyp state only.

INNERCACHE_RA bool

Is the inner cache allocate on read.

INNERCACHE_TRANSIENT bool

Is the inner write-through transient.

INNERCACHE_TYPE enum

Inner Caching scheme (NC/MB/WA).

INNERCACHE_WA bool

Is the inner cache allocate on write.

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MEMTYPE enum

Memory type.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC enum

Is secure side supposed to access secure or nonsecure memory.

OUTERCACHE_RA bool

Is the outer cache allocate on read.

OUTERCACHE_TRANSIENT bool

Is the outer write-through transient.

OUTERCACHE_TYPE enum

Outer Caching scheme (NC/MB/WA).

OUTERCACHE_WA bool

Is the outer cache allocate on write.

PADDR unsigned int

Physical address of the access.

PAGESIZE unsigned int

Page size as log2(size).

PAS enum

PAS supposed to be accessed is secure or nonsecure memory or Root/Realm if RME is implemented.

SH enum

Shareability.

SIDE enum

Inst / Data.

STAGE1_PERM unsigned int

Stage 1 permissions, [5:3] Privileged access (XWR) [2:0] User access (XWR).

STAGE2_PERM unsigned int

Stage 2 permission mask [5:3] Privileged access (XWR) [2:0] User access (XWR).

VADDR unsigned int

Virtual address of the access.

VMID unsigned int

Virtual machine identifier.

XS bool

XS Attribute.

nG enum

Flag indicating whether ASID will be matched.

MMU_TTB_READ

This event is triggered by reads caused by a translation table walk. Fields:

ABORTED bool

Was the walk successful.

DATA unsigned int

The data read.

DESC_KIND enum

The encoding scheme of the descriptors used for this walk.

IPA unsigned int

For stage 1, the IPA of the read.

LEVEL unsigned int

Translation table level.

LPAE bool

Is this for an LPAE translation (*DEPRECATED*, use DESC_KIND).

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSmem bool

Non secure memory access.

NSreq bool

Non secure request.

OUTADDR unsigned int

The output physical address.

PADDR unsigned int

The physical address of the read.

PASmem enum

Physical Address Space of the accessed memory.

PASreq enum

Physical Address Space of the requested lookup.

REGIME_EL enum

Entry matches in this translation regime.

REQUESTER enum

The requester of this table walk.

SIDE enum

Inst / Data.

STAGE unsigned int

Translation stage.

MMU_TTB_WRITE

This event is triggered by writes caused by a translation table walk. Fields:

ABORTED bool

Was the walk successful.

DATA unsigned int

The data written.

DESC_KIND enum

The encoding scheme of the descriptors used for this walk.

IPA unsigned int

For stage 1, the IPA of the write.

IS_HACDBS_ACCESS bool

If the access was due to HACDBS_CLEAN.

LEVEL unsigned int

Translation table level.

LPAE bool

Is this for an LPAE translation (*DEPRECATED*, use DESC_KIND).

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSmem bool

Non secure memory access.

NSreq bool

Non secure request.

OUTADDR unsigned int

The output physical address.

PADDR unsigned int

The physical address of the write.

PASmem enum

Physical Address Space of the accessed memory.

PASreq enum

Physical Address Space of the requested lookup.

REGIME_EL enum

Entry matches in this translation regime.

SIDE enum

Inst / Data.

STAGE unsigned int

Translation stage.

MODE_CHANGE

Mode change. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

MODE enum

The new mode.

NON_SECURE enum

The core's new non-secure bit.

OLD_MODE enum

The old mode.

SECURITY_STATE enum

The new security state.

NOT_TAG_CHECKED_INST

MTE Tag not checked for current instruction.

PERIODIC

Called for every quantum. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The instruction count of this CPU.

PC unsigned int

The address of the next instruction to be executed on this CPU.

PMU_COUNTER_OVERFLOW

PMU counter overflow. Fields:

COUNTER_GROUP enum

Counter group.

COUNTER_RANGE enum

Counter range.

EVENT_ID unsigned int

Event Identifier, field is valid only for Event counter group.

INDEX unsigned int

Counter index (as selected by PMSELR), field is valid only for Event/Cycle counter groups.

INTERRUPT bool

Is interrupt enabled on overflow for this counter.

POINTER_AUTH

Called when execution hits a pointer authentication check. Fields:

AUTH_STATE bool

Authentication result.

KEY_USED enum

Pointer authentication key.

POTINDEX_UPDATE

Value of POT index for current pc changed. Fields:

NEW_POTINDEX unsigned int

New POTIndex.

OLD_POTINDEX unsigned int

Old POTIndex.

PREFETCH_MEMORY64

Prefetch from PRFM or PRFUM instructions. Fields:

PRFOP enum

Prefetch hint.

VADDR unsigned int

Virtual address of the location that should be prefetched.

PRELOAD64_REQUEST

Preload requests from PRFM or PRFUM instructions. Fields:

PRFOP enum

Prefetch hint.

VADDR unsigned int

Virtual address of the location that should be prefetched.

PRELOAD_DATA

Data preload from PLD instruction. Fields:

VADDR unsigned int

Virtual address of the data that should be preloaded.

PRELOAD_INST

Instruction preload from PLI instruction. Fields:

VADDR unsigned int

Virtual address of the instruction that should be preloaded.

PRE_CORE_LOAD

Trace just before a core load. Fields:

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

PRE_CORE_STORE

Trace just before a core store. Fields:

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

PRE_TTB_READ

This event is triggered before reads caused by a translation table walk. Fields:

ADDR unsigned int

The physical address of the read.

PSTATE

PSTATE fields update. Fields:

ALLINT unsigned int

All interrupt mask bit.

BTYP unsigned int

Branch target identification bit.

CORE_NUM unsigned int

Core number in a multi processor.

DAIF unsigned int

Exception mask bits.

DIT unsigned int

Data Independent Timing bit.

EL unsigned int

Current Exception level field.

EXLOCK unsigned int

Exception return state lock bit.

IL unsigned int

Illegal Execution state bit.

NZCV unsigned int

Condition flags.

PACM unsigned int

PACM bit.

PAN unsigned int

Privileged Access Never state bit.

PM unsigned int

PMU exception mask bit.

PPEND unsigned int

PMU exception pending bit.

SM unsigned int

StreamingMode status bit.

SP unsigned int

Stack pointer register selection bit.

SS unsigned int

Software Step bit.

SSBS unsigned int

Speculative Store Bypass Safe bit.

TCO unsigned int

Tag Check Override bit.

TOUCHED unsigned int

Mask of the fields that have been touched since this trace has last been emitted. Note that touched does not necessarily mean the value changed. The index of the field is the bit index in the mask plus the index of this field plus one.

UAO unsigned int

User Access Override bit.

UINJ unsigned int

Undefined Instruction exceptions bit.

ZA unsigned int

ZA storage and ZT0 register enable bit.

nRW unsigned int

Current Execution state bit.

PSTATE_PACTIVE_OPMODE_TRANSITION

Trace pactive and Opmode mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Operating mode after the transition.

OLD_STATE enum

Operating mode before the transition.

PSTATE_PACTIVE_OPMODE_TRANSITION_DENIED

Trace the denied pactive and Opmode mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Operating mode after the transition.

OLD_STATE enum

Operating mode before the transition.

REASON string

Reason for the transition being denied.

PSTATE_PACTIVE_POWER_TRANSITION

Trace pactive and power mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Power mode after the transition.

OLD_STATE enum

Power mode before the transition.

PSTATE_PACTIVE_POWER_TRANSITION_DENIED

Trace the denied pactive and power mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Power mode after the transition.

OLD_STATE enum

Power mode before the transition.

REASON string

Reason for the transition being denied.

PSTATE_PACTIVE_THREAD_OPMODE_TRANSITION

Trace pactive and Opmode mode transitions for threads on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Operating mode after the transition.

OLD_STATE enum

Operating mode before the transition.

QUANTUM_START

Called at the start of every quantum.

RANGE_PRELOAD_REQUEST

Preload requests from RPRFM. Fields:

COUNT unsigned int

Number of blocks of data to access.

LENGTH unsigned int

Number of contiguous bytes to be accessed in each block.

OP unsigned int

Range prefetch operation .

REUSE unsigned int

Maximum number of bytes to be accessed before next RPRFM .

STRIDE signed int

Stride to advance after accessing LENGTH bytes.

VADDR unsigned int

Base virtual address of range prefetch request.

RUN_STATE

Run state transition. Fields:

INST_COUNT unsigned int

Ticks count at point of transition.

NEW enum

New run state.

OLD enum

Old run state.

SBP_SAMPLE_COLLISION

DISPLAY Statistical profiling samples collided .

SBP_SAMPLE_DISCARDED

DISPLAY Operation discarded by statistical profiling unit when configured in discard mode.

Fields:

BUFFER_POINTER unsigned int

Value of buffer pointer.

DATA unsigned int

Data discarded.

SIZE unsigned int

Size of sample, in bytes.

SBP_SAMPLE_END

DISPLAY Statistical profiling of an operation ended.

SBP_SAMPLE_POP

DISPLAY Operation which might be sampled by statistical profiling.

SBP_SAMPLE_RECORDED

DISPLAY Operation sampled by statistical profiling has been written to the buffer at %{BUFFER_ADDRESS} with size %{SIZE} and data %{DATA}. Fields:

BUFFER_ADDRESS unsigned int

Virtual Address of start of buffer write.

DATA unsigned int

Data written to the buffer.

SIZE unsigned int

Size of sample written to buffer, in bytes.

SBP_SAMPLE_START

DISPLAY Statistical profiling of an operation started.

SBP_SAMPLE_TAKEN

DISPLAY Operation sampled by statistical profiling after filtering.

SEMIHOSTING_CALL

Call of a semihost function occurred. Fields:

PC unsigned int

The program counter after the semihosting call.

SEMIHOSTING_PRECALL

About to call semihost. Fields:

EL2 **bool**

Use EL2 translation regime.

NS **bool**

Is Non-Secure.

REG_WIDTH **unsigned int**

The current register width in bytes.

SIGNAL

External signal state change. Fields:

SIGNAL **enum**

Signal that changed.

STATE **bool**

Signal asserted state.

SOFTWARE_STEP

Return the Debug Software Step state. Fields:

STATE **enum**

Software Step state.

SPSR

SPSR change. Fields:

CORE_NUM **unsigned int**

Core number in a multi processor.

MODE **enum**

Which of the banked SPSR registers is written.

OLD_VALUE **unsigned int**

The old SPSR value.

VALUE **unsigned int**

The new SPSR value.

START_COMPILE

Compilation started. Fields:

VADDR **unsigned int**

Instruction where compilation begins.

STG_MULTIPLE_MICRO_OPS_ADDRESS_RANGE

Address range for MTE stores with multiple uops. Fields:

ADDR_END **unsigned int**

Final address.

ADDR_START unsigned int

Starting address.

STORE_MULTIPLE_REGS_ADDRESS_RANGE

Address range for stores with multiple registers. Fields:

ADDR_END unsigned int

Final address.

ADDR_START unsigned int

Starting address.

SUBOPTIMAL_LDST_RETIRED

suboptimal load/store. Fields:

ACCESS_SIZE unsigned int

Log2 of the access width in bytes used for alignment checking (1: half-word, 2: word ...).

OPTIMAL_ALIGN_SELECTOR unsigned int

Selector to decide the optimal alignment size (1: access width, 2: 4-bytes, 3: 8-bytes, ..., 12: 4-kbytes).

VADDR unsigned int

The virtual address of the access.

SYNC

Called for every synchronization, that is every quantum break. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The instruction count of this CPU.

LOCAL_QUANTUM unsigned int

The local quantum of this CPU.

LOCAL_TIME unsigned int

The local time of this CPU.

SYNC_REASON enum

Reason for sync.

SYSCALL

System call instruction executed. Fields:

IMM unsigned int

Immediate value of the system call instruction.

TYPE enum

System call type.

VADDR unsigned int

Instruction that caused the system call.

SYSREG_READ128

System Coprocessor 128 bit register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

REGNUM enum

Internal register number.

REGNUM_PHYS enum

Internal register number physically accessed (e.g. in presense of VHE redirection).

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_READ64

System Coprocessor register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REGNUM enum

Internal register number.

REGNUM_OPERAND enum

Register number corresponding to the instruction operand.

REGNUM_PHYS enum

Internal register number physically accessed (e.g. in presense of VHE redirection).

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_UPDATE128

Triggers when the system updates a register. Fields:

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG enum

Register number.

UNKNOWN unsigned int

Bits of the register which became unknown.

VALUE unsigned int

Value written to the register.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_UPDATE32

Triggers when the system updates a register. Fields:

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG enum

Register number.

UNKNOWN unsigned int

Bits of the register which became unknown.

VALUE unsigned int

Value written to the register.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_UPDATE64

Triggers when the system updates a register. Fields:

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG enum

Register number.

UNKNOWN unsigned int

Bits of the register which became unknown.

VALUE unsigned int

Value written to the register.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int
opcode 2.

SYSREG_WRITE128

System Coprocessor 128 bit register write. Fields:

CORE_NUM unsigned int
Core number in a multi processor.

CRm unsigned int
CRm.

CRn unsigned int
CRn.

REGNUM enum
Internal register number.

REGNUM_PHYS enum
Internal register number physically accessed (e.g. in presense of VHE redirection).

UNDEF bool
The register accessed is undefined.

UPDATED_VALUE unsigned int
Updated value of the register now it has been written.

VALUE unsigned int
The new value written.

opc unsigned int
opcode 1.

opc0 unsigned int
opcode 0.

opc2 unsigned int
opcode 2.

SYSREG_WRITE64

System Coprocessor register write. Fields:

CORE_NUM unsigned int
Core number in a multi processor.

CRm unsigned int
CRm.

CRn unsigned int
CRn.

NS bool
Secure or nonsecure banked register is accessed.

REGNUM enum

Internal register number.

REGNUM_OPERAND enum

Register number corresponding to the instruction operand.

REGNUM_PHYS enum

Internal register number physically accessed (e.g. in presense of VHE redirection).

UNDEF bool

The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

UNALIGNED_LDST_RETIRED

Processor unaligned load/store. Fields:

ACCESS_SIZE unsigned int

Log2 of the access width in bytes used for alignment checking (1=>half-word, 2->word ...).

NUMBER_OF_BEATS unsigned int

Number of accesses (beats) in this burst (Total size in bytes is ACCESS_SIZE * NUMBER_OF_BEATS).

VADDR unsigned int

The virtual address of the access.

VFP_D_REGS

VFP/NEON D 64 bit register write. Fields:

ALIASING enum

Type of register aliasing used.

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a write to the corresponding bit in the register. For example, 0x1 would indicate a write to VALUE[0].

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

VFP_D_REGS_READ

VFP/NEON D 64 bit register read. Fields:

ID unsigned int

The register number.

VALUE unsigned int

The current value of the register.

VFP_Q_REGS

VFP/NEON Q 128 bit register write. Fields:

ALIASING enum

Type of register aliasing used.

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a write to the corresponding byte in the register. For example, 0x1 would indicate a write to VALUE[7:0].

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

VFP_Q_REGS_READ

VFP/NEON Q 128 bit register read. Fields:

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a read to the corresponding byte in the register. For example, 0x1 would indicate a read to VALUE[7:0].

VALUE unsigned int

The current value of the register.

VFP_SYS_REGS

Writes to the VFP/NEON units system registers. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ID enum

Which VFP system register is written.

MASK unsigned int

The mask indicating updated cumulative bits.

OLD_VALUE unsigned int

The register's old value overwritten.

VALUE unsigned int

The new value written to the VFP system register.

VFP_S_REGS

VFP/NEON S 32 bit register write. Fields:

ALIASING enum

Type of register aliasing used.

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a write to the corresponding bit in the register. For example, 0x1 would indicate a write to VALUE[0].

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

VFP_S_REGS_READ

VFP/NEON S 32 bit register read. Fields:

ID unsigned int

The register number.

VALUE unsigned int

The current value of the register.

VIRTUAL_TAG_ALLOCATED

Virtual tag information when allocated. Fields:

TAG_VADDR unsigned int

Virtual tag virtual address.

VADDR unsigned int

Data Virtual address.

VIRTUAL_TAG unsigned int

Virtual tag value.

WAYPOINT

Signals end of basic block execution. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ISET enum

Origin instruction set.

IS_COND bool

Indicates if this is a conditional waypoint.

PC unsigned int

Origin address (or 0 if unavailable).

TAKEN bool

Indicates if this waypoint was taken.

TARGET unsigned int

Destination address.

TARGET_ISET enum

Destination instruction set.

WFE_END

WFE ended. Fields:

INST_COUNT unsigned int

Ticks count when leaving WFE.

WFE_EVENT_REGISTER

WFE event register status: set/clear, reason. Fields:

INST_COUNT unsigned int

Ticks count.

REASON enum

Reason for set/clear. Only REASON==0 (Cleared by WFE) clears the bit, all other reasons set it.

WFE_IGNORED

WFE ignored. Fields:

EVENT bool

This WFE was ignored because the event register was set.

INST_COUNT unsigned int

Ticks count when ignoring WFE.

TRAPPED bool

This WFE was trapped.

WFE_START

WFE entered. Fields:

INST_COUNT unsigned int

Ticks count when entering WFE.

WFI_END

WFI ended. Fields:

INST_COUNT unsigned int

Ticks count when leaving WFI.

WFI_IGNORED

WFI ignored. Fields:

DISABLED bool

This WFI was ignored because WFI is disabled.

INST_COUNT unsigned int

Ticks count when ignoring WFI.

REASON enum

specifies reason why WFI trace was ignored.

TRAPPED bool

This WFI was trapped.

WFI_START

WFI entered. Fields:

INST_COUNT unsigned int

Ticks count when entering WFI.

WFI_WAKEUP

WFI wakeup. Fields:

INST_COUNT unsigned int

Ticks count when WFI wakeup occurred.

REASON enum

Reason for wakeup.

2.58 ARM_Cortex-X2

This section describes the trace sources.

AA64_ASE_SVE_REGS

Modification of AArch64 Advanced SIMD or SVE Vector Registers (via SIMD or Scalar operations), SVE Predicate Registers and ZT0. Fields:

ID *enum*

The register identifier.

MASK *unsigned int*

Bitmask of the register to signal the modified bits in the VALUE field.

SM *bool*

Whether the PE is in Streaming Mode.

VALUE *unsigned int*

Value of AArch64 vector register.

ASYNC_MEMORY_FAULT

Context ID Register write. Fields:

FAULT *unsigned int*

Fault status present in ESR.ISS encoding for Serror in ARCH64 and present in DFSR encoding for external abort format in ARCH32.

PADDR *unsigned int*

Physical Address (or 0 if unavailable).

VADDR *signed int*

Virtual Address (or 0 if unavailable).

ATOMIC_END_ACCESS

Bus trace access for atomics. Fields:

ACCESS_FAIL *bool*

Memory access failed.

ADDR *unsigned int*

The virtual address of the access.

ATTR *unsigned int*

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

COMPARE_VALUE *unsigned int*

Compare value for CAS.

LOAD_VALUE unsigned int

Loaded values.

NSDESC unsigned int

The security state of the access.

OPERAND_VALUE unsigned int

Operation's operand.

OPERATION enum

Operation type.

PADDR unsigned int

The physical address of the access.

PRIV bool

Is this a privileged access?.

ATOMIC_START_ACCESS

Bus trace access for atomics. Fields:

ADDR unsigned int

The virtual address of the access.

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

COMPARE_VALUE unsigned int

Compare value for CAS.

NSDESC unsigned int

The security state of the access.

OPERAND_VALUE unsigned int

Operation's operand.

OPERATION enum

Operation type.

PADDR unsigned int

The physical address of the access.

PRIV bool

Is this a privileged access?.

ArchMsg.Error.exit_code

DISPLAY exit code: %{EXITCODE} component kind: %{KIND} reason: %{REASON}. Fields:

EXITCODE signed int

Exit code to be returned.

KIND string

Component kind that invoked the exit code trace.

REASON string

Reason why the exit code trace was invoked (optional).

ArchMsg.Warning.IgnoredWarmResetRequest

Event fires when software attempts to request warm reset through write to deprecated DBGPRCR.CWRR.

ArchMsg.Warning.RAS_ErrselrUnpredictableState

DISPLAY ERRSELR is in unpredictable state due to value greater than number_of_error_records written to it.

ArchMsg.Warning.RAS_UnpredictableStateWrite

DISPLAY %{REG_NAME} write ignored due to unpredictable ERRSELR state. Fields:

REG_NAME string

Name of register being written.

ArchMsg.Warning.ReportedRevisionRegisterRead

DISPLAY Reported revision and patch do not match that of the cpu. Fields:

ORIGINAL_VALUE unsigned int

The value the register would have based on the modelled revision.

REGISTER string

The name of the register that has ben read.

REPORTED_VALUE unsigned int

The value the register has been overridden to.

ArchMsg.Warning.Unpredictably Indexed PM Event Register

TAGS UNPREDICTABLE DISPLAY Event %{IsDirect:(Selector value|Register)} %{SEL} is >= %{N}. Fields:

IsDirect unsigned int

Direct Access rather than indirect.

N unsigned int

Number of Accessible Registers.

SEL unsigned int

Selector.

ArchMsg.Warning.cache_contents_unknown

A cache has been activated without being invalidated and may contain **UNKNOWN** content. DISPLAY %{SIDE:(D|I)}-Cache was enabled in %{REGIME:(SECURE|NON_SECURE|HYP)} regime but was not invalidated since power-on: cache lines could contain **UNKNOWN** content %{SIDE:(Note: D-side caches must invalidate SECURE lines even if the cache is only used by NON_SECURE code in order to ensure that unexpected dirty lines tagged SECURE are not naturally evicted)}. Fields:

REGIME unsigned int

Current translation regime.

SIDE unsigned int

0 d-side cache or unified cache, 1 i-side.

ArchMsg.Warning.dcimvac_matches_watchpoint

TAGS **UNPREDICTABLE** DISPLAY Watchpoints matching an AArch32 DCIMVA are implementation defined.

ArchMsg.Warning.decode_fieldmismatch

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with different fields %{FIELD1} and %{FIELD2}. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_fracbitsnegative

TAGS **UNPREDICTABLE** DISPLAY VCVT Instruction is unpredictable with a negative number of fraction bits.

ArchMsg.Warning.decode_initblock

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable in an IT block.

ArchMsg.Warning.decode_initblocknotlast

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable in an IT block when not the last.

ArchMsg.Warning.decode_invalid_condition

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable when the condition is not AL.

ArchMsg.Warning.decode_invalidfieldcombination

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with the values in %{FIELD1} and %{FIELD2}. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_invalidvalue

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with the value in %{FIELD}. Fields:

FIELD enum

Field name.

ArchMsg.Warning.decode_mem_hint_unallocated

TAGS **UNPREDICTABLE** DISPLAY There is no memory hint allocated to this bit pattern.

ArchMsg.Warning.decode_registermatch

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with %{FIELD1} and %{FIELD2} as the same register. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_registermismatch

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with %{FIELD1} and %{FIELD2} as different registers. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_registeroutofrange

TAGS **UNPREDICTABLE** DISPLAY Floating-point instruction is unpredictable indexing beyond the end of the register bank.

ArchMsg.Warning.decode_sbzsbo

TAGS **UNPREDICTABLE** DISPLAY Reserved bits in the instruction are not canonical.

ArchMsg.Warning.decode_transactiontoobig

TAGS **UNPREDICTABLE** DISPLAY Attempt to perform a memory transaction of more than 128 bytes.

ArchMsg.Warning.decode_unpred_other

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable.

ArchMsg.Warning.decode_unpreduseofpc

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable when %{FIELD} is PC. Fields:

FIELD enum

Field name.

ArchMsg.Warning.decode_unpreduseofr13

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable when %{FIELD} is R13. Fields:

FIELD enum

Field name.

ArchMsg.Warning.decode_unpreduseofr13inreglist

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with R13 in register-list.

ArchMsg.Warning.decode_useofd16orhigher

TAGS **UNPREDICTABLE** DISPLAY Attempt to use register D16 or higher where it is not allowed.

ArchMsg.Warning.decode_writebackandbaseinlist

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with write back when the base register is in the transfer list.

ArchMsg.Warning.decode_zeroregistersinlist

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable when no registers are to be transferred.

ArchMsg.Warning.misaligned_ptrs

DISPLAY TRBPTR_EL1(%{TRBPTR_EL1}) or TRBTRG_EL1(%{TRBTRG_EL1}) are not aligned as per TRBIDR_EL1.Align(%{TRBIDR_EL1}). Fields:

TRBIDR_EL1 unsigned int

TRBIDR_EL1.Align value.

TRBPTR_EL1 unsigned int

Current write pointer.

TRBTRG_EL1 unsigned int

Trigger pointer.

ArchMsg.Warning.out_of_range_ptrs

DISPLAY TRBPTR_EL1(%{TRBPTR_EL1}) is out of range as per TRBBASER_EL1(%{TRBBASER_EL1}) or TRBLIMITR_EL1.LIMIT(%{TRBLIMITR_EL1}). Fields:

TRBBASER_EL1 unsigned int

Base pointer.

TRBLIMITR_EL1 unsigned int

Limit pointer.

TRBPTR_EL1 unsigned int

Current write pointer.

ArchMsg.Warning.recursive_branch

An instruction is performing a branch that targets the same instruction. If this was intended then a WFI might be more effective DISPLAY recursive branch detected at PC=%{PC}.

PRIMARY KEY PC. Fields:

PC unsigned int

Address of instruction causing the branch.

ArchMsg.Warning.recursive_exception

An instruction has generated an exception that targets the same instruction. DISPLAY recursive %{TYPE} exception detected at PC=%{PC} (CPSR=%{CPSR}). PRIMARY KEY PC. Fields:

CPSR unsigned int

Processor state.

PC unsigned int

Address of instruction causing the exception.

TYPE enum

Type of exception.

ArchMsg.Warning.reentrant_vector_catch

TAGS **UNPREDICTABLE** DISPLAY Vector catch debug event on Prefetch or Data abort vector %{ADDR}. Fields:

ADDR unsigned int

vector address.

ArchMsg.Warning.reserved_it_state

TAGS **UNPREDICTABLE** DISPLAY Execution with reserved IT state %{ITSTATE}. Fields:

ITSTATE unsigned int

reserved IT state.

ArchMsg.Warning.tlb_contents_unknown

Translation has been enabled without invalidating the TLB. DISPLAY Translation was enabled in the %{PAS:(Secure|Non-secure|Root|Realm)} EL%{EL:d} translation regime (vmid=%{VMID:x}) without invalidating %{INVALIDITY:(|d-side |i-side |)}entries since power-on: %{INVALIDITY:(|D|I|)}TLB could contain **UNKNOWN** content. Fields:

EL unsigned int

Exception level of translation regime.

INVALIDITY unsigned int

Which TLB may be invalid (if TLBs are separate).

NON_SECURE bool

Whether the translation regime is non-secure.

PAS unsigned int

Physical address space of translation regime.

VMID unsigned int

Current VMID, if applicable.

ArchMsg.Warning.unknown_DLR_DSPSR

Exiting debug state with unknown DLR or DSPSR.

ArchMsg.Warning.unknown_ELR_SPSR

Returning from debug exception with unknown ELR or SPSR.

ArchMsg.Warning.unpred_FAR_on_LRCPC3_non_atomic_mem_access

TAGS **UNPREDICTABLE** Unpredictable FAR/FSTATUS on LRCPC3 pair instructions resulting in two separate memory accessesDISPLAY Unpredictable FAR on LRCPC3 non atomic memory access at %{ADDR1} with FSTATUS %{FSTATUS1} or %{ADDR2} with FSTATUS %{FSTATUS2}. Fields:

ADDR1 unsigned int

Possible FAR address 1.

ADDR2 unsigned int

Possible FAR address 2.

FSTATUS1 unsigned int

Possible fault status value 1.

FSTATUS2 unsigned int

Possible fault status value 2.

ArchMsg.Warning.unpred_async_tag_check_fail_on_sve_speculative_load_do_update_ffr

Unpredictable set to FALSE of the FFR element associated to SVE speculative load with an async tag check fail TAGS **UNPREDICTABLE**.

ArchMsg.Warning.unpredictable_a32_breakpoint

TAGS **UNPREDICTABLE** DISPLAY Unpredictable breakpoint on a A32 instruction at %{ADDR1} Offset %{OFFSET} BAS %{BAS} for a %{IS_ADDRESS_MISMATCH_BREAKPOINT:(Mismatch|Match)} Breakpoint number %{N:d} Breakpoint address %{ADDR2}. Fields:

ADDR1 unsigned int

Address.

ADDR2 unsigned int

Breakpoint address.

BAS unsigned int

BAS field.

IS_ADDRESS_MISMATCH_BREAKPOINT bool

Is a Mismatch breakpoint, not a Match breakpoint.

N unsigned int

Breakpoint number.

OFFSET unsigned int

Always has a value of zero.

ArchMsg.Warning.unpredictable_banked_register_access

TAGS **UNPREDICTABLE** DISPLAY Access to banked register (R=%{R:d} SYSm=%{SYSM:2x}) is unpredictable. Fields:

R unsigned int

R bit.

SYSM unsigned int

SYSm.

ArchMsg.Warning.unpredictable_ccfail_undef

TAGS **UNPREDICTABLE** DISPLAY An instruction %{OPCODE} at VA=%{VADDR} failed its condition codes check but would otherwise have caused a non-data-dependent trap or undefined exception. Behaviour is implementation-defined choice of exception or nop. Fields:

OPCODE unsigned int

opcode of the instruction.

VADDR unsigned int

Virtual address of the instruction being executed.

ArchMsg.Warning.unpredictable_ici_bits

TAGS **UNPREDICTABLE** DISPLAY ICI bits do not apply or are architecturally unpredictable for instruction.

ArchMsg.Warning.unpredictable_t32_breakpoint

TAGS **UNPREDICTABLE** DISPLAY Unpredictable breakpoint on a T32 instruction at %{ADDR1} Offset %{OFFSET} BAS %{BAS} for a %{IS_ADDRESS_MISMATCH_BREAKPOINT:(Mismatch|Match)} Breakpoint number %{N:d} Breakpoint address %{ADDR2}. Fields:

ADDR1 unsigned int

Address.

ADDR2 unsigned int

Breakpoint address.

BAS unsigned int

BAS field.

IS_ADDRESS_MISMATCH_BREAKPOINT bool

Is a Mismatch breakpoint, not a Match breakpoint.

N unsigned int

Breakpoint number.

OFFSET unsigned int

Indicates which halfword of a 32-bit Thumb instruction the breakpoint is set on. 0 for the first halfword or if the instruction is 16 bits wide, or 1 for the second halfword.

ArchMsg.Warning.unpredictable_t32_vector_catch

TAGS **UNPREDICTABLE** DISPLAY Unpredictable Vector catch exception on a T32 instruction at %{ADDR} Offset %{OFFSET}. Fields:

ADDR unsigned int

Address.

OFFSET unsigned int

Indicates which halfword of a 32-bit Thumb instruction the vector catch exception is set on. 0 for the first halfword or if the instruction is 16 bits wide, or 1 for the second halfword.

ArchMsg.Warning.unpredictable_watchpoint_far

TAGS **UNPREDICTABLE** DISPLAY The value for FAR/EDWAR for hitting this watchpoint may be anywhere in the range %{LOWER_BOUND} to %{UPPER_BOUND}. Fields:

LOWER_BOUND unsigned int

the lowest address accessed by the instruction that triggered the watchpoint.

UPPER_BOUND unsigned int

the highest watchpointed address accessed by that instruction.

ArchMsg.Warning.unpredictable_watchpoint_far_isscmp

DISPLAY The value for FAR/EDWAR for hitting this watchpoint may be anywhere in the range %{LOWER_BOUND} to %{UPPER_BOUND}. Fields:

LOWER_BOUND unsigned int

the lowest address accessed by the instruction that triggered the watchpoint.

UPPER_BOUND unsigned int

the highest watchpointed address accessed by that instruction.

ArchMsg.Warning.warning_AdvSIMDExpandImmUnexpectedZero

TAGS UNPREDICTABLE DISPLAY AdvSIMDExpandImm may treat this immediate value as UNPREDICTABLE.

ArchMsg.Warning.warning_ConditionalSMC

TAGS UNPREDICTABLE DISPLAY SMC instruction has UNPREDICTABLE effects when conditional and when combined with traps.

ArchMsg.Warning.warning_TLBID_out_of_range

TLBID out of range in TLBI according to the configuration of NIS, NVIS, NOS and NVOS. Fields:

NIS unsigned int

NIS.

NOS unsigned int

NOS.

NVIS unsigned int

NVIS.

NVOS unsigned int

NVOS.

TLBID unsigned int

The TLBID value used in the TLBI instruction.

ArchMsg.Warning.warning_TTL_or_TG_mismatches_current_translation_granule

TTL or TG mismatches current translation granule in TLBI VA/RVA/IPA/RIPA. Fields:

CURRENT_TG unsigned int

Current TG as indicated by TCR_ELx or VTCR_EL2.

TLBI_TG unsigned int

TG (or TTL[3:2]) used in TLBI instruction.

ArchMsg.Warning.warning_access_crosses_page_boundary

TAGS UNPREDICTABLE DISPLAY Access crosses page boundary.

ArchMsg.Warning.warning_access_crosses_page_boundary_has_fault

TAGS UNPREDICTABLE DISPLAY Access that crosses page boundary has fault(s).

ArchMsg.Warning.warning_access_crosses_page_boundary_has_fault_on_both_pages

TAGS **UNPREDICTABLE** DISPLAY Access that crosses page boundary has a fault on each page.

Fields:

ADDR1 unsigned int

FAR address on first page.

ADDR2 unsigned int

FAR address on second page.

FSTATUS1 unsigned int

Fault status on first page.

FSTATUS2 unsigned int

Fault status on second page.

ArchMsg.Warning.warning_access_crosses_page_boundary_spanning_different_memory

TAGS **UNPREDICTABLE** DISPLAY Access at address %{ADDR} crosses page from MEM_TYPE %{MEMTYPE_PAGE1} to %{MEMTYPE_PAGE2}. Fields:

ADDR unsigned int

address of access crossing page.

MEMTYPE_PAGE1 signed int

First Page's memory type (0-Normal, 1-Non-Normal).

MEMTYPE_PAGE2 signed int

Second Page's memory type (0-Normal, 1-Non-Normal).

ArchMsg.Warning.warning_access_wraps_around_memory

TAGS **UNPREDICTABLE** DISPLAY Access wraps around memory in %{WIDTH:d}bit mode.

Fields:

WIDTH unsigned int

Address width.

ArchMsg.Warning.warning_bcr_linking_status

TAGS **UNPREDICTABLE** DISPLAY DBGBCR.{BT2,BT} combination gives unpredictable behaviour.

ArchMsg.Warning.warning_bcr_mask_reserved

TAGS **UNPREDICTABLE** DISPLAY DBGBCR_ELn.MASK is set to a reserved value.

ArchMsg.Warning.warning_bvr_masked_bit_not_zero

TAGS **UNPREDICTABLE** DISPLAY DBGBVRn_EL1.VA contains a bit masked by DBGBCRn_EL1.MASK that is not zero.

ArchMsg.Warning.warning_ccsidr_unimplemented_level

TAGS **UNPREDICTABLE** DISPLAY CCSIDR read while CSSELR %{CSSELR:x} points to an unimplemented cache level. Fields:

CSSELR unsigned int

current effective value of CSSELR.

ArchMsg.Warning.warning_change_to_ns_when_tge_set

TAGS **UNPREDICTABLE** DISPLAY Attempting to change NS to 1 when HCR.TGE = 1.

ArchMsg.Warning.warning_contiguous_bit_check_abort

TAGS **UNPREDICTABLE** DISPLAY An abort occurred whilst attempting to check TLB entry is contiguous at %{ENTRY_ADDR}. Fields:

ENTRY_ADDR unsigned int

address of conflicting TLB entry.

ArchMsg.Warning.warning_contiguous_bit_error

TAGS **UNPREDICTABLE** DISPLAY TLB entry at %{ENTRY_ADDR} was not contiguous with entry at %{CONTIG_ADDR} Contents are %{ENTRY_DATA} but expected %{CONTIG_DATA}. Fields:

CONTIG_ADDR unsigned int

address of the TLB first read, with which this entry is expected to be contiguous.

CONTIG_DATA unsigned int

expected contents based on the entry at CONTIG_ADDR.

ENTRY_ADDR unsigned int

address of conflicting TLB entry.

ENTRY_DATA unsigned int

contents of conflicting TLB entry.

ArchMsg.Warning.warning_contiguous_bit_error_D128

TAGS **UNPREDICTABLE** DISPLAY TLB entry at %{ENTRY_ADDR} was not contiguous with entry at %{CONTIG_ADDR} Contents are %{ENTRY_DATA_HI} %{ENTRY_DATA_LO} but expected %{CONTIG_DATA}. Fields:

CONTIG_ADDR unsigned int

address of the TLB first read, with which this entry is expected to be contiguous.

CONTIG_DATA unsigned int

expected contents based on the entry at CONTIG_ADDR.

ENTRY_ADDR unsigned int

address of conflicting TLB entry.

ENTRY_DATA_HI unsigned int

Higher bit contents of conflicting TLB entry.

ENTRY_DATA_LO unsigned int

Lower bit contents of conflicting TLB entry.

ArchMsg.Warning.warning_cp10_cp11_mismatch

TAGS **UNPREDICTABLE** DISPLAY Unpredictable configuration of CP10 and CP11 controls at EL %{EL:d}. Fields:

EL unsigned int

EL owning the control.

ArchMsg.Warning.warning_cp10_cp11_reserved_value

TAGS **UNPREDICTABLE** DISPLAY Unpredictable configuration of CPACR CP10 ({CP10}) or CP11 ({CP11}). Fields:

CP10 unsigned int

CP10 access.

CP11 unsigned int

CP11 access.

ArchMsg.Warning.warning_csselr_level_out_of_range

TAGS **UNPREDICTABLE** DISPLAY CSSELR.Level out of range, written value {LEVEL} with only {IMPLEMENTED:d} levels implemented. Fields:

IMPLEMENTED unsigned int

number of cache levels implemented.

LEVEL unsigned int

written value of CSSELR.Level.

TYPE enum

The type of cache selected by CSSELR.InD and TnD.

ArchMsg.Warning.warning_debug_flow_control_bits_not_obeyed

TAGS **UNPREDICTABLE** DISPLAY Flow control bits for {REGISTER} were not obeyed, causing {ERROR}. Fields:

ERROR enum

type of error resulting.

REGISTER enum

register under consideration.

ArchMsg.Warning.warning_debug_register_access_during_reset

TAGS **UNPREDICTABLE** DISPLAY Unpredictable {IS_WRITE:(read|write)} access to debug register offset {OFFSET} during reset. Fields:

IS_WRITE unsigned int

Write Not Read.

OFFSET unsigned int

Register Offset.

ArchMsg.Warning.warning_decode_cps_inconsistent_fields

TAGS **UNPREDICTABLE** DISPLAY The mode setting fields in the CPS instruction are inconsistent.

ArchMsg.Warning.warning_decode_invalid_state

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable in current exception-level/security state.

ArchMsg.Warning.warning_default_cacheable_mmu_on

TAGS **UNPREDICTABLE** DISPLAY Default cacheable is enabled (HCR.DC = 1) while MMU is enabled.

ArchMsg.Warning.warning_default_cacheable_vmmu_off

TAGS **UNPREDICTABLE** DISPLAY Default cacheable is enabled (HCR.DC = 1) while HCR.VM = 0.

ArchMsg.Warning.warning_deprecated_wvr_bit_2_set

TAGS **UNPREDICTABLE** DISPLAY DBGWVR%{N:d}_EL1=%{WVR:h} has bit 2 set, deprecated in ARMv8. Fields:

N unsigned int

index.

WVR unsigned int

DBGWVR.

ArchMsg.Warning.warning_eret_while_software_step_active_pending

Missing ISB between setting MDSCR_EL1.SS and ERET TAGS **UNPREDICTABLE**.

ArchMsg.Warning.warning_exclusive_to_non_normal

TAGS **UNPREDICTABLE** DISPLAY Exclusive Access to Non-Normal Memory %{ADDR}. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_exclusive_to_non_writeback

TAGS **UNPREDICTABLE** DISPLAY Exclusive Access to Non-Writeback-Cacheable Memory %{ADDR}. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_execute_from_device_memory

TAGS **UNPREDICTABLE** DISPLAY An attempt was made to execute from Device memory.

ArchMsg.Warning.warning_illegal_cpsr_mode

TAGS **UNPREDICTABLE** DISPLAY Writing an illegal or unimplement mode (%{NEW_MODE:h}) to CPSR was unpredictable. Fields:

NEW_MODE unsigned int

New value of CPSR.M.

ArchMsg.Warning.warning_illegal_srs_mode

TAGS **UNPREDICTABLE** DISPLAY Illegal or **UNPREDICTABLE** mode (%{MODE:h}) used for SRS instruction. Fields:

MODE unsigned int

mode for Banked SP.

ArchMsg.Warning.warning_implementation_defined_read_debug_register_in_SCS

Read Access to Debug registers from the processor is IMPLEMENTATION DEFINED TAGS IMP_DEF. Fields:

PPB_OFFSET unsigned int
debug register offset.

ArchMsg.Warning.warning_implementation_defined_sequential_security_transitions_supported

The behavior of sequential instruction fetches that cross from non-secure to secure memory and contain an SG instruction is CONSTRAINED UNPREDICTABLE TAGS UNPREDICTABLE.

ArchMsg.Warning.warning_implementation_defined_stack_limit_check_supported

It is IMPLEMENTATION DEFINED whether stack pointer limit checking is performed for this instruction TAGS IMP_DEF. Fields:

ADDRESS unsigned int
Address of instruction.

OPCODE unsigned int
instruction opcode.

ArchMsg.Warning.warning_implementation_defined_taggednonsharedmemorytype

TAGS IMP_DEF DISPLAY Memory attribute has Tagged, non shared memory type; it is IMPLEMENTATION DEFINED whether the memory location is treated as Tagged or non-Tagged.

ArchMsg.Warning.warning_implementation_defined_write_debug_register_in_SCS

Write Access to Debug registers from the processor is IMPLEMENTATION DEFINED TAGS IMP_DEF. Fields:

DATA unsigned int
data attempted to be written.

PPB_OFFSET unsigned int
debug register offset.

ArchMsg.Warning.warning_invalid_tcr_granule

TAGS UNPREDICTABLE DISPLAY TCR.TG%{TG_ID:(0|1)} bits have been set to a granule size %{{REQUEST:d}}K not implemented - using %{{SUBSTITUTE:d}}K. Fields:

REQUEST signed int
page size requested (or 0 for reserved).

SUBSTITUTE signed int
best guess available page size.

TG_ID bool
bits TG0 or TG1.

ArchMsg.Warning.warning_load_multiple_user_registers_from_user_mode

TAGS UNPREDICTABLE DISPLAY An LDM(user registers) instruction executed from user mode.

ArchMsg.Warning.warning_mte_in_mem_attr_not_supported

TAGS UNPREDICTABLE DISPLAY Memory attribute %{ATTR} has Tagged memory type but Armv8.5-MTE is not implemented. Fields:

ATTR unsigned int

Memory attribute.

ArchMsg.Warning.warning_reserved_breakpoint_state_match

TAGS UNPREDICTABLE DISPLAY %{IS_BREAKPOINT:(Watchpoint|Breakpoint)} programmed with a reserved combination of HMC, SSC and %{IS_BREAKPOINT:(PAC|PMC)}. Fields:

IS_BREAKPOINT unsigned int

Is a breakpoint, not a watchpoint.

ArchMsg.Warning.warning_shareability

TAGS UNPREDICTABLE DISPLAY Unpredictable: combination of the 1st and 2nd stages of translation is Normal Inner Non-Cacheable, Outer Non-Cacheable.

ArchMsg.Warning.warning_software_step_set_while_enabled

TAGS UNPREDICTABLE DISPLAY MDSCR_EL1.SS set to 1 while software step debug exceptions are enabled.

ArchMsg.Warning.warning_thumb_instruction_wraps_around_memory

TAGS UNPREDICTABLE DISPLAY Thumb instruction wraps around memory.

ArchMsg.Warning.warning_ttbr_sbz_bits_are_not_zero

TAGS UNPREDICTABLE DISPLAY %{TTBR}[%{MSB:d}:0] should be zero, but are not. Fields:

MSB unsigned int

x-1.

TTBR enum

which TTBR register.

ArchMsg.Warning.warning_unaligned_address_dbgdtrrx_write

TAGS UNPREDICTABLE DISPLAY Unaligned address %{ADDR} in X0/R0 when DBGDTRRX written in memory access mode. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unaligned_address_dbgdtrtx_read

TAGS UNPREDICTABLE DISPLAY Unaligned address %{ADDR} in X0/R0 when DBGDTRTX read in memory access mode. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unaligned_to_device

TAGS UNPREDICTABLE DISPLAY Unaligned Access to Device Memory %{ADDR}. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unaligned_to_strongly_ordered

TAGS **UNPREDICTABLE** DISPLAY Unaligned Access to Strongly Ordered Memory %{ADDR}.

Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unknown_sau_rnr

TAGS **UNPREDICTABLE** DISPLAY SAU_RNR was set to an unsupported value.

ArchMsg.Warning.warning_unpredictable_AIRCR_PRIS_and_BFHFNMINs

TAGS **UNPREDICTABLE** DISPLAY The effect of setting both AIRCR.BFHFNMINs and AIRCR.PRIS to 1 is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_AIRCR_VECTCLRACTIVE_when_not_in_debug

TAGS **UNPREDICTABLE** DISPLAY The effect of writing a 1 to AIRCR.VECTCLRACTIVE if the processor is not halted in Debug state is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_AIRCR_VECTRESET_and_SYSRESETREQ

TAGS **UNPREDICTABLE** DISPLAY When the processor is halted in Debug state, if a write to the register writes a 1 to both VECTRESET and SYSRESETREQ, the behavior is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_AIRCR_VECTRESET_when_not_in_debug

TAGS **UNPREDICTABLE** DISPLAY The effect of writing a 1 to AIRCR.VECTRESET if the processor is not halted in Debug state is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_AIRCR_incorrect_VKEY

TAGS **UNPREDICTABLE** DISPLAY The value 0x05FA must be written to AIRCR.VKEY, otherwise the register write is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_EXC_RETURN_Reserved_Bit

EXC_RETURN[23:7] are reserved with the special condition that all bits should be written as one. Values other than all 1s are **UNPREDICTABLE**. TAGS **UNPREDICTABLE**. Fields:

EXC_RETURN unsigned int

Special PC value.

SBO_MASK unsigned int

Which bits Should Be One to not UNPRED.

ArchMsg.Warning.warning_unpredictable_PMEVTYPEPERn_ELO_VS_reserved_value

TAGS **UNPREDICTABLE** DISPLAY Unpredictable reserved value of PMEVTYPER<n>_ELO.VS.

ArchMsg.Warning.warning_unpredictable_change_to_DEMCR_MON_STEP_at_insufficient_priority

TAGS **UNPREDICTABLE** DISPLAY The effect of changing DEMCR.MON_STEP at an execution priority that is lower than the priority of the DebugMonitor exception is **CONSTRAINED UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_change_to_priority_of_active_exception

TAGS UNPREDICTABLE DISPLAY Changing the priority of an active exception is UNPREDICTABLE.

ArchMsg.Warning.warning_unpredictable_clear_lspact

TAGS UNPREDICTABLE DISPLAY UNPREDICTABLE state on context switch. Fields:

DRVEXC_TAKEN unsigned int

Derived exception taken, not pended.

REACHED_LAST_STORE unsigned int

Only the last faulted.

ArchMsg.Warning.warning_unpredictable_dbg_exit_unaligned_dlr

TAGS UNPREDICTABLE DISPLAY Debug exit to AArch32 with DLR[0] set to 1. PC[0] can be set to 0 or to DLR[0].

ArchMsg.Warning.warning_unpredictable_exception_catch

TAGS UNPREDICTABLE DISPLAY Generation of an exception catch event is unpredictable.

ArchMsg.Warning.warning_unpredictable_exception_return_inconsistent_state

TAGS UNPREDICTABLE DISPLAY state on exception return is unpredictable.

ArchMsg.Warning.warning_unpredictable_exception_return_instruction

TAGS UNPREDICTABLE DISPLAY exception return instruction is unpredictable.

ArchMsg.Warning.warning_unpredictable_in_debug_state

TAGS UNPREDICTABLE DISPLAY Instruction is UNPREDICTABLE when executed in debug state.

ArchMsg.Warning.warning_unpredictable_mair_encoding

TAGS UNPREDICTABLE DISPLAY Memory attribute %{ATTR} has invalid MAIR encoding; this may lead to unpredictable results. Fields:

ATTR unsigned int

Memory attribute.

ArchMsg.Warning.warning_unpredictable_pmu_counter_access

TAGS UNPREDICTABLE DISPLAY Access to PMU counters from non-secure EL0 or EL1 is UNPREDICTABLE with MDCR_EL2.HPMN set to 0.

ArchMsg.Warning.warning_unpredictable_prioritization_breakpoint_match_vector_catch

TAGS UNPREDICTABLE DISPLAY Two events with the same priority occurred at the same instruction: Address Matching Vector Catch debug event, and Breakpoint debug event. It is constrained unpredictable which is taken.

ArchMsg.Warning.warning_unpredictable_stack_selection

TAGS UNPREDICTABLE DISPLAY UNPREDICTABLE state on context switch.

ArchMsg.Warning.warning_unpredictable_tsize_out_of_range

TAGS UNPREDICTABLE DISPLAY %{TCR}.T%{TTBR:d}SZ (%{TSIZE:d} bits) is out of range. Must be between 25 and %{TSIZE_MAX:d}. Fields:

TCR enum

Which TCR.

TSize signed int

Value of TSize.

TSize_MAX signed int

The maximum allowed value.

TTBR unsigned int

Which TTBR region.

ArchMsg.Warning.warning_unpredictable_unaligned_pc_as_base_register

TAGS **UNPREDICTABLE** DISPLAY The PC value must be word-aligned, otherwise the behavior of the instruction is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_unaligned_pop_stack

TAGS **UNPREDICTABLE** DISPLAY **CONSTRAINED UNPREDICTABLE** unaligned stack frame `%{FRAME_ADDR:x}` in exception-return. Fields:

FRAME_ADDR unsigned int

unaligned stack frame address.

ArchMsg.Warning.warning_unpredictable_unaligned_to_non_writeback

TAGS **UNPREDICTABLE** DISPLAY Unaligned Access to Non-Writeback-Cacheable Memory.

ArchMsg.Warning.warning_unpredictable_vmsa_memattrib

TAGS **UNPREDICTABLE** DISPLAY Unpredictable vmsa memory attribute with texcb value programmed to `%{TEXCB}` when tex remap is `%{TEX_REMAP}`. Fields:

TEXCB unsigned int

texcb value.

TEX_REMAP bool

use tex remap.

ArchMsg.Warning.warning_unpredictable_vtcr_t0sz_sl0

TAGS **UNPREDICTABLE** DISPLAY The combination of VTCR.TOSZ=`%{TOSZ:d}` and VTCR.SL0=`%{SL0:d}` is unpredictable. Fields:

SL0 signed int

VTCR.SL0, starting level for stage 2 translation table walks.

TOSZ signed int

VTCR.TOSZ, required input address range.

ArchMsg.Warning.warning_unpredictable_write_DHCSR

TAGS **UNPREDICTABLE** DISPLAY **UNPREDICTABLE** set of bit-changes in DHCSR. Fields:

NEW unsigned int

data attempted to be written.

OLD unsigned int

previous value.

ArchMsg.Warning.warning_unpredictable_write_SHCSR

TAGS UNPREDICTABLE DISPLAY UNPREDICTABLE setting a pending bit to 1 for an exception with priority \geq execution priority.

ArchMsg.Warning.warning_unsupported_access_to_memory_mapped_register

TAGS UNPREDICTABLE DISPLAY Access to memory mapped register is unsupported.

ArchMsg.Warning.warning_user_jmcr_access

TAGS UNPREDICTABLE DISPLAY User %{WRITE:(write|read)} access to JMCR is unpredictable. Fields:

WRITE unsigned int
access is write.

ArchMsg.Warning.warning_wcr_mask_and_bas

TAGS UNPREDICTABLE DISPLAY DBGWCR_EL1.MASK is non-zero and BAS is not 0xff.

ArchMsg.Warning.warning_wcr_mask_reserved

TAGS UNPREDICTABLE DISPLAY DBGWCR_ELn.MASK is set to a reserved value.

ArchMsg.Warning.warning_wcr_non_configuous_bas

TAGS UNPREDICTABLE DISPLAY DBGWCR_EL1.BAS has non-contiguous set of ones.

ArchMsg.Warning.warning_wvr_masked_bit_not_zero

TAGS UNPREDICTABLE DISPLAY DBGWVRn_EL1.VA contains a bit masked by DBGWCRn_EL1.MASK that is not zero.

ArchMsg.Why.why_branch_target_exception

A branch target exception has occurred. DISPLAY branch target exception: target pc=%{PC}, target opcode=%{OPCODE}. PRIMARY KEY PC. Fields:

BTYPE unsigned int
pstate.BTYPE at time branch taken.

OPCODE unsigned int
opcode at branch target.

PC unsigned int
Virtual Address of branch target.

ArchMsg.Why.why_illegal_state

An illegal mode change has occurred. DISPLAY illegal mode change: %{MESSAGE}. Fields:

MESSAGE string
The message.

BRANCH_MISPREDICT

Simulating branch mispredict. Fields:

PC unsigned int
Origin address (or 0 if unavailable).

BRA_DIR

Direct branches, to immediate address. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISET enum

The instruction set of the branch instruction.

IS_COND bool

Indicates if this is a conditional branch.

IS_HINTED bool

Whether the conditional instruction is hinted.

IS_ISB bool

Whether the instruction is ISB.

IS_LINK bool

Indicates if this is a branch with link.

PC unsigned int

The address of the branch instruction.

TARGET_ISET enum

The instruction set after the branch.

TARGET_PC unsigned int

The address the instruction branches to.

BRA_INDIR

Indirect branches, perhaps to a register. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISET enum

The instruction set of the branch instruction.

IS_COND bool

Indicates if this is a conditional branch.

IS_HINTED bool

Whether the conditional instruction is hinted.

IS_ISB bool

Whether the instruction is ISB.

IS_LINK bool

Indicates if this is a branch with link.

PC unsigned int

The address of the branch instruction.

TARGET_ISET enum

The instruction set after the branch.

TARGET_PC unsigned int

The address the instruction branches to.

BRA_INDIR_CCFAIL

Indirect branches fail, perhaps to a register.

BRA_RET_CCFAIL

Branch return fail, perhaps to a register.

CACHE_MAINTENANCE_OP

Cache Maintenance Operation. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

DATA unsigned int

Specified MVA or Set/way.

FUNCTION enum

Clean / Invalidate.

SCOPE enum

Affected region.

SIDE enum

Inst / Data.

CCFAIL

Conditional instruction condition check fail. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CCFAIL_UNC

Unconditional instruction condition check fail. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CCPASS

Conditional instruction condition check pass. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CCPASS_UNC

Unconditional instruction condition check pass. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CODE_CACHE_MAINT

Code cache maintenance. Fields:

END_INCL unsigned int

Inclusive end address for VA or PA requests.

NS bool

Non-secure world for PA.

START unsigned int

Start address for VA or PA requests.

TYPE enum

Request type.

COMPILE_BLOCK_END

Last instruction of basic block translated. Fields:

VADDR unsigned int

Address of next instruction after this basic block.

COMPILE_BLOCK_START

First instruction of basic block translated. Fields:

VADDR unsigned int

Address of first instruction in basic block.

COMPILE_INST

ARM instruction compiled. Fields:

DISASS string

Disassembly of the instruction.

ISSET enum

The instruction set of this instruction.

ITSTATE unsigned int

The ITSTATE current for the instruction.

OPCODE unsigned int

The opcode of the instruction.

PC unsigned int

The address of the instruction.

SIZE unsigned int

The size of the instruction in bytes.

CONTEXTIDR

Context ID Register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

NS bool

Secure or nonsecure banked register is accessed.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The new value written.

CONTEXT_SYNC

Called for every context synchronization event. Fields:

ARCH_CSE bool

Architectural Context Synchronization Event.

PC unsigned int

The address of the synchronization instruction.

SYSREG_SYNC bool

System register synchronization event.

CORE_ENDIAN

Core BE8 Big-Endian state changed. Fields:

BE8 bool

Core BE8 Big-Endian state.

CORE_INFO

Static processor attributes. Only triggered by a call to DumpState(). Fields:

ARCH_PROFILE enum

The architecture profile of the core.

CLUSTER_ID unsigned int

The cluster ID of this processor.

CORE_NUM unsigned int

The number of this core in an MP processor.

DETAILED_CORE_FEATURES enum

Detailed core features as a bitfield. Enum values are bitmasks to extract the info from the field.

FPU_VERSION enum

The VFP version implemented by the core.

MEM_ARCH enum

The memory architecture of the core.

MIDR_EL1 unsigned int

MIDR_EL1 value.

NUM_CORES unsigned int

The number of cores in this MP processor.

QUANTUM_SIZE unsigned int

The default quantum size of the core (in ticks).

SECURITY_FEATURES bool

Does the core have security features?.

CORE_LOADS

Processor load accesses. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ACQREL enum

Is this an acquire/release.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DATA unsigned int

The data read or written.

ELEMENT_SIZE unsigned int

Width of each element.

ENDIANNESS enum

Byte ordering of this access.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

LOCK enum

Normal, exclusive or locked access.

MECID unsigned int

Memory Encryption Context ID, if applicable.

MECID2 unsigned int

Memory Encryption Context ID of the second page, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PADDR_TAG unsigned int

If tag checked, the physical address tag for the access.

PAGE_SIZE unsigned intPage size as $\log_2(\text{size in bytes})$.**PAS enum**

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK, 2=Exclusive Failed, 3=Unsupported.

SIZE unsigned int

Width of the access in bytes. Only required if DATA is not traced.

TAG_CHECKED bool

Is this access tag checked.

TAG_CHECKED2 bool

Is the second page access tag checked.

TRANS bool

Is this a translated access.

VADDR unsigned int

The virtual address of the access.

CORE_REGS

Changes of the core registers R0 to R14. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number, 0 to 14.

MODE enum

Bank of the register accessed.

OLD_VALUE unsigned int

The old value overwritten.

PHYS_ID enum

The physical register accessed.

VALUE unsigned int

The new value written to the register.

CORE_REGS64

Changes of the core registers X0..X30, SP_ELn. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ID enum

The register number, 0-30 for X0-X30, >=32 for SP_ELn.

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

CORE_REGS64_READ

General purpose AArch64 register has been read X0..X30, SP_ELn. Fields:

ID enum

The register number, 0-30 for X0-X30, >=32 for SP_ELn.

VALUE unsigned int

The value read from the register.

CORE_STALLED_TRANSITION

Detect when a core is prevented from progressing PChannel states. Fields:

CURRENT_STATE enum

Current power mode of core.

REASON string

The reason a transition was stalled, and what must be done to unblock it.

TARGET_STATE enum

Desired power mode of core after transition.

CORE_STORES

Processor store accesses. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ACQREL enum

Is this an acquire/release.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DATA unsigned int

The data read or written.

ELEMENT_SIZE unsigned int

Width of each element.

ENDIANNESS enum

Byte ordering of this access.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

LOCK enum

Normal, exclusive or locked access.

MECID unsigned int

Memory Encryption Context ID, if applicable.

MECID2 unsigned int

Memory Encryption Context ID of the second page, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PADDR_TAG unsigned int

If tag checked, the physical address tag for the access.

PAGE_SIZE unsigned int

Page size as $\log_2(\text{size in bytes})$.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK, 2=Exclusive Failed, 3=Unsupported.

SIZE unsigned int

Width of the access in bytes. Only required if DATA is not traced.

TAG_CHECKED bool

Is this access tag checked.

TAG_CHECKED2 bool

Is the second page access tag checked.

TRANS bool

Is this a translated access.

VADDR unsigned int

The virtual address of the access.

CP14_READ

System Coprocessor register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP14 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc1 unsigned int

opcode 1.

opc2 unsigned int

opcode 2.

CP14_WRITE

System Coprocessor register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP14 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc1 unsigned int

opcode 1.

opc2 unsigned int

opcode 2.

CP15_READ

System Coprocessor register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP15 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc1 unsigned int

opcode 1.

opc2 unsigned int

opcode 2.

CP15_READ64

System Coprocessor register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP15 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc unsigned int

opcode 1.

CP15_WRITE

System Control Coprocessor register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP15 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc1 unsigned int

opcode 1.

opc2 unsigned int

opcode 2.

CP15_WRITE64

System Coprocessor register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP15 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc unsigned int

opcode 1.

CPSR

CPSR change. (In AArch64 PSTATE represented in SPSR format). Fields:

CORE_NUM unsigned int

Core number in a multi processor.

OLD_VALUE unsigned int

The old CPSR value.

UNKNOWN unsigned int

Bits within the register that have unknown value.

VALUE unsigned int

The new CPSR value.

CRYPTO_SPEC

Every crypto instruction speculatively executed.

CorePowerStateChange

Core power-up state has changed. DISPLAY Core powered %{PU:(up|down)}. Fields:

PU bool

Whether the core powered up.

DATA_CACHE_SET_ALLOC_TAGS

Set allocation tags from DC GVA instruction. Fields:

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

NSDESC unsigned int

The security state of the access.

PADDR unsigned int

The physical address of the access.

PAS enum

The physical address space of the page.

RESPONSE enum

0=Aborted, 1=OK.

SIZE unsigned int

The size of memory in bytes.

TAG unsigned int

The value of the tag being set.

TAGGING_TYPE enum

Type of Memory tagging (Physical or Virtual).

TAG_VADDR unsigned int

The virtual address of the Virtual Tag if Virtual tagging is used; otherwise 0.

VADDR unsigned int

Virtual address used for setting allocation tags.

DATA_CACHE_ZERO

Zero and invalidate cache line from DC ZVA instruction. Fields:

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

NSDESC unsigned int

The security state of the access.

PADDR unsigned int

The physical address of the access.

PADDR_TAG unsigned int

If tag checked, the physical address tag for the access.

PAS enum

The physical address space of the page.

RESPONSE enum

0=Aborted, 1=OK.

SIZE unsigned int

The size of zero'd memory in bytes.

TAG_CHECKED bool

Is this access tag checked.

VADDR unsigned int

Virtual address of the zero'd cache line.

DEBUG_EVENT

Hardware debug support event. Fields:

EVENT enum

Description of event.

VALUE unsigned int

data value.

DEBUG_TARGET_ISET

Hardware debug target ISA. Fields:

TARGET_ISET enum

The target instruction set of the debug state.

DMI_ALLOCATE

DMI allocated. Fields:

ACCESS_DIRECTION enum

READ / WRITE.

PADDR unsigned int

The physical address of the DMI.

PTR unsigned int

DMI range pointer.

SIDE enum

DSIDE / ISIDE.

SIZE unsigned int

The size of DMI.

VADDR unsigned int

The virtual address of the DMI..

DMI_HIT

DMI hit, access used DMI (only fire for small DMI regions i.e. less than a full page). Fields:

ACCESS_DIRECTION enum

READ / WRITE.

PADDR unsigned int

The physical address of the DMI.

SIDE enum

DSIDE / ISIDE.

SIZE unsigned int

The size of the DMI.

VADDR unsigned int

The virtual address of the DMI..

DMI_REVOKE

DMI revoked. Fields:

ACCESS_DIRECTION enum

READ / WRITE.

PADDR unsigned int

The physical address of the DMI.

SIDE enum

DSIDE / ISIDE.

SIZE unsigned int

The size of DMI.

VADDR unsigned int

The virtual address of the DMI..

END_COMPILE

Compilation end. Fields:

END_OF_PAGE_COUNT unsigned int

Number of basic blocks exited due to page boundary since START_COMPILE.

FETCHFAIL_COUNT unsigned int

Number of basic blocks exited due to ifetch failure START_COMPILE.

INST_COUNT unsigned int

Number of instructions compiled since START_COMPILE.

NONSEQ_COUNT unsigned int

Number of basic blocks exited due to non-sequential instructions since START_COMPILE.

PAGE_STRADDLE_COUNT unsigned int

Number of basic blocks exited due to unaligned instructions crossing page since START_COMPILE.

ESB

Error Synchronization Barrier hit. Fields:

ASYNC bool

ESB is async.

IMPLICIT bool

ESB is implicit.

EXCEPTION

Exception taken. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ESR unsigned int

The value assigned to ESR when exception is taken.

ESR_EC unsigned int

The Value assigned to ESR.EC when exception is taken.

IS_PHYSICAL bool

Physical or Virtual exception.

IS_TRAP bool

Exception coming from a trap.

LR unsigned int

The value assigned to the link register.

PC unsigned int

The location where the exception occurred.

PREFERRED_RETURN unsigned int

The preferred return address for the exception.

TAKEN_LOCALLY bool

Whether exception is taken locally(same exception level).

TARGET_ISET enum

The instruction set of the exception handler code.

TARGET_PC unsigned int

The address the exception branches to.

VECTOR enum

The exception vector.

VECTOR_OFFSET unsigned int

The offset of an exception vector from the base address.

EXCEPTION_END

Every exception completed.

EXCEPTION_INFO

Trace human-readable reason of the current exception for data/instruction aborts, floating point exceptions and unknown exceptions. Fields:

DESCRIPTION string

Reason for current exception.

ESR_EC unsigned int

The value of ESR.EC when exception is taken.

EXCEPTION_RAISE

Every exception raised.

EXCEPTION_RETURN

Branches on leaving exception. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISET enum

The instruction set of the branch instruction.

IS_COND bool

Indicates if this is a conditional branch.

IS_HINTED bool

Whether the conditional instruction is hinted.

IS_ISB bool

Whether the instruction is ISB.

IS_LINK bool

Indicates if this is a branch with link.

PC unsigned int

The address of the branch instruction.

TARGET_ISET enum

The instruction set after the branch.

TARGET_PC unsigned int

The address the instruction branches to.

EXCEPTION_RETURN_PREBRANCH

Exception return event that must occur before PC and processor state have updated.

EXCEPTION_START

Every exception started.

FIQ_TAKEN

FIQ taken exception.

FP_EXCEPTIONS_SIGNALLED

Trace which FP exceptions were signalled as a part of the FP operations performed by an instruction. Fields:

FP_EXCEPTIONS_SIGNALLED_MASK unsigned int

Indicates which FP exceptions were signalled by an instruction. This uses the same encoding as the cumulative floating-point exception bitfields in FPSR.

FP_STATE

Floating point state. Fields:

FPSR unsigned int

FPSR register value.

MASK unsigned int

The mask indicating updated cumulative bits.

FREQ_CHANGED

Core frequency changed. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The instruction count of this core.

NEW_FREQ signed int

The new frequency of this core (expressed in Hz).

OLD_FREQ signed int

The old frequency of this core (expressed in Hz).

HLT

HLT instruction occurred. Fields:

IMM16 unsigned int

The 16-bit immediate value encoded in HLT instruction.

INST

Every instruction executed. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DEBUG_STATE bool

The instruction is executed in debug state.

DISASS string

Disassembly of instruction.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISSET enum

The current instruction set.

ITSTATE unsigned int

The current ITSTATE.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

MODE enum

The mode the core is in.

NS unsigned int

The core's non-secure bit.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

OPCODE unsigned int

The opcode of the instruction.

PADDR unsigned int

The physical address of the instruction.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the instruction.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the page.

PC unsigned int

The address of the instruction.

SECURITY_STATE enum

The core's security state.

SIZE unsigned int

The size of the instruction in bytes.

INST_END

Every instruction completed.

INST_START

Every instruction started. Fields:

DEBUG_STATE bool

The instruction is executed in debug state.

ISSET enum

The current instruction set.

MODE enum

The mode the core is in.

NS enum

The current Secure State.

PC unsigned int

The address of the conditional instruction.

SECURITY_STATE enum

The current Security State.

INST_STRADDLE

Instruction straddles boundary. Fields:

DEBUG_STATE bool

The instruction is executed in debug state.

ISSET enum

The current instruction set.

MODE enum

The mode the core is in.

NS enum

The current Secure State.

PC unsigned int

The address of the conditional instruction.

SECURITY_STATE enum

The current Security State.

IRQ_TAKEN

IRQ taken exception.

ISB

ISB instruction occurred. Fields:

PC unsigned int

The address of the ISB instruction.

LOAD_MULTIPLE_REGS_LIST

List of destination registers (not source) of a multiple registers load instruction. Fields:

REG string

Register to add into the list.

LOCAL_MONITOR

Local monitor activity. Fields:

PADDR unsigned int

Local Monitor Address.

State enum

State of the monitor (Open/Exclusive).

MAPPING_CREATE

A memory mapping has been created. This is an implementation detail of the model and so may change without notice. Fields:

ASID unsigned int

The ASID used during translation, or 0 if no ASID was used.

IS_VALID bool

Whether the mapping is valid or not.

PADDR unsigned int

Base physical address of region, or 0 if not known.

SIDE enum

I-side or D-side.

SIZE unsigned int

Size of the page, or 0 if not known.

VADDR unsigned int

Base virtual address of region.

VMID unsigned int

The VMID used during translation, or 0 if no VMID was used.

MAPPING_DESTROY

A memory mapping has been destroyed. This is an implementation detail of the model and so may change without notice. Fields:

ASID unsigned int

The ASID used during translation.

CAUSE enum

The cause of invalidation.

PADDR unsigned int

Base physical address of page.

SIDE enum

I-side or D-side.

SIZE unsigned int

Size of the page.

VADDR unsigned int

Base virtual address of page.

VMID unsigned int

The VMID used during translation.

MEMMAP_DEBUG_READ

Memory mapped reads to the debug registers. Fields:

ADDR unsigned int

address.

CORE_NUM unsigned int

Core number in a multi processor.

EXT bool

Whether access is from an external device (such as the DAP).

NS enum

Secure state of the access.

REG_NAME string

Name of the debug register accessed.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

MEMMAP_DEBUG_WRITE

Memory mapped writes to the debug registers. Fields:

ADDR **unsigned int**

address.

CORE_NUM **unsigned int**

Core number in a multi processor.

EXT **bool**

Whether access is from an external device (such as the DAP).

NS **enum**

Secure state of the access.

REG_NAME **string**

Name of the debug register accessed.

UNDEF **bool**

The register accessed is undefined.

VALUE **unsigned int**

The new value written.

MEMTAG_LOADS

Allocation tag memory read; when attributes and system configuration allow tag accesses.

Fields:

ACCESS_TYPE **enum**

The type of instruction performing the access.

ATTR **unsigned int**

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 **unsigned int**

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

NSDESC **unsigned int**

The physical address non-secure bit.

NSDESC2 **unsigned int**

The second page physical address non-secure bit.

NUM_GRANULES **unsigned int**

Number of tags to be Loaded/stored.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK.

TAGGING_TYPE enum

Type of Memory tagging (Physical or Virtual).

TAGS unsigned int

The tags read or written. Tags for subsequent granules would be found at subsequent bytes in this field (One byte for each tag).

TAG_VADDR unsigned int

The virtual address of the Virtual Tag if Virtual tagging is used; otherwise 0.

VADDR unsigned int

The virtual address of the access.

MEMTAG_LOAD_INST

Allocation tag load instruction; regardless of tags being actually accessed. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

NUM_GRANULES unsigned int

Number of tags to be Loaded/stored.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK.

TAGGING_TYPE enum

Type of Memory tagging (Physical or Virtual).

TAGS unsigned int

The tags read or written. Tags for subsequent granules would be found at subsequent bytes in this field (One byte for each tag).

TAG_VADDR unsigned int

The virtual address of the Virtual Tag if Virtual tagging is used; otherwise 0.

VADDR unsigned int

The virtual address of the access.

MEMTAG_STORES

Allocation tag memory write; when attributes and system configuration allow tag accesses.

Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

NUM_GRANULES unsigned int

Number of tags to be Loaded/stored.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK.

TAGGING_TYPE enum

Type of Memory tagging (Physical or Virtual).

TAGS unsigned int

The tags read or written. Tags for subsequent granules would be found at subsequent bytes in this field (One byte for each tag).

TAG_VADDR unsigned int

The virtual address of the Virtual Tag if Virtual tagging is used; otherwise 0.

VADDR unsigned int

The virtual address of the access.

MEMTAG_STORE_INST

Allocation tag store instruction; regardless of tags being actually accessed. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

NUM_GRANULES unsigned int

Number of tags to be Loaded/stored.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK.

TAGGING_TYPE enum

Type of Memory tagging (Physical or Virtual).

TAGS unsigned int

The tags read or written. Tags for subsequent granules would be found at subsequent bytes in this field (One byte for each tag).

TAG_VADDR unsigned int

The virtual address of the Virtual Tag if Virtual tagging is used; otherwise 0.

VADDR unsigned int

The virtual address of the access.

MMU_D128_TTB_READ

This event is triggered by reads caused by a D128 translation table walk. Fields:

ABORTED bool

Was the walk successful.

DATA_HIGH unsigned int

The data upper bits read.

DATA_LOW unsigned int

The data lower bits read.

IPA unsigned int

For stage 1, the IPA of the read.

LEVEL unsigned int

Translation table level.

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

OUTADDR unsigned int

The output physical address.

PADDR unsigned int

The physical address of the read.

PASmem enum

Physical Address Space of the accessed memory.

PASreq enum

Physical Address Space of the requested lookup.

REGIME_EL enum

Entry matches in this translation regime.

REQUESTER enum

The requester of this table walk.

SIDE enum

Inst / Data.

STAGE unsigned int

Translation stage.

MMU_D128_TTB_WRITE

This event is triggered by writes caused by a D128 translation table walk. Fields:

ABORTED bool

Was the walk successful.

DATA_HIGH unsigned int

The data upper bits written.

DATA_LOW unsigned int

The data lower bits written.

IPA unsigned int

For stage 1, the IPA of the write.

IS_HACDBS_ACCESS bool

If the access was due to HACDBS_CLEAN.

LEVEL unsigned int

Translation table level.

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

OUTADDR unsigned int

The output physical address.

PADDR unsigned int

The physical address of the write.

PASmem enum

Physical Address Space of the accessed memory.

PASreq enum

Physical Address Space of the requested lookup.

REGIME_EL enum

Entry matches in this translation regime.

SIDE enum

Inst / Data.

STAGE unsigned int

Translation stage.

MMU_TRANS

Address translation information. Fields:

ASID unsigned int

Address space identifier.

CORE_NUM unsigned int

Core number in a multi processor.

Hyp bool

Entry matches in Hyp state only.

INNERCACHE_RA bool

Is the inner cache allocate on read.

INNERCACHE_TRANSIENT bool

Is the inner write-through transient.

INNERCACHE_TYPE enum

Inner Caching scheme (NC/MB/WA).

INNERCACHE_WA bool

Is the inner cache allocate on write.

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MEMTYPE enum

Memory type.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC enum

Is secure side supposed to access secure or nonsecure memory.

OUTERCACHE_RA bool

Is the outer cache allocate on read.

OUTERCACHE_TRANSIENT bool

Is the outer write-through transient.

OUTERCACHE_TYPE enum

Outer Caching scheme (NC/MB/WA).

OUTERCACHE_WA bool

Is the outer cache allocate on write.

PADDR unsigned int

Physical address of the access.

PAGESIZE unsigned int

Page size as $\log_2(\text{size})$.

PAS enum

PAS supposed to be accessed is secure or nonsecure memory or Root/Realm if RME is implemented.

SH enum

Shareability.

SIDE enum

Inst / Data.

STAGE1_PERM unsigned int

Stage 1 permissions, [5:3] Privileged access (XWR) [2:0] User access (XWR).

STAGE2_PERM unsigned int

Stage 2 permission mask [5:3] Privileged access (XWR) [2:0] User access (XWR).

VADDR unsigned int

Virtual address of the access.

VMID unsigned int

Virtual machine identifier.

XS bool

XS Attribute.

nG enum

Flag indicating whether ASID will be matched.

MMU_TTB_READ

This event is triggered by reads caused by a translation table walk. Fields:

ABORTED bool

Was the walk successful.

DATA unsigned int

The data read.

DESC_KIND enum

The encoding scheme of the descriptors used for this walk.

IPA unsigned int

For stage 1, the IPA of the read.

LEVEL unsigned int

Translation table level.

LPAE bool

Is this for an LPAE translation (*DEPRECATED*, use DESC_KIND).

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSmem bool

Non secure memory access.

NSreq bool

Non secure request.

OUTADDR **unsigned int**

The output physical address.

PADDR **unsigned int**

The physical address of the read.

PASmem **enum**

Physical Address Space of the accessed memory.

PASreq **enum**

Physical Address Space of the requested lookup.

REGIME_EL **enum**

Entry matches in this translation regime.

REQUESTER **enum**

The requester of this table walk.

SIDE **enum**

Inst / Data.

STAGE **unsigned int**

Translation stage.

MMU_TTB_WRITE

This event is triggered by writes caused by a translation table walk. Fields:

ABORTED **bool**

Was the walk successful.

DATA **unsigned int**

The data written.

DESC_KIND **enum**

The encoding scheme of the descriptors used for this walk.

IPA **unsigned int**

For stage 1, the IPA of the write.

IS_HACDBS_ACCESS **bool**

If the access was due to HACDBS_CLEAN.

LEVEL **unsigned int**

Translation table level.

LPAAE **bool**

Is this for an LPAAE translation (*DEPRECATED*, use DESC_KIND).

MECID **unsigned int**

Memory Encryption Context(MEC) ID, if applicable.

MPAM_PARTID **unsigned int**

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSmem bool

Non secure memory access.

NSreq bool

Non secure request.

OUTADDR unsigned int

The output physical address.

PADDR unsigned int

The physical address of the write.

PASmem enum

Physical Address Space of the accessed memory.

PASreq enum

Physical Address Space of the requested lookup.

REGIME_EL enum

Entry matches in this translation regime.

SIDE enum

Inst / Data.

STAGE unsigned int

Translation stage.

MODE_CHANGE

Mode change. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

MODE enum

The new mode.

NON_SECURE enum

The core's new non-secure bit.

OLD_MODE enum

The old mode.

SECURITY_STATE enum

The new security state.

NOT_TAG_CHECKED_INST

MTE Tag not checked for current instruction.

PERIODIC

Called for every quantum. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The instruction count of this CPU.

PC unsigned int

The address of the next instruction to be executed on this CPU.

PMU_COUNTER_OVERFLOW

PMU counter overflow. Fields:

COUNTER_GROUP enum

Counter group.

COUNTER_RANGE enum

Counter range.

EVENT_ID unsigned int

Event Identifier, field is valid only for Event counter group.

INDEX unsigned int

Counter index (as selected by PMSELR), field is valid only for Event/Cycle counter groups.

INTERRUPT bool

Is interrupt enabled on overflow for this counter.

POINTER_AUTH

Called when execution hits a pointer authentication check. Fields:

AUTH_STATE bool

Authentication result.

KEY_USED enum

Pointer authentication key.

POTINDEX_UPDATE

Value of POT index for current pc changed. Fields:

NEW_POTINDEX unsigned int

New POTIndex.

OLD_POTINDEX unsigned int

Old POTIndex.

PREFETCH_MEMORY64

Prefetch from PRFM or PRFUM instructions. Fields:

PRFOP enum

Prefetch hint.

VADDR unsigned int

Virtual address of the location that should be prefetched.

PRELOAD64_REQUEST

Preload requests from PRFM or PRFUM instructions. Fields:

PRFOP enum

Prefetch hint.

VADDR unsigned int

Virtual address of the location that should be prefetched.

PRELOAD_DATA

Data preload from PLD instruction. Fields:

VADDR unsigned int

Virtual address of the data that should be preloaded.

PRELOAD_INST

Instruction preload from PLI instruction. Fields:

VADDR unsigned int

Virtual address of the instruction that should be preloaded.

PRE_CORE_LOAD

Trace just before a core load. Fields:

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

PRE_CORE_STORE

Trace just before a core store. Fields:

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

PRE_TTB_READ

This event is triggered before reads caused by a translation table walk. Fields:

ADDR unsigned int

The physical address of the read.

PSTATE

PSTATE fields update. Fields:

ALLINT unsigned int

All interrupt mask bit.

BTYPE unsigned int

Branch target identification bit.

CORE_NUM unsigned int

Core number in a multi processor.

DAIF unsigned int

Exception mask bits.

DIT unsigned int

Data Independent Timing bit.

EL unsigned int

Current Exception level field.

EXLOCK unsigned int

Exception return state lock bit.

IL unsigned int

Illegal Execution state bit.

NZCV unsigned int

Condition flags.

PACM unsigned int

PACM bit.

PAN unsigned int

Privileged Access Never state bit.

PM unsigned int

PMU exception mask bit.

PPEND unsigned int

PMU exception pending bit.

SM unsigned int

StreamingMode status bit.

SP unsigned int

Stack pointer register selection bit.

SS unsigned int

Software Step bit.

SSBS unsigned int

Speculative Store Bypass Safe bit.

TCO unsigned int

Tag Check Override bit.

TOUCHED unsigned int

Mask of the fields that have been touched since this trace has last been emitted. Note that touched does not necessarily mean the value changed. The index of the field is the bit index in the mask plus the index of this field plus one.

UAO unsigned int

User Access Override bit.

UINJ unsigned int

Undefined Instruction exceptions bit.

ZA unsigned int

ZA storage and ZT0 register enable bit.

nRW unsigned int

Current Execution state bit.

PSTATE_PACTIVE_OPMODE_TRANSITION

Trace pactive and Opmode mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Operating mode after the transition.

OLD_STATE enum

Operating mode before the transition.

PSTATE_PACTIVE_OPMODE_TRANSITION_DENIED

Trace the denied pactive and Opmode mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Operating mode after the transition.

OLD_STATE enum

Operating mode before the transition.

REASON string

Reason for the transition being denied.

PSTATE_PACTIVE_POWER_TRANSITION

Trace pactive and power mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Power mode after the transition.

OLD_STATE enum

Power mode before the transition.

PSTATE_PACTIVE_POWER_TRANSITION_DENIED

Trace the denied pactive and power mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Power mode after the transition.

OLD_STATE enum

Power mode before the transition.

REASON string

Reason for the transition being denied.

PSTATE_PACTIVE_THREAD_OPMODE_TRANSITION

Trace pactive and Opmode mode transitions for threads on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Operating mode after the transition.

OLD_STATE enum

Operating mode before the transition.

QUANTUM_START

Called at the start of every quantum.

RANGE_PRELOAD_REQUEST

Preload requests from RPRFM. Fields:

COUNT unsigned int

Number of blocks of data to access.

LENGTH unsigned int

Number of contiguous bytes to be accessed in each block.

OP unsigned int

Range prefetch operation .

REUSE unsigned int

Maximum number of bytes to be accessed before next RPRFM .

STRIDE signed int

Stride to advance after accessing LENGTH bytes.

VADDR unsigned int

Base virtual address of range prefetch request.

RUN_STATE

Run state transition. Fields:

INST_COUNT **unsigned int**

Ticks count at point of transition.

NEW **enum**

New run state.

OLD **enum**

Old run state.

SEMIHOSTING_CALL

Call of a semihost function occurred. Fields:

PC **unsigned int**

The program counter after the semihosting call.

SEMIHOSTING_PRECALL

About to call semihost. Fields:

EL2 **bool**

Use EL2 translation regime.

NS **bool**

Is Non-Secure.

REG_WIDTH **unsigned int**

The current register width in bytes.

SIGNAL

External signal state change. Fields:

SIGNAL **enum**

Signal that changed.

STATE **bool**

Signal asserted state.

SOFTWARE_STEP

Return the Debug Software Step state. Fields:

STATE **enum**

Software Step state.

SPSR

SPSR change. Fields:

CORE_NUM **unsigned int**

Core number in a multi processor.

MODE **enum**

Which of the banked SPSR registers is written.

OLD_VALUE unsigned int

The old SPSR value.

VALUE unsigned int

The new SPSR value.

START_COMPILE

Compilation started. Fields:

VADDR unsigned int

Instruction where compilation begins.

STG_MULTIPLE_MICRO_OPS_ADDRESS_RANGE

Address range for MTE stores with multiple uops. Fields:

ADDR_END unsigned int

Final address.

ADDR_START unsigned int

Starting address.

STORE_MULTIPLE_REGS_ADDRESS_RANGE

Address range for stores with multiple registers. Fields:

ADDR_END unsigned int

Final address.

ADDR_START unsigned int

Starting address.

SUBOPTIMAL_LDST_RETIRED

suboptimal load/store. Fields:

ACCESS_SIZE unsigned int

Log2 of the access width in bytes used for alignment checking (1: half-word, 2: word ...).

OPTIMAL_ALIGN_SELECTOR unsigned int

Selector to decide the optimal alignment size (1: access width, 2: 4-bytes, 3: 8-bytes, ..., 12: 4-kbytes).

VADDR unsigned int

The virtual address of the access.

SVE_INST_SPEC

SVE operations speculatively executed.

SVE_LD_RETIRED

SVE load instructions architecturally executed.

SVE_LOADS

SVE load accesses. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ACQREL enum

Is this an acquire/release.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DATA unsigned int

The data read or written.

ELEMENT_SIZE unsigned int

Width of each element.

ENDIANNESS enum

Byte ordering of this access.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

LOCK enum

Normal, exclusive or locked access.

MECID unsigned int

Memory Encryption Context ID, if applicable.

MECID2 unsigned int

Memory Encryption Context ID of the second page, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PADDR_TAG unsigned int

If tag checked, the physical address tag for the access.

PAGE_SIZE unsigned int

Page size as $\log_2(\text{size in bytes})$.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK, 2=Exclusive Failed, 3=Unsupported.

SIZE unsigned int

Width of the access in bytes. Only required if DATA is not traced.

TAG_CHECKED bool

Is this access tag checked.

TAG_CHECKED2 bool

Is the second page access tag checked.

TRANS bool

Is this a translated access.

VADDR unsigned int

The virtual address of the access.

SVE_MEM_PREFETCH

Scalable Vector Extension memory prefetch. Fields:

CACHE_LEVEL enum

The preferred cache level.

HINT enum

What the memory is expected to be used for.

SIZE unsigned int

The prefetch size in bytes.

TEMPORAL bool

Whether the data is temporal.

VADDR unsigned int

The virtual address of the prefetch.

SVE_REG_READ

Scalable Vector Extension register read. Fields:

ID enum

The register identifier.

SM bool

Whether the PE is in Streaming Mode.

VALUE unsigned int

The read data.

SVE_REG_UPDATE

Scalable Vector Extension register update. Fields:

ID enum

The register identifier.

SM bool

Whether the PE is in Streaming Mode.

VALUE unsigned int

The written data.

SVE_REG_WRITE

Scalable Vector Extension register write. Fields:

ID enum

The register identifier.

SM bool

Whether the PE is in Streaming Mode.

VALUE unsigned int

The written data.

SVE_STORES

SVE store accesses. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ACQREL enum

Is this an acquire/release.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DATA unsigned int

The data read or written.

ELEMENT_SIZE unsigned int

Width of each element.

ENDIANNESS enum

Byte ordering of this access.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

LOCK enum

Normal, exclusive or locked access.

MECID unsigned int

Memory Encryption Context ID, if applicable.

MECID2 unsigned int

Memory Encryption Context ID of the second page, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PADDR_TAG unsigned int

If tag checked, the physical address tag for the access.

PAGE_SIZE unsigned int

Page size as \log_2 (size in bytes).

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK, 2=Exclusive Failed, 3=Unsupported.

SIZE unsigned int

Width of the access in bytes. Only required if DATA is not traced.

TAG_CHECKED bool

Is this access tag checked.

TAG_CHECKED2 bool

Is the second page access tag checked.

TRANS bool

Is this a translated access.

VADDR unsigned int

The virtual address of the access.

SVE_ST_RETIRED

SVE store instructions architecturally executed.

SVE_UNALIGNED_LDST_RETIRED

SVE load or store instructions architecturally executed that would cause alignment fault if A bit is set.

SVE_Z_PRED_MEM_ACCESS_INFO

Information about SVE/SME predicated accesses to Z registers (non Gather/Scatter). Fields:

MEM_ELEM_SIZE unsigned int

Width of elements accessed in memory (in bytes).

RnW bool

Whether this is a Read not a Write.

VA unsigned int

Virtual base address computed for accesses.

SYNC

Called for every synchronization, that is every quantum break. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The instruction count of this CPU.

LOCAL_QUANTUM unsigned int

The local quantum of this CPU.

LOCAL_TIME unsigned int

The local time of this CPU.

SYNC_REASON enum

Reason for sync.

SYSCALL

System call instruction executed. Fields:

IMM unsigned int

Immediate value of the system call instruction.

TYPE enum

System call type.

VADDR unsigned int

Instruction that caused the system call.

SYSREG_READ128

System Coprocessor 128 bit register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

REGNUM enum

Internal register number.

REGNUM_PHYS enum

Internal register number physically accessed (e.g. in presense of VHE redirection).

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_READ64

System Coprocessor register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REGNUM enum

Internal register number.

REGNUM_OPERAND enum

Register number corresponding to the instruction operand.

REGNUM_PHYS enum

Internal register number physically accessed (e.g. in presense of VHE redirection).

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_UPDATE128

Triggers when the system updates a register. Fields:

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG enum

Register number.

UNKNOWN unsigned int

Bits of the register which became unknown.

VALUE unsigned int

Value written to the register.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_UPDATE32

Triggers when the system updates a register. Fields:

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG enum

Register number.

UNKNOWN unsigned int

Bits of the register which became unknown.

VALUE unsigned int

Value written to the register.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_UPDATE64

Triggers when the system updates a register. Fields:

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG enum

Register number.

UNKNOWN unsigned int

Bits of the register which became unknown.

VALUE unsigned int

Value written to the register.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_WRITE128

System Coprocessor 128 bit register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

REGNUM enum

Internal register number.

REGNUM_PHYS enum

Internal register number physically accessed (e.g. in presense of VHE redirection).

UNDEF bool

The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int
opcode 2.

SYSREG_WRITE64

System Coprocessor register write. Fields:

CORE_NUM unsigned int
Core number in a multi processor.

CRm unsigned int
CRm.

CRn unsigned int
CRn.

NS bool
Secure or nonsecure banked register is accessed.

REGNUM enum
Internal register number.

REGNUM_OPERAND enum
Register number corresponding to the instruction operand.

REGNUM_PHYS enum
Internal register number physically accessed (e.g. in presense of VHE redirection).

UNDEF bool
The register accessed is undefined.

UPDATED_VALUE unsigned int
Updated value of the register now it has been written.

VALUE unsigned int
The new value written.

opc unsigned int
opcode 1.

opc0 unsigned int
opcode 0.

opc2 unsigned int
opcode 2.

TRBU_BUFFER_WRAP

DISPLAY Trace buffer current write pointer wrapped.

TRBU_TRIGGER_EVENT

DISPLAY Trace buffer Trigger Event.

TRBU_WRITE

DISPLAY PETU trace data has been written to the TRBU memory buffer at %{BUFFER_ADDRESS} with size %{SIZE} and data %{DATA}. Fields:

BUFFER_ADDRESS unsigned int

Virtual Address of start of buffer write.

DATA unsigned int

Data written to the buffer.

SIZE unsigned int

Size of sample written to buffer, in bytes.

UNALIGNED_LDST_RETIRED

Processor unaligned load/store. Fields:

ACCESS_SIZE unsigned int

Log2 of the access width in bytes used for alignment checking (1=>half-word, 2->word ...).

NUMBER_OF_BEATS unsigned int

Number of accesses (beats) in this burst (Total size in bytes is ACCESS_SIZE * NUMBER_OF_BEATS).

VADDR unsigned int

The virtual address of the access.

VFP_D_REGS

VFP/NEON D 64 bit register write. Fields:

ALIASING enum

Type of register aliasing used.

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a write to the corresponding bit in the register. For example, 0x1 would indicate a write to VALUE[0].

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

VFP_D_REGS_READ

VFP/NEON D 64 bit register read. Fields:

ID unsigned int

The register number.

VALUE unsigned int

The current value of the register.

VFP_Q_REGS

VFP/NEON Q 128 bit register write. Fields:

ALIASING enum

Type of register aliasing used.

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a write to the corresponding byte in the register. For example, 0x1 would indicate a write to VALUE[7:0].

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

VFP_Q_REGS_READ

VFP/NEON Q 128 bit register read. Fields:

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a read to the corresponding byte in the register. For example, 0x1 would indicate a read to VALUE[7:0].

VALUE unsigned int

The current value of the register.

VFP_SYS_REGS

Writes to the VFP/NEON units system registers. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ID enum

Which VFP system register is written.

MASK unsigned int

The mask indicating updated cumulative bits.

OLD_VALUE unsigned int

The register's old value overwritten.

VALUE unsigned int

The new value written to the VFP system register.

VFP_S_REGS

VFP/NEON S 32 bit register write. Fields:

ALIASING enum

Type of register aliasing used.

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a write to the corresponding bit in the register. For example, 0x1 would indicate a write to VALUE[0].

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

VFP_S_REGS_READ

VFP/NEON S 32 bit register read. Fields:

ID unsigned int

The register number.

VALUE unsigned int

The current value of the register.

VIRTUAL_TAG_ALLOCATED

Virtual tag information when allocated. Fields:

TAG_VADDR unsigned int

Virtual tag virtual address.

VADDR unsigned int

Data Virtual address.

VIRTUAL_TAG unsigned int

Virtual tag value.

WAYPOINT

Signals end of basic block execution. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ISSET enum

Origin instruction set.

IS_COND bool

Indicates if this is a conditional waypoint.

PC unsigned int

Origin address (or 0 if unavailable).

TAKEN bool

Indicates if this waypoint was taken.

TARGET unsigned int

Destination address.

TARGET_ISET enum

Destination instruction set.

WFE_END

WFE ended. Fields:

INST_COUNT unsigned int

Ticks count when leaving WFE.

WFE_EVENT_REGISTER

WFE event register status: set/clear, reason. Fields:

INST_COUNT unsigned int

Ticks count.

REASON enum

Reason for set/clear. Only REASON==0 (Cleared by WFE) clears the bit, all other reasons set it.

WFE_IGNORED

WFE ignored. Fields:

EVENT bool

This WFE was ignored because the event register was set.

INST_COUNT unsigned int

Ticks count when ignoring WFE.

TRAPPED bool

This WFE was trapped.

WFE_START

WFE entered. Fields:

INST_COUNT unsigned int

Ticks count when entering WFE.

WFI_END

WFI ended. Fields:

INST_COUNT unsigned int

Ticks count when leaving WFI.

WFI_IGNORED

WFI ignored. Fields:

DISABLED bool

This WFI was ignored because WFI is disabled.

INST_COUNT unsigned int

Ticks count when ignoring WFI.

REASON enum

specifies reason why WFI trace was ignored.

TRAPPED bool

This WFI was trapped.

WFI_START

WFI entered. Fields:

INST_COUNT unsigned int

Ticks count when entering WFI.

WFI_WAKEUP

WFI wakeup. Fields:

INST_COUNT unsigned int

Ticks count when WFI wakeup occurred.

REASON enum

Reason for wakeup.

2.59 ARM_Cortex-X3

This section describes the trace sources.

AA64_ASE_SVE_REGS

Modification of AArch64 Advanced SIMD or SVE Vector Registers (via SIMD or Scalar operations), SVE Predicate Registers and ZT0. Fields:

ID enum

The register identifier.

MASK unsigned int

Bitmask of the register to signal the modified bits in the VALUE field.

SM bool

Whether the PE is in Streaming Mode.

VALUE unsigned int

Value of AArch64 vector register.

ASYNC_MEMORY_FAULT

Context ID Register write. Fields:

FAULT unsigned int

Fault status present in ESR.ISS encoding for Serror in ARCH64 and present in DFSR encoding for external abort format in ARCH32.

PADDR unsigned int

Physical Address (or 0 if unavailable).

VADDR signed int

Virtual Address (or 0 if unavailable).

ATOMIC_END_ACCESS

Bus trace access for atomics. Fields:

ACCESS_FAIL bool

Memory access failed.

ADDR unsigned int

The virtual address of the access.

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

COMPARE_VALUE unsigned int

Compare value for CAS.

LOAD_VALUE unsigned int

Loaded values.

NSDESC unsigned int

The security state of the access.

OPERAND_VALUE unsigned int

Operation's operand.

OPERATION enum

Operation type.

PADDR unsigned int

The physical address of the access.

PRIV bool

Is this a privileged access?.

ATOMIC_START_ACCESS

Bus trace access for atomics. Fields:

ADDR unsigned int

The virtual address of the access.

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

COMPARE_VALUE unsigned int

Compare value for CAS.

NSDESC unsigned int

The security state of the access.

OPERAND_VALUE unsigned int

Operation's operand.

OPERATION enum

Operation type.

PADDR unsigned int

The physical address of the access.

PRIV bool

Is this a privileged access?.

ArchMsg.Error.exit_code

DISPLAY exit code: %{EXITCODE} component kind: %{KIND} reason: %{REASON}. Fields:

EXITCODE signed int

Exit code to be returned.

KIND string

Component kind that invoked the exit code trace.

REASON string

Reason why the exit code trace was invoked (optional).

ArchMsg.Warning.IgnoredWarmResetRequest

Event fires when software attempts to request warm reset through write to deprecated DBGPRCR.CWRR.

ArchMsg.Warning.RAS_ErrselrUnpredictableState

DISPLAY ERRSELR is in unpredictable state due to value greater than number_of_error_records written to it.

ArchMsg.Warning.RAS_UnpredictableStateWrite

DISPLAY %{REG_NAME} write ignored due to unpredictable ERRSELR state. Fields:

REG_NAME string

Name of register being written.

ArchMsg.Warning.ReportedRevisionRegisterRead

DISPLAY Reported revision and patch do not match that of the cpu. Fields:

ORIGINAL_VALUE unsigned int

The value the register would have based on the modelled revision.

REGISTER string

The name of the register that has ben read.

REPORTED_VALUE unsigned int

The value the register has been overridden to.

ArchMsg.Warning.Unpredictably Indexed PM Event Register

TAGS **UNPREDICTABLE** DISPLAY Event %{IsDirect:(Selector value|Register)} %{SEL} is >= %{N}. Fields:

IsDirect unsigned int

Direct Access rather than indirect.

N unsigned int

Number of Accessible Registers.

SEL unsigned int

Selector.

ArchMsg.Warning.cache_contents_unknown

A cache has been activated without being invalidated and may contain **UNKNOWN** content. DISPLAY %{SIDE:(D|I)}-Cache was enabled in %{REGIME:(SECURE|NON_SECURE|HYP)} regime but was not invalidated since power-on: cache lines could contain **UNKNOWN** content %{SIDE:(Note: D-side caches must invalidate SECURE lines even if the cache is only used by NON_SECURE code in order to ensure that unexpected dirty lines tagged SECURE are not naturally evicted)}. Fields:

REGIME unsigned int

Current translation regime.

SIDE unsigned int

0 d-side cache or unified cache, 1 i-side.

ArchMsg.Warning.dcimvac_matches_watchpoint

TAGS **UNPREDICTABLE** DISPLAY Watchpoints matching an AArch32 DCIMVA are implementation defined.

ArchMsg.Warning.decode_fieldmismatch

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with different fields %{FIELD1} and %{FIELD2}. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_frachitsnegative

TAGS **UNPREDICTABLE** DISPLAY VCVT Instruction is unpredictable with a negative number of fraction bits.

ArchMsg.Warning.decode_initblock

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable in an IT block.

ArchMsg.Warning.decode_initblocknotlast

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable in an IT block when not the last.

ArchMsg.Warning.decode_invalid_condition

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable when the condition is not AL.

ArchMsg.Warning.decode_invalidfieldcombination

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with the values in %{FIELD1} and %{FIELD2}. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_invalidvalue

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with the value in %{FIELD}. Fields:

FIELD enum

Field name.

ArchMsg.Warning.decode_mem_hint_unallocated

TAGS **UNPREDICTABLE** DISPLAY There is no memory hint allocated to this bit pattern.

ArchMsg.Warning.decode_registermatch

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with %{FIELD1} and %{FIELD2} as the same register. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_registermismatch

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with %{FIELD1} and %{FIELD2} as different registers. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_registeroutofrange

TAGS **UNPREDICTABLE** DISPLAY Floating-point instruction is unpredictable indexing beyond the end of the register bank.

ArchMsg.Warning.decode_sbzsbo

TAGS **UNPREDICTABLE** DISPLAY Reserved bits in the instruction are not canonical.

ArchMsg.Warning.decode_transactiontoobig

TAGS **UNPREDICTABLE** DISPLAY Attempt to perform a memory transaction of more than 128 bytes.

ArchMsg.Warning.decode_unpred_other

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable.

ArchMsg.Warning.decode_unpreduseofpc

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable when %{FIELD} is PC. Fields:

FIELD enum

Field name.

ArchMsg.Warning.decode_unpreduseofr13

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable when %{FIELD} is R13. Fields:

FIELD enum

Field name.

ArchMsg.Warning.decode_unpreduseofr13inreglist

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with R13 in register-list.

ArchMsg.Warning.decode_useofd16orhigher

TAGS **UNPREDICTABLE** DISPLAY Attempt to use register D16 or higher where it is not allowed.

ArchMsg.Warning.decode_writebackandbaseinlist

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with write back when the base register is in the transfer list.

ArchMsg.Warning.decode_zeroregistersinlist

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable when no registers are to be transferred.

ArchMsg.Warning.misaligned_ptrs

DISPLAY TRBPTR_EL1(%{TRBPTR_EL1}) or TRBTRG_EL1(%{TRBTRG_EL1}) are not aligned as per TRBIDR_EL1.Align(%{TRBIDR_EL1}). Fields:

TRBIDR_EL1 unsigned int

TRBIDR_EL1.Align value.

TRBPTR_EL1 unsigned int

Current write pointer.

TRBTRG_EL1 unsigned int

Trigger pointer.

ArchMsg.Warning.out_of_range_ptrs

DISPLAY TRBPTR_EL1(%{TRBPTR_EL1}) is out of range as per TRBBASER_EL1(%{TRBBASER_EL1}) or TRBLIMITR_EL1.LIMIT(%{TRBLIMITR_EL1}). Fields:

TRBBASER_EL1 unsigned int

Base pointer.

TRBLIMITR_EL1 unsigned int

Limit pointer.

TRBPTR_EL1 unsigned int

Current write pointer.

ArchMsg.Warning.recursive_branch

An instruction is performing a branch that targets the same instruction. If this was intended then a WFI might be more effective. DISPLAY recursive branch detected at PC=%{PC}. PRIMARY KEY PC. Fields:

PC unsigned int

Address of instruction causing the branch.

ArchMsg.Warning.recursive_exception

An instruction has generated an exception that targets the same instruction. DISPLAY recursive %{TYPE} exception detected at PC=%{PC} (CPSR=%{CPSR}). PRIMARY KEY PC. Fields:

CPSR unsigned int

Processor state.

PC unsigned int

Address of instruction causing the exception.

TYPE enum

Type of exception.

ArchMsg.Warning.reentrant_vector_catch

TAGS UNPREDICTABLE DISPLAY Vector catch debug event on Prefetch or Data abort vector %{ADDR}. Fields:

ADDR unsigned int

vector address.

ArchMsg.Warning.reserved_it_state

TAGS UNPREDICTABLE DISPLAY Execution with reserved IT state %{ITSTATE}. Fields:

ITSTATE unsigned int

reserved IT state.

ArchMsg.Warning.tlb_contents_unknown

Translation has been enabled without invalidating the TLB. DISPLAY Translation was enabled in the %{PAS:(Secure|Non-secure|Root|Realm)} EL%{EL:d} translation regime (vmid=

{VMID:x}) without invalidating {INVALIDITY:(|d-side |i-side |)} entries since power-on: {INVALIDITY:(|D||)} TLB could contain **UNKNOWN** content. Fields:

EL unsigned int

Exception level of translation regime.

INVALIDITY unsigned int

Which TLB may be invalid (if TLBs are separate).

NON_SECURE bool

Whether the translation regime is non-secure.

PAS unsigned int

Physical address space of translation regime.

VMID unsigned int

Current VMID, if applicable.

ArchMsg.Warning.unknown_DLR_DSPSR

Exiting debug state with unknown DLR or DSPSR.

ArchMsg.Warning.unknown_ELR_SPSR

Returning from debug exception with unknown ELR or SPSR.

ArchMsg.Warning.unpred_FAR_on_LRCPC3_non_atomic_mem_access

TAGS **UNPREDICTABLE** Unpredictable FAR/FSTATUS on LRCPC3 pair instructions resulting in two separate memory accesses
DISPLAY Unpredictable FAR on LRCPC3 non atomic memory access at {ADDR1} with FSTATUS {FSTATUS1} or {ADDR2} with FSTATUS {FSTATUS2}. Fields:

ADDR1 unsigned int

Possible FAR address 1.

ADDR2 unsigned int

Possible FAR address 2.

FSTATUS1 unsigned int

Possible fault status value 1.

FSTATUS2 unsigned int

Possible fault status value 2.

ArchMsg.Warning.unpred_async_tag_check_fail_on_sve_speculative_load_do_update_ffr

Unpredictable set to FALSE of the FFR element associated to SVE speculative load with an async tag check fail TAGS **UNPREDICTABLE**.

ArchMsg.Warning.unpredictable_a32_breakpoint

TAGS **UNPREDICTABLE** DISPLAY Unpredictable breakpoint on a A32 instruction at {ADDR1} Offset {OFFSET} BAS {BAS} for a {IS_ADDRESS_MISMATCH_BREAKPOINT:(Mismatch|Match)} Breakpoint number {N:d} Breakpoint address {ADDR2}. Fields:

ADDR1 unsigned int

Address.

ADDR2 unsigned int

Breakpoint address.

BAS unsigned int

BAS field.

IS_ADDRESS_MISMATCH_BREAKPOINT bool

Is a Mismatch breakpoint, not a Match breakpoint.

N unsigned int

Breakpoint number.

OFFSET unsigned int

Always has a value of zero.

ArchMsg.Warning.unpredictable_banked_register_access

TAGS **UNPREDICTABLE** DISPLAY Access to banked register (R=%{R:d} SYSm=%{SYSM:2x}) is unpredictable. Fields:

R unsigned int

R bit.

SYSM unsigned int

SYSm.

ArchMsg.Warning.unpredictable_ccfail_undef

TAGS **UNPREDICTABLE** DISPLAY An instruction %{OPCODE} at VA=%{VADDR} failed its condition codes check but would otherwise have caused a non-data-dependent trap or undefined exception. Behaviour is implementation-defined choice of exception or nop. Fields:

OPCODE unsigned int

opcode of the instruction.

VADDR unsigned int

Virtual address of the instruction being executed.

ArchMsg.Warning.unpredictable_ici_bits

TAGS **UNPREDICTABLE** DISPLAY ICI bits do not apply or are architecturally unpredictable for instruction.

ArchMsg.Warning.unpredictable_t32_breakpoint

TAGS **UNPREDICTABLE** DISPLAY Unpredictable breakpoint on a T32 instruction at %{ADDR1} Offset %{OFFSET} BAS %{BAS} for a %{IS_ADDRESS_MISMATCH_BREAKPOINT:(Mismatch|Match)} Breakpoint number %{N:d} Breakpoint address %{ADDR2}. Fields:

ADDR1 unsigned int

Address.

ADDR2 unsigned int

Breakpoint address.

BAS unsigned int

BAS field.

IS_ADDRESS_MISMATCH_BREAKPOINT bool

Is a Mismatch breakpoint, not a Match breakpoint.

N unsigned int

Breakpoint number.

OFFSET unsigned int

Indicates which halfword of a 32-bit Thumb instruction the breakpoint is set on. 0 for the first halfword or if the instruction is 16 bits wide, or 1 for the second halfword.

ArchMsg.Warning.unpredictable_t32_vector_catch

TAGS **UNPREDICTABLE** DISPLAY Unpredictable Vector catch exception on a T32 instruction at %{ADDR} Offset %{OFFSET}. Fields:

ADDR unsigned int

Address.

OFFSET unsigned int

Indicates which halfword of a 32-bit Thumb instruction the vector catch exception is set on. 0 for the first halfword or if the instruction is 16 bits wide, or 1 for the second halfword.

ArchMsg.Warning.unpredictable_watchpoint_far

TAGS **UNPREDICTABLE** DISPLAY The value for FAR/EDWAR for hitting this watchpoint may be anywhere in the range %{LOWER_BOUND} to %{UPPER_BOUND}. Fields:

LOWER_BOUND unsigned int

the lowest address accessed by the instruction that triggered the watchpoint.

UPPER_BOUND unsigned int

the highest watchpointed address accessed by that instruction.

ArchMsg.Warning.unpredictable_watchpoint_far_isscmp

DISPLAY The value for FAR/EDWAR for hitting this watchpoint may be anywhere in the range %{LOWER_BOUND} to %{UPPER_BOUND}. Fields:

LOWER_BOUND unsigned int

the lowest address accessed by the instruction that triggered the watchpoint.

UPPER_BOUND unsigned int

the highest watchpointed address accessed by that instruction.

ArchMsg.Warning.warning_AdvSIMDExpandImmUnexpectedZero

TAGS **UNPREDICTABLE** DISPLAY AdvSIMDExpandImm may treat this immediate value as **UNPREDICTABLE**.

ArchMsg.Warning.warning_ConditionalSMC

TAGS **UNPREDICTABLE** DISPLAY SMC instruction has **UNPREDICTABLE** effects when conditional and when combined with traps.

ArchMsg.Warning.warning_TLBID_out_of_range

TLBID out of range in TLBI according to the configuration of NIS, NVIS, NOS and NVOS.
Fields:

NIS unsigned int

NIS.

NOS unsigned int

NOS.

NVIS unsigned int

NVIS.

NVOS unsigned int

NVOS.

TLBID unsigned int

The TLBID value used in the TLBI instruction.

ArchMsg.Warning.warning_TTL_or_TG_mismatches_current_translation_granule

TTL or TG mismatches current translation granule in TLBI VA/RVA/IPA/RIPA. Fields:

CURRENT_TG unsigned int

Current TG as indicated by TCR_ELx or VTCR_EL2.

TLBI_TG unsigned int

TG (or TTL[3:2]) used in TLBI instruction.

ArchMsg.Warning.warning_access_crosses_page_boundary

TAGS UNPREDICTABLE DISPLAY Access crosses page boundary.

ArchMsg.Warning.warning_access_crosses_page_boundary_has_fault

TAGS UNPREDICTABLE DISPLAY Access that crosses page boundary has fault(s).

ArchMsg.Warning.warning_access_crosses_page_boundary_has_fault_on_both_pages

TAGS UNPREDICTABLE DISPLAY Access that crosses page boundary has a fault on each page.
Fields:

ADDR1 unsigned int

FAR address on first page.

ADDR2 unsigned int

FAR address on second page.

FSTATUS1 unsigned int

Fault status on first page.

FSTATUS2 unsigned int

Fault status on second page.

ArchMsg.Warning.warning_access_crosses_page_boundary_spanning_different_memory

TAGS UNPREDICTABLE DISPLAY Access at address %{ADDR} crosses page from MEM_TYPE %{MEMTYPE_PAGE1} to %{MEMTYPE_PAGE2}. Fields:

ADDR unsigned int

address of access crossing page.

MEMTYPE_PAGE1 signed int

First Page's memory type (0-Normal, 1-Non-Normal).

MEMTYPE_PAGE2 signed int

Second Page's memory type (0-Normal, 1-Non-Normal).

ArchMsg.Warning.warning_access_wraps_around_memory

TAGS **UNPREDICTABLE** DISPLAY Access wraps around memory in %{WIDTH:d}bit mode.
Fields:

WIDTH unsigned int

Address width.

ArchMsg.Warning.warning_bcr_linking_status

TAGS **UNPREDICTABLE** DISPLAY DBGBCR.{BT2,BT} combination gives unpredictable behaviour.

ArchMsg.Warning.warning_bcr_mask_reserved

TAGS **UNPREDICTABLE** DISPLAY DBGBCR_ELn.MASK is set to a reserved value.

ArchMsg.Warning.warning_bvr_masked_bit_not_zero

TAGS **UNPREDICTABLE** DISPLAY DBGBVRn_EL1.VA contains a bit masked by DBGBCRn_EL1.MASK that is not zero.

ArchMsg.Warning.warning_ccsidr_unimplemented_level

TAGS **UNPREDICTABLE** DISPLAY CCSIDR read while CSSELR %{CSSELR:x} points to an unimplemented cache level. Fields:

CSSELR unsigned int

current effective value of CSSELR.

ArchMsg.Warning.warning_change_to_ns_when_tge_set

TAGS **UNPREDICTABLE** DISPLAY Attempting to change NS to 1 when HCR.TGE = 1.

ArchMsg.Warning.warning_contiguous_bit_check_abort

TAGS **UNPREDICTABLE** DISPLAY An abort occurred whilst attempting to check TLB entry is contiguous at %{ENTRY_ADDR}. Fields:

ENTRY_ADDR unsigned int

address of conflicting TLB entry.

ArchMsg.Warning.warning_contiguous_bit_error

TAGS **UNPREDICTABLE** DISPLAY TLB entry at %{ENTRY_ADDR} was not contiguous with entry at %{CONTIG_ADDR} Contents are %{ENTRY_DATA} but expected %{CONTIG_DATA}.
Fields:

CONTIG_ADDR unsigned int

address of the TLB first read, with which this entry is expected to be contiguous.

CONTIG_DATA unsigned int

expected contents based on the entry at CONTIG_ADDR.

ENTRY_ADDR unsigned int

address of conflicting TLB entry.

ENTRY_DATA unsigned int

contents of conflicting TLB entry.

ArchMsg.Warning.warning_contiguous_bit_error_D128

TAGS **UNPREDICTABLE** DISPLAY TLB entry at %{ENTRY_ADDR} was not contiguous with entry at %{CONTIG_ADDR} Contents are %{ENTRY_DATA_HI} %{ENTRY_DATA_LO} but expected %{CONTIG_DATA}. Fields:

CONTIG_ADDR unsigned int

address of the TLB first read, with which this entry is expected to be contiguous.

CONTIG_DATA unsigned int

expected contents based on the entry at CONTIG_ADDR.

ENTRY_ADDR unsigned int

address of conflicting TLB entry.

ENTRY_DATA_HI unsigned int

Higher bit contents of conflicting TLB entry.

ENTRY_DATA_LO unsigned int

Lower bit contents of conflicting TLB entry.

ArchMsg.Warning.warning_cp10_cp11_mismatch

TAGS **UNPREDICTABLE** DISPLAY Unpredictable configuration of CP10 and CP11 controls at EL %{EL:d}. Fields:

EL unsigned int

EL owning the control.

ArchMsg.Warning.warning_cp10_cp11_reserved_value

TAGS **UNPREDICTABLE** DISPLAY Unpredictable configuration of CPACR CP10 (%{CP10}) or CP11 (%{CP11}). Fields:

CP10 unsigned int

CP10 access.

CP11 unsigned int

CP11 access.

ArchMsg.Warning.warning_csselr_level_out_of_range

TAGS **UNPREDICTABLE** DISPLAY CSSELR.Level out of range, written value %{LEVEL} with only %{IMPLEMENTED:d} levels implemented. Fields:

IMPLEMENTED unsigned int

number of cache levels implemented.

LEVEL unsigned int

written value of CSSELR.Level.

TYPE enum

The type of cache selected by CSSELR.InD and TnD.

ArchMsg.Warning.warning_debug_flow_control_bits_not_obeyed

TAGS UNPREDICTABLE DISPLAY Flow control bits for %{REGISTER} were not obeyed, causing %{ERROR}. Fields:

ERROR enum

type of error resulting.

REGISTER enum

register under consideration.

ArchMsg.Warning.warning_debug_register_access_during_reset

TAGS UNPREDICTABLE DISPLAY Unpredictable %{IS_WRITE:(read|write)} access to debug register offset %{OFFSET} during reset. Fields:

IS_WRITE unsigned int

Write Not Read.

OFFSET unsigned int

Register Offset.

ArchMsg.Warning.warning_decode_cps_inconsistent_fields

TAGS UNPREDICTABLE DISPLAY The mode setting fields in the CPS instruction are inconsistent.

ArchMsg.Warning.warning_decode_invalid_state

TAGS UNPREDICTABLE DISPLAY Instruction is unpredictable in current exception-level/security state.

ArchMsg.Warning.warning_default_cacheable_mmu_on

TAGS UNPREDICTABLE DISPLAY Default cacheable is enabled (HCR.DC = 1) while MMU is enabled.

ArchMsg.Warning.warning_default_cacheable_vmmu_off

TAGS UNPREDICTABLE DISPLAY Default cacheable is enabled (HCR.DC = 1) while HCR.VM = 0.

ArchMsg.Warning.warning_deprecated_wvr_bit_2_set

TAGS UNPREDICTABLE DISPLAY DBGWVR%{N:d}_EL1=%{WVR:h} has bit 2 set, deprecated in ARMv8. Fields:

N unsigned int

index.

WVR unsigned int

DBGWVR.

ArchMsg.Warning.warning_eret_while_software_step_active_pending

Missing ISB between setting MDSCR_EL1.SS and ERET TAGS **UNPREDICTABLE**.

ArchMsg.Warning.warning_exclusive_to_non_normal

TAGS **UNPREDICTABLE** DISPLAY Exclusive Access to Non-Normal Memory %{ADDR}. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_exclusive_to_non_writeback

TAGS **UNPREDICTABLE** DISPLAY Exclusive Access to Non-Writeback-Cacheable Memory %{ADDR}. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_execute_from_device_memory

TAGS **UNPREDICTABLE** DISPLAY An attempt was made to execute from Device memory.

ArchMsg.Warning.warning_illegal_cpsr_mode

TAGS **UNPREDICTABLE** DISPLAY Writing an illegal or unimplement mode (%{NEW_MODE:h}) to CPSR was unpredictable. Fields:

NEW_MODE unsigned int

New value of CPSR.M.

ArchMsg.Warning.warning_illegal_srs_mode

TAGS **UNPREDICTABLE** DISPLAY Illegal or **UNPREDICTABLE** mode (%{MODE:h}) used for SRS instruction. Fields:

MODE unsigned int

mode for Banked SP.

ArchMsg.Warning.warning_implementation_defined_read_debug_register_in_SCS

Read Access to Debug registers from the processor is IMPLEMENTATION DEFINED TAGS IMP_DEF. Fields:

PPB_OFFSET unsigned int

debug register offset.

ArchMsg.Warning.warning_implementation_defined_sequential_security_transitions_supported

The behavior of sequential instruction fetches that cross from non-secure to secure memory and contain an SG instruction is CONSTRAINED **UNPREDICTABLE** TAGS **UNPREDICTABLE**.

ArchMsg.Warning.warning_implementation_defined_stack_limit_check_supported

It is **IMPLEMENTATION DEFINED** whether stack pointer limit checking is performed for this instruction TAGS IMP_DEF. Fields:

ADDRESS unsigned int

Address of instruction.

OPCODE unsigned int

instruction opcode.

ArchMsg.Warning.warning_implementation_defined_taggednonsharedmemorytype

TAGS IMP_DEF DISPLAY Memory attribute has Tagged, non shared memory type; it is IMPLEMENTATION DEFINED whether the memory location is treated as Tagged or non-Tagged.

ArchMsg.Warning.warning_implementation_defined_write_debug_register_in_SCS

Write Access to Debug registers from the processor is IMPLEMENTATION DEFINED TAGS IMP_DEF. Fields:

DATA unsigned int

data attempted to be written.

PPB_OFFSET unsigned int

debug register offset.

ArchMsg.Warning.warning_invalid_tcr_granule

TAGS UNPREDICTABLE DISPLAY TCR.TG{%TG_ID:(0|1)} bits have been set to a granule size {%REQUEST:d}K not implemented - using {%SUBSTITUTE:d}K. Fields:

REQUEST signed int

page size requested (or 0 for reserved).

SUBSTITUTE signed int

best guess available page size.

TG_ID bool

bits TG0 or TG1.

ArchMsg.Warning.warning_load_multiple_user_registers_from_user_mode

TAGS UNPREDICTABLE DISPLAY An LDM(user registers) instruction executed from user mode.

ArchMsg.Warning.warning_mte_in_mem_attr_not_supported

TAGS UNPREDICTABLE DISPLAY Memory attribute {%ATTR} has Tagged memory type but Armv8.5-MTE is not implemented. Fields:

ATTR unsigned int

Memory attribute.

ArchMsg.Warning.warning_reserved_breakpoint_state_match

TAGS UNPREDICTABLE DISPLAY {%IS_BREAKPOINT:(Watchpoint|Breakpoint)} programmed with a reserved combination of HMC, SSC and {%IS_BREAKPOINT:(PAC|PMC)}. Fields:

IS_BREAKPOINT unsigned int

Is a breakpoint, not a watchpoint.

ArchMsg.Warning.warning_shareability

TAGS UNPREDICTABLE DISPLAY Unpredictable: combination of the 1st and 2nd stages of translation is Normal Inner Non-Cacheable, Outer Non-Cacheable.

ArchMsg.Warning.warning_software_step_set_while_enabled

TAGS UNPREDICTABLE DISPLAY MDSCR_EL1.SS set to 1 while software step debug exceptions are enabled.

ArchMsg.Warning.warning_thumb_instruction_wraps_around_memory

TAGS UNPREDICTABLE DISPLAY Thumb instruction wraps around memory.

ArchMsg.Warning.warning_ttbr_sbz_bits_are_not_zero

TAGS UNPREDICTABLE DISPLAY %{TTBR}[%{MSB:d}:0] should be zero, but are not. Fields:

MSB unsigned int

x-1.

TTBR enum

which TTBR register.

ArchMsg.Warning.warning_unaligned_address_dbgdtrrx_write

TAGS UNPREDICTABLE DISPLAY Unaligned address %{ADDR} in X0/R0 when DBGDTRRX written in memory access mode. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unaligned_address_dbgdtrtx_read

TAGS UNPREDICTABLE DISPLAY Unaligned address %{ADDR} in X0/R0 when DBGDTRTX read in memory access mode. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unaligned_to_device

TAGS UNPREDICTABLE DISPLAY Unaligned Access to Device Memory %{ADDR}. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unaligned_to_strongly_ordered

TAGS UNPREDICTABLE DISPLAY Unaligned Access to Strongly Ordered Memory %{ADDR}. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unknown_sau_rnr

TAGS UNPREDICTABLE DISPLAY SAU_RNR was set to an unsupported value.

ArchMsg.Warning.warning_unpredictable_AIRCR_PRIS_and_BFHFNMINs

TAGS UNPREDICTABLE DISPLAY The effect of setting both AIRCR.BFHFNMINs and AIRCR.PRIS to 1 is UNPREDICTABLE.

ArchMsg.Warning.warning_unpredictable_AIRCR_VECTCLRACTIVE_when_not_in_debug

TAGS **UNPREDICTABLE** DISPLAY The effect of writing a 1 to AIRCR.VECTCLRACTIVE if the processor is not halted in Debug state is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_AIRCR_VECTRESET_and_SYSRESETREQ

TAGS **UNPREDICTABLE** DISPLAY When the processor is halted in Debug state, if a write to the register writes a 1 to both VECTRESET and SYSRESETREQ, the behavior is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_AIRCR_VECTRESET_when_not_in_debug

TAGS **UNPREDICTABLE** DISPLAY The effect of writing a 1 to AIRCR.VECTRESET if the processor is not halted in Debug state is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_AIRCR_incorrect_VKEY

TAGS **UNPREDICTABLE** DISPLAY The value 0x05FA must be written to AIRCR.VKEY, otherwise the register write is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_EXC_RETURN_Reserved_Bit

EXC_RETURN[23:7] are reserved with the special condition that all bits should be written as one. Values other than all 1s are **UNPREDICTABLE**. TAGS **UNPREDICTABLE**. Fields:

EXC_RETURN unsigned int

Special PC value.

SBO_MASK unsigned int

Which bits Should Be One to not UNPRED.

ArchMsg.Warning.warning_unpredictable_PMEVTYPEPn_ELO_VS_reserved_value

TAGS **UNPREDICTABLE** DISPLAY Unpredictable reserved value of PMEVTYPEN<n>_ELO.VS.

ArchMsg.Warning.warning_unpredictable_change_to_DEMCR_MON_STEP_at_insufficient_priority

TAGS **UNPREDICTABLE** DISPLAY The effect of changing DEMCR.MON_STEP at an execution priority that is lower than the priority of the DebugMonitor exception is **CONSTRAINED UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_change_to_priority_of_active_exception

TAGS **UNPREDICTABLE** DISPLAY Changing the priority of an active exception is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_clear_lspact

TAGS **UNPREDICTABLE** DISPLAY **UNPREDICTABLE** state on context switch. Fields:

DRVEXC_TAKEN unsigned int

Derived exception taken, not pended.

REACHED_LAST_STORE unsigned int

Only the last faulted.

ArchMsg.Warning.warning_unpredictable_dbg_exit_unaligned_dlr

TAGS **UNPREDICTABLE** DISPLAY Debug exit to AArch32 with DLR[0] set to 1. PC[0] can be set to 0 or to DLR[0].

ArchMsg.Warning.warning_unpredictable_exception_catch

TAGS **UNPREDICTABLE** DISPLAY Generation of an exception catch event is unpredictable.

ArchMsg.Warning.warning_unpredictable_exception_return_inconsistent_state

TAGS **UNPREDICTABLE** DISPLAY state on exception return is unpredictable.

ArchMsg.Warning.warning_unpredictable_exception_return_instruction

TAGS **UNPREDICTABLE** DISPLAY exception return instruction is unpredictable.

ArchMsg.Warning.warning_unpredictable_in_debug_state

TAGS **UNPREDICTABLE** DISPLAY Instruction is **UNPREDICTABLE** when executed in debug state.

ArchMsg.Warning.warning_unpredictable_mair_encoding

TAGS **UNPREDICTABLE** DISPLAY Memory attribute %{ATTR} has invalid MAIR encoding; this may lead to unpredictable results. Fields:

ATTR unsigned int

Memory attribute.

ArchMsg.Warning.warning_unpredictable_pmu_counter_access

TAGS **UNPREDICTABLE** DISPLAY Access to PMU counters from non-secure EL0 or EL1 is **UNPREDICTABLE** with MDCR_EL2.HPMN set to 0.

ArchMsg.Warning.warning_unpredictable_prioritization_breakpoint_match_vector_catch

TAGS **UNPREDICTABLE** DISPLAY Two events with the same priority occurred at the same instruction: Address Matching Vector Catch debug event, and Breakpoint debug event. It is constrained unpredictable which is taken.

ArchMsg.Warning.warning_unpredictable_stack_selection

TAGS **UNPREDICTABLE** DISPLAY **UNPREDICTABLE** state on context switch.

ArchMsg.Warning.warning_unpredictable_tsize_out_of_range

TAGS **UNPREDICTABLE** DISPLAY %{TCR}.T%{TTBR:d}SZ (%{TSIZE:d} bits) is out of range. Must be between 25 and %{TSIZE_MAX:d}. Fields:

TCR enum

Which TCR.

TSIZE signed int

Value of TSize.

TSIZE_MAX signed int

The maximum allowed value.

TTBR unsigned int

Which TTBR region.

ArchMsg.Warning.warning_unpredictable_unaligned_pc_as_base_register

TAGS **UNPREDICTABLE** DISPLAY The PC value must be word-aligned, otherwise the behavior of the instruction is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_unaligned_pop_stack

TAGS **UNPREDICTABLE** DISPLAY **CONSTRAINED UNPREDICTABLE** unaligned stack frame %{FRAME_ADDR:x} in exception-return. Fields:

FRAME_ADDR unsigned int
unaligned stack frame address.

ArchMsg.Warning.warning_unpredictable_unaligned_to_non_writeback
TAGS **UNPREDICTABLE** DISPLAY Unaligned Access to Non-Writeback-Cacheable Memory.

ArchMsg.Warning.warning_unpredictable_vmsa_memattrib
TAGS **UNPREDICTABLE** DISPLAY Unpredictable vmsa memory attribute with texcb value programmed to %{TEXCB} when tex remap is %{TEX_REMAP}. Fields:

TEXCB unsigned int
texcb value.

TEX_REMAP bool
use tex remap.

ArchMsg.Warning.warning_unpredictable_vtcr_t0sz_sl0
TAGS **UNPREDICTABLE** DISPLAY The combination of VTCR.TOSZ=%{TOSZ:d} and VTCR.SLO=%{SLO:d} is unpredictable. Fields:

SLO signed int
VTCR.SLO, starting level for stage 2 translation table walks.

TOSZ signed int
VTCR.TOSZ, required input address range.

ArchMsg.Warning.warning_unpredictable_write_DHCSR
TAGS **UNPREDICTABLE** DISPLAY **UNPREDICTABLE** set of bit-changes in DHCSR. Fields:

NEW unsigned int
data attempted to be written.

OLD unsigned int
previous value.

ArchMsg.Warning.warning_unpredictable_write_SHCSR
TAGS **UNPREDICTABLE** DISPLAY **UNPREDICTABLE** setting a pending bit to 1 for an exception with priority >= execution priority.

ArchMsg.Warning.warning_unsupported_access_to_memory_mapped_register
TAGS **UNPREDICTABLE** DISPLAY Access to memory mapped register is unsupported.

ArchMsg.Warning.warning_user_jmcr_access
TAGS **UNPREDICTABLE** DISPLAY User %{WRITE:(write|read)} access to JMCR is unpredictable. Fields:

WRITE unsigned int
access is write.

ArchMsg.Warning.warning_wcr_mask_and_bas
TAGS **UNPREDICTABLE** DISPLAY DBGWCR_EL1.MASK is non-zero and BAS is not 0xff.

ArchMsg.Warning.warning_wcr_mask_reserved

TAGS **UNPREDICTABLE** DISPLAY DBGWCR_ELn.MASK is set to a reserved value.

ArchMsg.Warning.warning_wcr_non_configuous_bas

TAGS **UNPREDICTABLE** DISPLAY DBGWCR_EL1.BAS has non-contiguous set of ones.

ArchMsg.Warning.warning_wvr_masked_bit_not_zero

TAGS **UNPREDICTABLE** DISPLAY DBGWVRn_EL1.VA contains a bit masked by DBGWCRn_EL1.MASK that is not zero.

ArchMsg.Why.why_branch_target_exception

A branch target exception has occurred. DISPLAY branch target exception: target pc=%{PC}, target opcode=%{OPCODE}. PRIMARY KEY PC. Fields:

BTYPE unsigned int

pstate.BTYPE at time branch taken.

OPCODE unsigned int

opcode at branch target.

PC unsigned int

Virtual Address of branch target.

ArchMsg.Why.why_illegal_state

An illegal mode change has occurred. DISPLAY illegal mode change: %{MESSAGE}. Fields:

MESSAGE string

The message.

ArchMsg.warning_invalid_interval_counter_reload_value

Invalid interval counter reload value in PMSIRR_EL1.INTERVAL: software should set this to a value greater than or equal to the minimum indicated by PMSIDR_EL1.Interval.

ArchMsg.warning_invalid_spe_buffer_write_pointer

Invalid value programmed in PMBPTR_EL1.PTR or PMBLIMITR_EL1.LIMIT: Current write pointer must be at least one sample record below the write limit pointer and PMBPTR_EL1.PTR[63:56] must be equal to PMBLIMITR_EL1.LIMIT[63:56].

BRANCH_MISPREDICT

Simulating branch mispredict. Fields:

PC unsigned int

Origin address (or 0 if unavailable).

BRA_DIR

Direct branches, to immediate address. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISSET enum

The instruction set of the branch instruction.

IS_COND bool

Indicates if this is a conditional branch.

IS_HINTED bool

Whether the conditional instruction is hinted.

IS_ISB bool

Whether the instruction is ISB.

IS_LINK bool

Indicates if this is a branch with link.

PC unsigned int

The address of the branch instruction.

TARGET_ISSET enum

The instruction set after the branch.

TARGET_PC unsigned int

The address the instruction branches to.

BRA_INDIR

Indirect branches, perhaps to a register. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISSET enum

The instruction set of the branch instruction.

IS_COND bool

Indicates if this is a conditional branch.

IS_HINTED bool

Whether the conditional instruction is hinted.

IS_ISB bool

Whether the instruction is ISB.

IS_LINK bool

Indicates if this is a branch with link.

PC unsigned int

The address of the branch instruction.

TARGET_ISSET enum

The instruction set after the branch.

TARGET_PC unsigned int

The address the instruction branches to.

BRA_INDIR_CCFAIL

Indirect branches fail, perhaps to a register.

BRA_RET_CCFAIL

Branch return fail, perhaps to a register.

CACHE_MAINTENANCE_OP

Cache Maintenance Operation. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

DATA unsigned int

Specified MVA or Set/way.

FUNCTION enum

Clean / Invalidate.

SCOPE enum

Affected region.

SIDE enum

Inst / Data.

CCFAIL

Conditional instruction condition check fail. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CCFAIL_UNC

Unconditional instruction condition check fail. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CCPASS

Conditional instruction condition check pass. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CCPASS_UNC

Unconditional instruction condition check pass. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CODE_CACHE_MAINT

Code cache maintenance. Fields:

END_INCL unsigned int

Inclusive end address for VA or PA requests.

NS bool

Non-secure world for PA.

START unsigned int

Start address for VA or PA requests.

TYPE enum

Request type.

COMPILE_BLOCK_END

Last instruction of basic block translated. Fields:

VADDR unsigned int

Address of next instruction after this basic block.

COMPILE_BLOCK_START

First instruction of basic block translated. Fields:

VADDR unsigned int

Address of first instruction in basic block.

COMPILE_INST

ARM instruction compiled. Fields:

DISASS string

Disassembly of the instruction.

ISSET enum

The instruction set of this instruction.

ITSTATE unsigned int

The ITSTATE current for the instruction.

OPCODE unsigned int

The opcode of the instruction.

PC unsigned int

The address of the instruction.

SIZE unsigned int

The size of the instruction in bytes.

CONTEXTIDR

Context ID Register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

NS bool

Secure or nonsecure banked register is accessed.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The new value written.

CONTEXT_SYNC

Called for every context synchronization event. Fields:

ARCH_CSE bool

Architectural Context Synchronization Event.

PC unsigned int

The address of the synchronization instruction.

SYSREG_SYNC bool

System register synchronization event.

CORE_ENDIAN

Core BE8 Big-Endian state changed. Fields:

BE8 bool

Core BE8 Big-Endian state.

CORE_INFO

Static processor attributes. Only triggered by a call to DumpState(). Fields:

ARCH_PROFILE enum

The architecture profile of the core.

CLUSTER_ID unsigned int

The cluster ID of this processor.

CORE_NUM unsigned int

The number of this core in an MP processor.

DETAILED_CORE_FEATURES enum

Detailed core features as a bitfield. Enum values are bitmasks to extract the info from the field.

FPU_VERSION enum

The VFP version implemented by the core.

MEM_ARCH enum

The memory architecture of the core.

MIDR_EL1 unsigned int

MIDR_EL1 value.

NUM_CORES unsigned int

The number of cores in this MP processor.

QUANTUM_SIZE unsigned int

The default quantum size of the core (in ticks).

SECURITY_FEATURES bool

Does the core have security features?.

CORE_LOADS

Processor load accesses. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ACQREL enum

Is this an acquire/release.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DATA unsigned int

The data read or written.

ELEMENT_SIZE unsigned int

Width of each element.

ENDIANNESS enum

Byte ordering of this access.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

LOCK enum

Normal, exclusive or locked access.

MECID unsigned int

Memory Encryption Context ID, if applicable.

MECID2 unsigned int

Memory Encryption Context ID of the second page, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PADDR_TAG unsigned int

If tag checked, the physical address tag for the access.

PAGE_SIZE unsigned int

Page size as $\log_2(\text{size in bytes})$.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK, 2=Exclusive Failed, 3=Unsupported.

SIZE unsigned int

Width of the access in bytes. Only required if DATA is not traced.

TAG_CHECKED bool

Is this access tag checked.

TAG_CHECKED2 bool

Is the second page access tag checked.

TRANS bool

Is this a translated access.

VADDR unsigned int

The virtual address of the access.

CORE_REGS

Changes of the core registers R0 to R14. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number, 0 to 14.

MODE enum

Bank of the register accessed.

OLD_VALUE unsigned int

The old value overwritten.

PHYS_ID enum

The physical register accessed.

VALUE unsigned int

The new value written to the register.

CORE_REGS64

Changes of the core registers X0..X30, SP_ELn. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ID enum

The register number, 0-30 for X0-X30, >=32 for SP_ELn.

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

CORE_REGS64_READ

General purpose AArch64 register has been read X0..X30, SP_ELn. Fields:

ID enum

The register number, 0-30 for X0-X30, >=32 for SP_ELn.

VALUE unsigned int

The value read from the register.

CORE_STALLED_TRANSITION

Detect when a core is prevented from progressing PChannel states. Fields:

CURRENT_STATE enum

Current power mode of core.

REASON string

The reason a transition was stalled, and what must be done to unblock it.

TARGET_STATE enum

Desired power mode of core after transition.

CORE_STORES

Processor store accesses. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ACQREL enum

Is this an acquire/release.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DATA unsigned int

The data read or written.

ELEMENT_SIZE unsigned int

Width of each element.

ENDIANNESS enum

Byte ordering of this access.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

LOCK enum

Normal, exclusive or locked access.

MECID unsigned int

Memory Encryption Context ID, if applicable.

MECID2 unsigned int

Memory Encryption Context ID of the second page, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PADDR_TAG unsigned int

If tag checked, the physical address tag for the access.

PAGE_SIZE unsigned int

Page size as $\log_2(\text{size in bytes})$.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK, 2=Exclusive Failed, 3=Unsupported.

SIZE unsigned int

Width of the access in bytes. Only required if DATA is not traced.

TAG_CHECKED bool

Is this access tag checked.

TAG_CHECKED2 bool

Is the second page access tag checked.

TRANS bool

Is this a translated access.

VADDR unsigned int

The virtual address of the access.

CP14_READ

System Coprocessor register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP14 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc1 unsigned int

opcode 1.

opc2 unsigned int

opcode 2.

CP14_WRITE

System Coprocessor register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP14 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc1 unsigned int

opcode 1.

opc2 unsigned int

opcode 2.

CP15_READ

System Coprocessor register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP15 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc1 unsigned int

opcode 1.

opc2 unsigned int

opcode 2.

CP15_READ64

System Coprocessor register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP15 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc unsigned int

opcode 1.

CP15_WRITE

System Control Coprocessor register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP15 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc1 unsigned int

opcode 1.

opc2 unsigned int

opcode 2.

CP15_WRITE64

System Coprocessor register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP15 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc unsigned int

opcode 1.

CPSR

CPSR change. (In AArch64 PSTATE represented in SPSR format). Fields:

CORE_NUM unsigned int

Core number in a multi processor.

OLD_VALUE unsigned int

The old CPSR value.

UNKNOWN unsigned int

Bits within the register that have unknown value.

VALUE unsigned int

The new CPSR value.

CRYPTO_SPEC

Every crypto instruction speculatively executed.

CorePowerStateChange

Core power-up state has changed. DISPLAY Core powered %{PU:(up|down)}. Fields:

PU bool

Whether the core powered up.

DATA_CACHE_SET_ALLOC_TAGS

Set allocation tags from DC GVA instruction. Fields:

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

NSDESC unsigned int

The security state of the access.

PADDR unsigned int

The physical address of the access.

PAS enum

The physical address space of the page.

RESPONSE enum

0=Aborted, 1=OK.

SIZE unsigned int

The size of memory in bytes.

TAG unsigned int

The value of the tag being set.

TAGGING_TYPE enum

Type of Memory tagging (Physical or Virtual).

TAG_VADDR unsigned int

The virtual address of the Virtual Tag if Virtual tagging is used; otherwise 0.

VADDR unsigned int

Virtual address used for setting allocation tags.

DATA_CACHE_ZERO

Zero and invalidate cache line from DC ZVA instruction. Fields:

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

NSDESC unsigned int

The security state of the access.

PADDR unsigned int

The physical address of the access.

PADDR_TAG unsigned int

If tag checked, the physical address tag for the access.

PAS enum

The physical address space of the page.

RESPONSE enum

0=Aborted, 1=OK.

SIZE unsigned int

The size of zero'd memory in bytes.

TAG_CHECKED bool

Is this access tag checked.

VADDR unsigned int

Virtual address of the zero'd cache line.

DEBUG_EVENT

Hardware debug support event. Fields:

EVENT enum

Description of event.

VALUE unsigned int

data value.

DEBUG_TARGET_ISET

Hardware debug target ISA. Fields:

TARGET_ISET enum

The target instruction set of the debug state.

DMI_ALLOCATE

DMI allocated. Fields:

ACCESS_DIRECTION enum

READ / WRITE.

PADDR unsigned int

The physical address of the DMI.

PTR unsigned int

DMI range pointer.

SIDE enum

DSIDE / ISIDE.

SIZE unsigned int

The size of DMI.

VADDR unsigned int

The virtual address of the DMI..

DMI_HIT

DMI hit, access used DMI (only fire for small DMI regions i.e. less than a full page). Fields:

ACCESS_DIRECTION enum

READ / WRITE.

PADDR unsigned int

The physical address of the DMI.

SIDE enum

DSIDE / ISIDE.

SIZE unsigned int

The size of the DMI.

VADDR unsigned int

The virtual address of the DMI..

DMI_REVOKE

DMI revoked. Fields:

ACCESS_DIRECTION enum

READ / WRITE.

PADDR unsigned int

The physical address of the DMI.

SIDE enum

DSIDE / ISIDE.

SIZE unsigned int

The size of DMI.

VADDR unsigned int

The virtual address of the DMI..

END_COMPILE

Compilation end. Fields:

END_OF_PAGE_COUNT unsigned int

Number of basic blocks exited due to page boundary since START_COMPILE.

FETCHFAIL_COUNT unsigned int

Number of basic blocks exited due to ifetch failure START_COMPILE.

INST_COUNT unsigned int

Number of instructions compiled since START_COMPILE.

NONSEQ_COUNT unsigned int

Number of basic blocks exited due to non-sequential instructions since START_COMPILE.

PAGE_STRADDLE_COUNT unsigned int

Number of basic blocks exited due to unaligned instructions crossing page since START_COMPILE.

ESB

Error Synchronization Barrier hit. Fields:

ASYNC bool

ESB is async.

IMPLICIT bool

ESB is implicit.

EXCEPTION

Exception taken. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ESR unsigned int

The value assigned to ESR when exception is taken.

ESR_EC unsigned int

The Value assigned to ESR.EC when exception is taken.

IS_PHYSICAL bool

Physical or Virtual exception.

IS_TRAP bool

Exception coming from a trap.

LR unsigned int

The value assigned to the link register.

PC unsigned int

The location where the exception occurred.

PREFERRED_RETURN unsigned int

The preferred return address for the exception.

TAKEN_LOCALLY bool

Whether exception is taken locally(same exception level).

TARGET_ISET enum

The instruction set of the exception handler code.

TARGET_PC unsigned int

The address the exception branches to.

VECTOR enum

The exception vector.

VECTOR_OFFSET unsigned int

The offset of an exception vector from the base address.

EXCEPTION_END

Every exception completed.

EXCEPTION_INFO

Trace human-readable reason of the current exception for data/instruction aborts, floating point exceptions and unknown exceptions. Fields:

DESCRIPTION string

Reason for current exception.

ESR_EC unsigned int

The value of ESR.EC when exception is taken.

EXCEPTION_RAISE

Every exception raised.

EXCEPTION_RETURN

Branches on leaving exception. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISET enum

The instruction set of the branch instruction.

IS_COND bool

Indicates if this is a conditional branch.

IS_HINTED bool

Whether the conditional instruction is hinted.

IS_ISB bool

Whether the instruction is ISB.

IS_LINK bool

Indicates if this is a branch with link.

PC unsigned int

The address of the branch instruction.

TARGET_ISET enum

The instruction set after the branch.

TARGET_PC unsigned int

The address the instruction branches to.

EXCEPTION_RETURN_PREBRANCH

Exception return event that must occur before PC and processor state have updated.

EXCEPTION_START

Every exception started.

FIQ_TAKEN

FIQ taken exception.

FP_EXCEPTIONS_SIGNALLED

Trace which FP exceptions were signalled as a part of the FP operations performed by an instruction. Fields:

FP_EXCEPTIONS_SIGNALLED_MASK unsigned int

Indicates which FP exceptions were signalled by an instruction. This uses the same encoding as the cumulative floating-point exception bitfields in FPSR.

FP_STATE

Floating point state. Fields:

FPSR unsigned int

FPSR register value.

MASK unsigned int

The mask indicating updated cumulative bits.

FREQ_CHANGED

Core frequency changed. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The instruction count of this core.

NEW_FREQ signed int

The new frequency of this core (expressed in Hz).

OLD_FREQ signed int

The old frequency of this core (expressed in Hz).

HLT

HLT instruction occurred. Fields:

IMM16 unsigned int

The 16-bit immediate value encoded in HLT instruction.

INST

Every instruction executed. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DEBUG_STATE bool

The instruction is executed in debug state.

DISASS string

Disassembly of instruction.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISSET enum

The current instruction set.

ITSTATE unsigned int

The current ITSTATE.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

MODE enum

The mode the core is in.

NS unsigned int

The core's non-secure bit.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

OPCODE unsigned int

The opcode of the instruction.

PADDR unsigned int

The physical address of the instruction.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the instruction.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the page.

PC unsigned int

The address of the instruction.

SECURITY_STATE enum

The core's security state.

SIZE unsigned int

The size of the instruction in bytes.

INST_END

Every instruction completed.

INST_START

Every instruction started. Fields:

DEBUG_STATE bool

The instruction is executed in debug state.

ISET enum

The current instruction set.

MODE enum

The mode the core is in.

NS enum

The current Secure State.

PC unsigned int

The address of the conditional instruction.

SECURITY_STATE enum

The current Security State.

INST_STRADDLE

Instruction straddles boundary. Fields:

DEBUG_STATE bool

The instruction is executed in debug state.

ISET enum

The current instruction set.

MODE enum

The mode the core is in.

NS enum

The current Secure State.

PC unsigned int

The address of the conditional instruction.

SECURITY_STATE enum

The current Security State.

IRQ_TAKEN

IRQ taken exception.

ISB

ISB instruction occurred. Fields:

PC unsigned int

The address of the ISB instruction.

LOAD_MULTIPLE_REGS_LIST

List of destination registers (not source) of a multiple registers load instruction. Fields:

REG string

Register to add into the list.

LOCAL_MONITOR

Local monitor activity. Fields:

PADDR unsigned int

Local Monitor Address.

State enum

State of the monitor (Open/Exclusive).

MAPPING_CREATE

A memory mapping has been created. This is an implementation detail of the model and so may change without notice. Fields:

ASID unsigned int

The ASID used during translation, or 0 if no ASID was used.

IS_VALID bool

Whether the mapping is valid or not.

PADDR unsigned int

Base physical address of region, or 0 if not known.

SIDE enum

I-side or D-side.

SIZE unsigned int

Size of the page, or 0 if not known.

VADDR unsigned int

Base virtual address of region.

VMID unsigned int

The VMID used during translation, or 0 if no VMID was used.

MAPPING_DESTROY

A memory mapping has been destroyed. This is an implementation detail of the model and so may change without notice. Fields:

ASID unsigned int

The ASID used during translation.

CAUSE enum

The cause of invalidation.

PADDR unsigned int

Base physical address of page.

SIDE enum

I-side or D-side.

SIZE unsigned int

Size of the page.

VADDR unsigned int

Base virtual address of page.

VMID unsigned int

The VMID used during translation.

MEMMAP_DEBUG_READ

Memory mapped reads to the debug registers. Fields:

ADDR unsigned int

address.

CORE_NUM unsigned int

Core number in a multi processor.

EXT bool

Whether access is from an external device (such as the DAP).

NS enum

Secure state of the access.

REG_NAME string

Name of the debug register accessed.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

MEMMAP_DEBUG_WRITE

Memory mapped writes to the debug registers. Fields:

ADDR unsigned int

address.

CORE_NUM unsigned int

Core number in a multi processor.

EXT bool

Whether access is from an external device (such as the DAP).

NS enum

Secure state of the access.

REG_NAME string

Name of the debug register accessed.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The new value written.

MEMTAG_LOADS

Allocation tag memory read; when attributes and system configuration allow tag accesses.

Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

NUM_GRANULES unsigned int

Number of tags to be Loaded/stored.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK.

TAGGING_TYPE enum

Type of Memory tagging (Physical or Virtual).

TAGS unsigned int

The tags read or written. Tags for subsequent granules would be found at subsequent bytes in this field (One byte for each tag).

TAG_VADDR unsigned int

The virtual address of the Virtual Tag if Virtual tagging is used; otherwise 0.

VADDR unsigned int

The virtual address of the access.

MEMTAG_LOAD_INST

Allocation tag load instruction; regardless of tags being actually accessed. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

NUM_GRANULES unsigned int

Number of tags to be Loaded/stored.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK.

TAGGING_TYPE enum

Type of Memory tagging (Physical or Virtual).

TAGS unsigned int

The tags read or written. Tags for subsequent granules would be found at subsequent bytes in this field (One byte for each tag).

TAG_VADDR unsigned int

The virtual address of the Virtual Tag if Virtual tagging is used; otherwise 0.

VADDR unsigned int

The virtual address of the access.

MEMTAG_STORES

Allocation tag memory write; when attributes and system configuration allow tag accesses.

Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

NUM_GRANULES unsigned int

Number of tags to be Loaded/stored.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK.

TAGGING_TYPE enum

Type of Memory tagging (Physical or Virtual).

TAGS unsigned int

The tags read or written. Tags for subsequent granules would be found at subsequent bytes in this field (One byte for each tag).

TAG_VADDR unsigned int

The virtual address of the Virtual Tag if Virtual tagging is used; otherwise 0.

VADDR unsigned int

The virtual address of the access.

MENTAG_STORE_INST

Allocation tag store instruction; regardless of tags being actually accessed. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

NUM_GRANULES unsigned int

Number of tags to be Loaded/stored.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK.

TAGGING_TYPE enum

Type of Memory tagging (Physical or Virtual).

TAGS unsigned int

The tags read or written. Tags for subsequent granules would be found at subsequent bytes in this field (One byte for each tag).

TAG_VADDR unsigned int

The virtual address of the Virtual Tag if Virtual tagging is used; otherwise 0.

VADDR unsigned int

The virtual address of the access.

MMU_D128_TTB_READ

This event is triggered by reads caused by a D128 translation table walk. Fields:

ABORTED bool

Was the walk successful.

DATA_HIGH unsigned int

The data upper bits read.

DATA_LOW unsigned int

The data lower bits read.

IPA unsigned int

For stage 1, the IPA of the read.

LEVEL unsigned int

Translation table level.

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

OUTADDR unsigned int

The output physical address.

PADDR unsigned int

The physical address of the read.

PASmem enum

Physical Address Space of the accessed memory.

PASreq enum

Physical Address Space of the requested lookup.

REGIME_EL enum

Entry matches in this translation regime.

REQUESTER enum

The requester of this table walk.

SIDE enum

Inst / Data.

STAGE unsigned int

Translation stage.

MMU_D128_TTB_WRITE

This event is triggered by writes caused by a D128 translation table walk. Fields:

ABORTED bool

Was the walk successful.

DATA_HIGH unsigned int

The data upper bits written.

DATA_LOW unsigned int

The data lower bits written.

IPA unsigned int

For stage 1, the IPA of the write.

IS_HACDBS_ACCESS bool

If the access was due to HACDBS_CLEAN.

LEVEL unsigned int

Translation table level.

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

OUTADDR unsigned int

The output physical address.

PADDR unsigned int

The physical address of the write.

PASmem enum

Physical Address Space of the accessed memory.

PASreq enum

Physical Address Space of the requested lookup.

REGIME_EL enum

Entry matches in this translation regime.

SIDE enum

Inst / Data.

STAGE unsigned int

Translation stage.

MMU_TRANS

Address translation information. Fields:

ASID unsigned int

Address space identifier.

CORE_NUM unsigned int

Core number in a multi processor.

Hyp bool

Entry matches in Hyp state only.

INNERCACHE_RA bool

Is the inner cache allocate on read.

INNERCACHE_TRANSIENT bool

Is the inner write-through transient.

INNERCACHE_TYPE enum

Inner Caching scheme (NC/MB/WA).

INNERCACHE_WA bool

Is the inner cache allocate on write.

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MEMTYPE enum

Memory type.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC enum

Is secure side supposed to access secure or nonsecure memory.

OUTERCACHE_RA bool

Is the outer cache allocate on read.

OUTERCACHE_TRANSIENT bool

Is the outer write-through transient.

OUTERCACHE_TYPE enum

Outer Caching scheme (NC/MB/WA).

OUTERCACHE_WA bool

Is the outer cache allocate on write.

PADDR unsigned int

Physical address of the access.

PAGESIZE unsigned int

Page size as log2(size).

PAS enum

PAS supposed to be accessed is secure or nonsecure memory or Root/Realm if RME is implemented.

SH enum

Shareability.

SIDE enum

Inst / Data.

STAGE1_PERM unsigned int

Stage 1 permissions, [5:3] Privileged access (XWR) [2:0] User access (XWR).

STAGE2_PERM unsigned int

Stage 2 permission mask [5:3] Privileged access (XWR) [2:0] User access (XWR).

VADDR unsigned int

Virtual address of the access.

VMID unsigned int

Virtual machine identifier.

XS bool

XS Attribute.

nG enum

Flag indicating whether ASID will be matched.

MMU_TTB_READ

This event is triggered by reads caused by a translation table walk. Fields:

ABORTED bool

Was the walk successful.

DATA unsigned int

The data read.

DESC_KIND enum

The encoding scheme of the descriptors used for this walk.

IPA unsigned int

For stage 1, the IPA of the read.

LEVEL unsigned int

Translation table level.

LPAE bool

Is this for an LPAE translation (*DEPRECATED*, use DESC_KIND).

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSmem bool

Non secure memory access.

NSreq bool

Non secure request.

OUTADDR unsigned int

The output physical address.

PADDR unsigned int

The physical address of the read.

PASmem enum

Physical Address Space of the accessed memory.

PASreq enum

Physical Address Space of the requested lookup.

REGIME_EL enum

Entry matches in this translation regime.

REQUESTER enum

The requester of this table walk.

SIDE enum

Inst / Data.

STAGE unsigned int

Translation stage.

MMU_TTB_WRITE

This event is triggered by writes caused by a translation table walk. Fields:

ABORTED bool

Was the walk successful.

DATA unsigned int

The data written.

DESC_KIND enum

The encoding scheme of the descriptors used for this walk.

IPA unsigned int

For stage 1, the IPA of the write.

IS_HACDBS_ACCESS bool

If the access was due to HACDBS_CLEAN.

LEVEL unsigned int

Translation table level.

LPAE bool

Is this for an LPAE translation (*DEPRECATED*, use DESC_KIND).

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSmem bool

Non secure memory access.

NSreq bool

Non secure request.

OUTADDR unsigned int

The output physical address.

PADDR unsigned int

The physical address of the write.

PASmem enum

Physical Address Space of the accessed memory.

PASreq enum

Physical Address Space of the requested lookup.

REGIME_EL enum

Entry matches in this translation regime.

SIDE enum

Inst / Data.

STAGE unsigned int

Translation stage.

MODE_CHANGE

Mode change. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

MODE enum

The new mode.

NON_SECURE enum

The core's new non-secure bit.

OLD_MODE enum

The old mode.

SECURITY_STATE enum

The new security state.

NOT_TAG_CHECKED_INST

MTE Tag not checked for current instruction.

PERIODIC

Called for every quantum. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The instruction count of this CPU.

PC unsigned int

The address of the next instruction to be executed on this CPU.

PMU_COUNTER_OVERFLOW

PMU counter overflow. Fields:

COUNTER_GROUP enum

Counter group.

COUNTER_RANGE enum

Counter range.

EVENT_ID unsigned int

Event Identifier, field is valid only for Event counter group.

INDEX unsigned int

Counter index (as selected by PMSELR), field is valid only for Event/Cycle counter groups.

INTERRUPT bool

Is interrupt enabled on overflow for this counter.

POINTER_AUTH

Called when execution hits a pointer authentication check. Fields:

AUTH_STATE bool

Authentication result.

KEY_USED enum

Pointer authentication key.

POTINDEX_UPDATE

Value of POT index for current pc changed. Fields:

NEW_POTINDEX unsigned int

New POTIndex.

OLD_POTINDEX unsigned int

Old POTIndex.

PREFETCH_MEMORY64

Prefetch from PRFM or PRFUM instructions. Fields:

PRFOP enum

Prefetch hint.

VADDR unsigned int

Virtual address of the location that should be prefetched.

PRELOAD64_REQUEST

Preload requests from PRFM or PRFUM instructions. Fields:

PRFOP enum

Prefetch hint.

VADDR unsigned int

Virtual address of the location that should be prefetched.

PRELOAD_DATA

Data preload from PLD instruction. Fields:

VADDR unsigned int

Virtual address of the data that should be preloaded.

PRELOAD_INST

Instruction preload from PLI instruction. Fields:

VADDR unsigned int

Virtual address of the instruction that should be preloaded.

PRE_CORE_LOAD

Trace just before a core load. Fields:

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

PRE_CORE_STORE

Trace just before a core store. Fields:

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

PRE_TTB_READ

This event is triggered before reads caused by a translation table walk. Fields:

ADDR unsigned int

The physical address of the read.

PSTATE

PSTATE fields update. Fields:

ALLINT unsigned int

All interrupt mask bit.

BTYPE unsigned int

Branch target identification bit.

CORE_NUM unsigned int

Core number in a multi processor.

DAIF unsigned int

Exception mask bits.

DIT unsigned int

Data Independent Timing bit.

EL unsigned int

Current Exception level field.

EXLOCK unsigned int

Exception return state lock bit.

IL unsigned int

Illegal Execution state bit.

NZCV unsigned int

Condition flags.

PACM unsigned int

PACM bit.

PAN unsigned int

Privileged Access Never state bit.

PM unsigned int

PMU exception mask bit.

PPEND unsigned int

PMU exception pending bit.

SM unsigned int

StreamingMode status bit.

SP unsigned int

Stack pointer register selection bit.

SS unsigned int

Software Step bit.

SSBS unsigned int

Speculative Store Bypass Safe bit.

TCO unsigned int

Tag Check Override bit.

TOUCHED unsigned int

Mask of the fields that have been touched since this trace has last been emitted. Note that touched does not necessarily mean the value changed. The index of the field is the bit index in the mask plus the index of this field plus one.

UAO unsigned int

User Access Override bit.

UINJ unsigned int

Undefined Instruction exceptions bit.

ZA unsigned int

ZA storage and ZT0 register enable bit.

nRW unsigned int

Current Execution state bit.

PSTATE_PACTIVE_OPMODE_TRANSITION

Trace pactive and Opmode mode transitions on PChannel. Fields:

Copyright © 2023–2025 Arm Limited (or its affiliates). All rights reserved.

Non-Confidential

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Operating mode after the transition.

OLD_STATE enum

Operating mode before the transition.

PSTATE_PACTIVE_OPMODE_TRANSITION_DENIED

Trace the denied pactive and Opmode mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Operating mode after the transition.

OLD_STATE enum

Operating mode before the transition.

REASON string

Reason for the transition being denied.

PSTATE_PACTIVE_POWER_TRANSITION

Trace pactive and power mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Power mode after the transition.

OLD_STATE enum

Power mode before the transition.

PSTATE_PACTIVE_POWER_TRANSITION_DENIED

Trace the denied pactive and power mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Power mode after the transition.

OLD_STATE enum

Power mode before the transition.

REASON string

Reason for the transition being denied.

PSTATE_PACTIVE_THREAD_OPMODE_TRANSITION

Trace pactive and Opmode mode transitions for threads on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Operating mode after the transition.

OLD_STATE enum

Operating mode before the transition.

QUANTUM_START

Called at the start of every quantum.

RANGE_PRELOAD_REQUEST

Preload requests from RPRFM. Fields:

COUNT unsigned int

Number of blocks of data to access.

LENGTH unsigned int

Number of contiguous bytes to be accessed in each block.

OP unsigned int

Range prefetch operation .

REUSE unsigned int

Maximum number of bytes to be accessed before next RPRFM .

STRIDE signed int

Stride to advance after accessing LENGTH bytes.

VADDR unsigned int

Base virtual address of range prefetch request.

RUN_STATE

Run state transition. Fields:

INST_COUNT unsigned int

Ticks count at point of transition.

NEW enum

New run state.

OLD enum

Old run state.

SBP_SAMPLE_COLLISION

DISPLAY Statistical profiling samples collided .

SBP_SAMPLE_DISCARDED

DISPLAY Operation discarded by statistical profiling unit when configured in discard mode.
Fields:

BUFFER_POINTER unsigned int

Value of buffer pointer.

DATA unsigned int

Data discarded.

SIZE unsigned int

Size of sample, in bytes.

SBP_SAMPLE_END

DISPLAY Statistical profiling of an operation ended.

SBP_SAMPLE_POP

DISPLAY Operation which might be sampled by statistical profiling.

SBP_SAMPLE_RECORDED

DISPLAY Operation sampled by statistical profiling has been written to the buffer at %{BUFFER_ADDRESS} with size %{SIZE} and data %{DATA}. Fields:

BUFFER_ADDRESS unsigned int

Virtual Address of start of buffer write.

DATA unsigned int

Data written to the buffer.

SIZE unsigned int

Size of sample written to buffer, in bytes.

SBP_SAMPLE_START

DISPLAY Statistical profiling of an operation started.

SBP_SAMPLE_TAKEN

DISPLAY Operation sampled by statistical profiling after filtering.

SEMIHOSTING_CALL

Call of a semihost function occurred. Fields:

PC unsigned int

The program counter after the semihosting call.

SEMIHOSTING_PRECALL

About to call semihost. Fields:

EL2 bool

Use EL2 translation regime.

NS bool

Is Non-Secure.

REG_WIDTH unsigned int

The current register width in bytes.

SIGNAL

External signal state change. Fields:

SIGNAL enum

Signal that changed.

STATE bool

Signal asserted state.

SOFTWARE_STEP

Return the Debug Software Step state. Fields:

STATE enum

Software Step state.

SPSR

SPSR change. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

MODE enum

Which of the banked SPSR registers is written.

OLD_VALUE unsigned int

The old SPSR value.

VALUE unsigned int

The new SPSR value.

START_COMPILE

Compilation started. Fields:

VADDR unsigned int

Instruction where compilation begins.

STG_MULTIPLE_MICRO_OPS_ADDRESS_RANGE

Address range for MTE stores with multiple uops. Fields:

ADDR_END unsigned int

Final address.

ADDR_START unsigned int

Starting address.

STORE_MULTIPLE_REGS_ADDRESS_RANGE

Address range for stores with multiple registers. Fields:

ADDR_END unsigned int

Final address.

ADDR_START unsigned int

Starting address.

SUBOPTIMAL_LDST_RETIRED

suboptimal load/store. Fields:

ACCESS_SIZE unsigned int

Log2 of the access width in bytes used for alignment checking (1: half-word, 2: word ...).

OPTIMAL_ALIGN_SELECTOR unsigned int

Selector to decide the optimal alignment size (1: access width, 2: 4-bytes, 3: 8-bytes, ..., 12: 4-kbytes).

VADDR unsigned int

The virtual address of the access.

SVE_INST_SPEC

SVE operations speculatively executed.

SVE_LD_RETIRED

SVE load instructions architecturally executed.

SVE_LOADS

SVE load accesses. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ACQREL enum

Is this an acquire/release.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DATA unsigned int

The data read or written.

ELEMENT_SIZE unsigned int

Width of each element.

ENDIANNESS enum

Byte ordering of this access.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

LOCK enum

Normal, exclusive or locked access.

MECID unsigned int

Memory Encryption Context ID, if applicable.

MECID2 unsigned int

Memory Encryption Context ID of the second page, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PADDR_TAG unsigned int

If tag checked, the physical address tag for the access.

PAGE_SIZE unsigned int

Page size as $\log_2(\text{size in bytes})$.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK, 2=Exclusive Failed, 3=Unsupported.

SIZE unsigned int

Width of the access in bytes. Only required if DATA is not traced.

TAG_CHECKED bool

Is this access tag checked.

TAG_CHECKED2 bool

Is the second page access tag checked.

TRANS bool

Is this a translated access.

VADDR unsigned int

The virtual address of the access.

SVE_MEM_PREFETCH

Scalable Vector Extension memory prefetch. Fields:

CACHE_LEVEL enum

The preferred cache level.

HINT enum

What the memory is expected to be used for.

SIZE unsigned int

The prefetch size in bytes.

TEMPORAL bool

Whether the data is temporal.

VADDR unsigned int

The virtual address of the prefetch.

SVE_REG_READ

Scalable Vector Extension register read. Fields:

ID enum

The register identifier.

SM bool

Whether the PE is in Streaming Mode.

VALUE unsigned int

The read data.

SVE_REG_UPDATE

Scalable Vector Extension register update. Fields:

ID enum

The register identifier.

SM bool

Whether the PE is in Streaming Mode.

VALUE unsigned int

The written data.

SVE_REG_WRITE

Scalable Vector Extension register write. Fields:

ID enum

The register identifier.

SM bool

Whether the PE is in Streaming Mode.

VALUE unsigned int

The written data.

SVE_STORES

SVE store accesses. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ACQREL enum

Is this an acquire/release.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DATA unsigned int

The data read or written.

ELEMENT_SIZE unsigned int

Width of each element.

ENDIANNESS enum

Byte ordering of this access.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

LOCK enum

Normal, exclusive or locked access.

MECID unsigned int

Memory Encryption Context ID, if applicable.

MECID2 unsigned int

Memory Encryption Context ID of the second page, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PADDR_TAG unsigned int

If tag checked, the physical address tag for the access.

PAGE_SIZE unsigned int

Page size as $\log_2(\text{size in bytes})$.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK, 2=Exclusive Failed, 3=Unsupported.

SIZE unsigned int

Width of the access in bytes. Only required if DATA is not traced.

TAG_CHECKED bool

Is this access tag checked.

TAG_CHECKED2 bool

Is the second page access tag checked.

TRANS bool

Is this a translated access.

VADDR unsigned int

The virtual address of the access.

SVE_ST_RETIRED

SVE store instructions architecturally executed.

SVE_UNALIGNED_LDST_RETIRED

SVE load or store instructions architecturally executed that would cause alignment fault if A bit is set.

SVE_Z_PRED_MEM_ACCESS_INFO

Information about SVE/SME predicated accesses to Z registers (non Gather/Scatter). Fields:

MEM_ELEM_SIZE unsigned int

Width of elements accessed in memory (in bytes).

RnW bool

Whether this is a Read not a Write.

VA unsigned int

Virtual base address computed for accesses.

SYNC

Called for every synchronization, that is every quantum break. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The instruction count of this CPU.

LOCAL_QUANTUM unsigned int

The local quantum of this CPU.

LOCAL_TIME unsigned int

The local time of this CPU.

SYNC_REASON enum

Reason for sync.

SYSCALL

System call instruction executed. Fields:

IMM unsigned int

Immediate value of the system call instruction.

TYPE enum

System call type.

VADDR unsigned int

Instruction that caused the system call.

SYSREG_READ128

System Coprocessor 128 bit register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

REGNUM enum

Internal register number.

REGNUM_PHYS enum

Internal register number physically accessed (e.g. in presense of VHE redirection).

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_READ64

System Coprocessor register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REGNUM enum

Internal register number.

REGNUM_OPERAND enum

Register number corresponding to the instruction operand.

REGNUM_PHYS enum

Internal register number physically accessed (e.g. in presense of VHE redirection).

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_UPDATE128

Triggers when the system updates a register. Fields:

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG enum

Register number.

UNKNOWN unsigned int

Bits of the register which became unknown.

VALUE unsigned int

Value written to the register.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_UPDATE32

Triggers when the system updates a register. Fields:

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG enum

Register number.

UNKNOWN unsigned int

Bits of the register which became unknown.

VALUE unsigned int

Value written to the register.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_UPDATE64

Triggers when the system updates a register. Fields:

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG enum

Register number.

UNKNOWN unsigned int

Bits of the register which became unknown.

VALUE unsigned int

Value written to the register.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_WRITE128

System Coprocessor 128 bit register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

REGNUM enum

Internal register number.

REGNUM_PHYS enum

Internal register number physically accessed (e.g. in presense of VHE redirection).

UNDEF bool

The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_WRITE64

System Coprocessor register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REGNUM enum

Internal register number.

REGNUM_OPERAND enum

Register number corresponding to the instruction operand.

REGNUM_PHYS enum

Internal register number physically accessed (e.g. in presense of VHE redirection).

UNDEF bool

The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

TRBU_BUFFER_WRAP

DISPLAY Trace buffer current write pointer wrapped.

TRBU_TRIGGER_EVENT

DISPLAY Trace buffer Trigger Event.

TRBU_WRITE

DISPLAY PETU trace data has been written to the TRBU memory buffer at `%{BUFFER_ADDRESS}` with size `%{SIZE}` and data `%{DATA}`. Fields:

BUFFER_ADDRESS unsigned int

Virtual Address of start of buffer write.

DATA unsigned int

Data written to the buffer.

SIZE unsigned int

Size of sample written to buffer, in bytes.

UNALIGNED_LDST_RETIRED

Processor unaligned load/store. Fields:

ACCESS_SIZE unsigned int

Log2 of the access width in bytes used for alignment checking (1=>half-word, 2->word ...).

NUMBER_OF_BEATS unsigned int

Number of accesses (beats) in this burst (Total size in bytes is `ACCESS_SIZE * NUMBER_OF_BEATS`).

VADDR unsigned int

The virtual address of the access.

VFP_D_REGS

VFP/NEON D 64 bit register write. Fields:

ALIASING enum

Type of register aliasing used.

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a write to the corresponding bit in the register. For example, 0x1 would indicate a write to VALUE[0].

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

VFP_D_REGS_READ

VFP/NEON D 64 bit register read. Fields:

ID unsigned int

The register number.

VALUE unsigned int

The current value of the register.

VFP_Q_REGS

VFP/NEON Q 128 bit register write. Fields:

ALIASING enum

Type of register aliasing used.

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a write to the corresponding byte in the register. For example, 0x1 would indicate a write to VALUE[7:0].

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

VFP_Q_REGS_READ

VFP/NEON Q 128 bit register read. Fields:

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a read to the corresponding byte in the register. For example, 0x1 would indicate a read to VALUE[7:0].

VALUE unsigned int

The current value of the register.

VFP_SYS_REGS

Writes to the VFP/NEON units system registers. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ID enum

Which VFP system register is written.

MASK unsigned int

The mask indicating updated cumulative bits.

OLD_VALUE unsigned int

The register's old value overwritten.

VALUE unsigned int

The new value written to the VFP system register.

VFP_S_REGS

VFP/NEON S 32 bit register write. Fields:

ALIASING enum

Type of register aliasing used.

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a write to the corresponding bit in the register. For example, 0x1 would indicate a write to VALUE[0].

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

VFP_S_REGS_READ

VFP/NEON S 32 bit register read. Fields:

ID unsigned int

The register number.

VALUE unsigned int

The current value of the register.

VIRTUAL_TAG_ALLOCATED

Virtual tag information when allocated. Fields:

TAG_VADDR unsigned int

Virtual tag virtual address.

VADDR unsigned int

Data Virtual address.

VIRTUAL_TAG unsigned int

Virtual tag value.

WAYPOINT

Signals end of basic block execution. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ISET enum

Origin instruction set.

IS_COND bool

Indicates if this is a conditional waypoint.

PC unsigned int

Origin address (or 0 if unavailable).

TAKEN bool

Indicates if this waypoint was taken.

TARGET unsigned int

Destination address.

TARGET_ISET enum

Destination instruction set.

WFE_END

WFE ended. Fields:

INST_COUNT unsigned int

Ticks count when leaving WFE.

WFE_EVENT_REGISTER

WFE event register status: set/clear, reason. Fields:

INST_COUNT unsigned int

Ticks count.

REASON enum

Reason for set/clear. Only REASON==0 (Cleared by WFE) clears the bit, all other reasons set it.

WFE_IGNORED

WFE ignored. Fields:

EVENT bool

This WFE was ignored because the event register was set.

INST_COUNT unsigned int

Ticks count when ignoring WFE.

TRAPPED bool

This WFE was trapped.

WFE_START

WFE entered. Fields:

INST_COUNT unsigned int

Ticks count when entering WFE.

WFI_END

WFI ended. Fields:

INST_COUNT unsigned int

Ticks count when leaving WFI.

WFI_IGNORED

WFI ignored. Fields:

DISABLED bool

This WFI was ignored because WFI is disabled.

INST_COUNT unsigned int

Ticks count when ignoring WFI.

REASON enum

specifies reason why WFI trace was ignored.

TRAPPED bool

This WFI was trapped.

WFI_START

WFI entered. Fields:

INST_COUNT unsigned int

Ticks count when entering WFI.

WFI_WAKEUP

WFI wakeup. Fields:

INST_COUNT unsigned int

Ticks count when WFI wakeup occurred.

REASON enum

Reason for wakeup.

2.60 ARM_Cortex-X4

This section describes the trace sources.

AA64_ASE_SVE_REGS

Modification of AArch64 Advanced SIMD or SVE Vector Registers (via SIMD or Scalar operations), SVE Predicate Registers and ZT0. Fields:

ID **enum**

The register identifier.

MASK **unsigned int**

Bitmask of the register to signal the modified bits in the VALUE field.

SM **bool**

Whether the PE is in Streaming Mode.

VALUE **unsigned int**

Value of AArch64 vector register.

ASYNC_MEMORY_FAULT

Context ID Register write. Fields:

FAULT **unsigned int**

Fault status present in ESR.ISS encoding for Serror in ARCH64 and present in DFSR encoding for external abort format in ARCH32.

PADDR **unsigned int**

Physical Address (or 0 if unavailable).

VADDR **signed int**

Virtual Address (or 0 if unavailable).

ATOMIC_END_ACCESS

Bus trace access for atomics. Fields:

ACCESS_FAIL **bool**

Memory access failed.

ADDR **unsigned int**

The virtual address of the access.

ATTR **unsigned int**

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

COMPARE_VALUE **unsigned int**

Compare value for CAS.

LOAD_VALUE unsigned int

Loaded values.

NSDESC unsigned int

The security state of the access.

OPERAND_VALUE unsigned int

Operation's operand.

OPERATION enum

Operation type.

PADDR unsigned int

The physical address of the access.

PRIV bool

Is this a privileged access?.

ATOMIC_START_ACCESS

Bus trace access for atomics. Fields:

ADDR unsigned int

The virtual address of the access.

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

COMPARE_VALUE unsigned int

Compare value for CAS.

NSDESC unsigned int

The security state of the access.

OPERAND_VALUE unsigned int

Operation's operand.

OPERATION enum

Operation type.

PADDR unsigned int

The physical address of the access.

PRIV bool

Is this a privileged access?.

ArchMsg.Error.exit_code

DISPLAY exit code: %{EXITCODE} component kind: %{KIND} reason: %{REASON}. Fields:

EXITCODE signed int

Exit code to be returned.

KIND string

Component kind that invoked the exit code trace.

REASON string

Reason why the exit code trace was invoked (optional).

ArchMsg.Warning.IgnoredWarmResetRequest

Event fires when software attempts to request warm reset through write to deprecated DBGPRCR.CWRR.

ArchMsg.Warning.RAS_ErrselrUnpredictableState

DISPLAY ERRSELR is in unpredictable state due to value greater than number_of_error_records written to it.

ArchMsg.Warning.RAS_UnpredictableStateWrite

DISPLAY %{REG_NAME} write ignored due to unpredictable ERRSELR state. Fields:

REG_NAME string

Name of register being written.

ArchMsg.Warning.ReportedRevisionRegisterRead

DISPLAY Reported revision and patch do not match that of the cpu. Fields:

ORIGINAL_VALUE unsigned int

The value the register would have based on the modelled revision.

REGISTER string

The name of the register that has ben read.

REPORTED_VALUE unsigned int

The value the register has been overridden to.

ArchMsg.Warning.Unpredictably Indexed PM Event Register

TAGS UNPREDICTABLE DISPLAY Event %{IsDirect:(Selector value|Register)} %{SEL} is >= %{N}. Fields:

IsDirect unsigned int

Direct Access rather than indirect.

N unsigned int

Number of Accessible Registers.

SEL unsigned int

Selector.

ArchMsg.Warning.cache_contents_unknown

A cache has been activated without being invalidated and may contain **UNKNOWN** content. DISPLAY %{SIDE:(D|I)}-Cache was enabled in %{REGIME:(SECURE|NON_SECURE|HYP)} regime but was not invalidated since power-on: cache lines could contain **UNKNOWN** content %{SIDE:(Note: D-side caches must invalidate SECURE lines even if the cache is only used by NON_SECURE code in order to ensure that unexpected dirty lines tagged SECURE are not naturally evicted)}. Fields:

REGIME unsigned int

Current translation regime.

SIDE unsigned int

0 d-side cache or unified cache, 1 i-side.

ArchMsg.Warning.dcimvac_matches_watchpoint

TAGS **UNPREDICTABLE** DISPLAY Watchpoints matching an AArch32 DCIMVA are implementation defined.

ArchMsg.Warning.decode_fieldmismatch

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with different fields %{FIELD1} and %{FIELD2}. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_fracbitsnegative

TAGS **UNPREDICTABLE** DISPLAY VCVT Instruction is unpredictable with a negative number of fraction bits.

ArchMsg.Warning.decode_initblock

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable in an IT block.

ArchMsg.Warning.decode_initblocknotlast

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable in an IT block when not the last.

ArchMsg.Warning.decode_invalid_condition

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable when the condition is not AL.

ArchMsg.Warning.decode_invalidfieldcombination

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with the values in %{FIELD1} and %{FIELD2}. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_invalidvalue

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with the value in %{FIELD}. Fields:

FIELD enum

Field name.

ArchMsg.Warning.decode_mem_hint_unallocated

TAGS **UNPREDICTABLE** DISPLAY There is no memory hint allocated to this bit pattern.

ArchMsg.Warning.decode_registermatch

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with %{FIELD1} and %{FIELD2} as the same register. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_registermismatch

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with %{FIELD1} and %{FIELD2} as different registers. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_registeroutofrange

TAGS **UNPREDICTABLE** DISPLAY Floating-point instruction is unpredictable indexing beyond the end of the register bank.

ArchMsg.Warning.decode_sbzsbo

TAGS **UNPREDICTABLE** DISPLAY Reserved bits in the instruction are not canonical.

ArchMsg.Warning.decode_transactiontoobig

TAGS **UNPREDICTABLE** DISPLAY Attempt to perform a memory transaction of more than 128 bytes.

ArchMsg.Warning.decode_unpred_other

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable.

ArchMsg.Warning.decode_unpreduseofpc

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable when %{FIELD} is PC. Fields:

FIELD enum

Field name.

ArchMsg.Warning.decode_unpreduseofr13

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable when %{FIELD} is R13. Fields:

FIELD enum

Field name.

ArchMsg.Warning.decode_unpreduseofr13inreglist

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with R13 in register-list.

ArchMsg.Warning.decode_useofd16orhigher

TAGS **UNPREDICTABLE** DISPLAY Attempt to use register D16 or higher where it is not allowed.

ArchMsg.Warning.decode_writebackandbaseinlist

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with write back when the base register is in the transfer list.

ArchMsg.Warning.decode_zeroregistersinlist

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable when no registers are to be transferred.

ArchMsg.Warning.misaligned_ptrs

DISPLAY TRBPTR_EL1(%{TRBPTR_EL1}) or TRBTRG_EL1(%{TRBTRG_EL1}) are not aligned as per TRBIDR_EL1.Align(%{TRBIDR_EL1}). Fields:

TRBIDR_EL1 unsigned int

TRBIDR_EL1.Align value.

TRBPTR_EL1 unsigned int

Current write pointer.

TRBTRG_EL1 unsigned int

Trigger pointer.

ArchMsg.Warning.out_of_range_ptrs

DISPLAY TRBPTR_EL1(%{TRBPTR_EL1}) is out of range as per TRBBASER_EL1(%{TRBBASER_EL1}) or TRBLIMITR_EL1.LIMIT(%{TRBLIMITR_EL1}). Fields:

TRBBASER_EL1 unsigned int

Base pointer.

TRBLIMITR_EL1 unsigned int

Limit pointer.

TRBPTR_EL1 unsigned int

Current write pointer.

ArchMsg.Warning.recursive_branch

An instruction is performing a branch that targets the same instruction. If this was intended then a WFI might be more effective DISPLAY recursive branch detected at PC=%{PC}.

PRIMARY KEY PC. Fields:

PC unsigned int

Address of instruction causing the branch.

ArchMsg.Warning.recursive_exception

An instruction has generated an exception that targets the same instruction. DISPLAY recursive %{TYPE} exception detected at PC=%{PC} (CPSR=%{CPSR}). PRIMARY KEY PC. Fields:

CPSR unsigned int

Processor state.

PC unsigned int

Address of instruction causing the exception.

TYPE enum

Type of exception.

ArchMsg.Warning.reentrant_vector_catch

TAGS **UNPREDICTABLE** DISPLAY Vector catch debug event on Prefetch or Data abort vector %{ADDR}. Fields:

ADDR unsigned int

vector address.

ArchMsg.Warning.reserved_it_state

TAGS **UNPREDICTABLE** DISPLAY Execution with reserved IT state %{ITSTATE}. Fields:

ITSTATE unsigned int

reserved IT state.

ArchMsg.Warning.tlb_contents_unknown

Translation has been enabled without invalidating the TLB. DISPLAY Translation was enabled in the %{PAS:(Secure|Non-secure|Root|Realm)} EL%{EL:d} translation regime (vmid=%{VMID:x}) without invalidating %{INVALIDITY:(|d-side |i-side |)}entries since power-on: %{INVALIDITY:(|D|I|)}TLB could contain **UNKNOWN** content. Fields:

EL unsigned int

Exception level of translation regime.

INVALIDITY unsigned int

Which TLB may be invalid (if TLBs are separate).

NON_SECURE bool

Whether the translation regime is non-secure.

PAS unsigned int

Physical address space of translation regime.

VMID unsigned int

Current VMID, if applicable.

ArchMsg.Warning.unknown_DLR_DSPSR

Exiting debug state with unknown DLR or DSPSR.

ArchMsg.Warning.unknown_ELR_SPSR

Returning from debug exception with unknown ELR or SPSR.

ArchMsg.Warning.unpred_FAR_on_LRCPC3_non_atomic_mem_access

TAGS **UNPREDICTABLE** Unpredictable FAR/FSTATUS on LRCPC3 pair instructions resulting in two separate memory accessesDISPLAY Unpredictable FAR on LRCPC3 non atomic memory access at %{ADDR1} with FSTATUS %{FSTATUS1} or %{ADDR2} with FSTATUS %{FSTATUS2}. Fields:

ADDR1 unsigned int

Possible FAR address 1.

ADDR2 unsigned int

Possible FAR address 2.

FSTATUS1 unsigned int

Possible fault status value 1.

FSTATUS2 unsigned int

Possible fault status value 2.

ArchMsg.Warning.unpred_async_tag_check_fail_on_sve_speculative_load_do_update_ffr

Unpredictable set to FALSE of the FFR element associated to SVE speculative load with an async tag check fail TAGS **UNPREDICTABLE**.

ArchMsg.Warning.unpredictable_a32_breakpoint

TAGS **UNPREDICTABLE** DISPLAY Unpredictable breakpoint on a A32 instruction at %{ADDR1} Offset %{OFFSET} BAS %{BAS} for a %{IS_ADDRESS_MISMATCH_BREAKPOINT:(Mismatch|Match)} Breakpoint number %{N:d} Breakpoint address %{ADDR2}. Fields:

ADDR1 unsigned int

Address.

ADDR2 unsigned int

Breakpoint address.

BAS unsigned int

BAS field.

IS_ADDRESS_MISMATCH_BREAKPOINT bool

Is a Mismatch breakpoint, not a Match breakpoint.

N unsigned int

Breakpoint number.

OFFSET unsigned int

Always has a value of zero.

ArchMsg.Warning.unpredictable_banked_register_access

TAGS **UNPREDICTABLE** DISPLAY Access to banked register (R=%{R:d} SYSm=%{SYSM:2x}) is unpredictable. Fields:

R unsigned int

R bit.

SYSM unsigned int

SYSm.

ArchMsg.Warning.unpredictable_ccfail_undef

TAGS **UNPREDICTABLE** DISPLAY An instruction %{OPCODE} at VA=%{VADDR} failed its condition codes check but would otherwise have caused a non-data-dependent trap or undefined exception. Behaviour is implementation-defined choice of exception or nop. Fields:

OPCODE unsigned int

opcode of the instruction.

VADDR unsigned int

Virtual address of the instruction being executed.

ArchMsg.Warning.unpredictable_ici_bits

TAGS **UNPREDICTABLE** DISPLAY ICI bits do not apply or are architecturally unpredictable for instruction.

ArchMsg.Warning.unpredictable_t32_breakpoint

TAGS **UNPREDICTABLE** DISPLAY Unpredictable breakpoint on a T32 instruction at %{ADDR1} Offset %{OFFSET} BAS %{BAS} for a %{IS_ADDRESS_MISMATCH_BREAKPOINT:(Mismatch|Match)} Breakpoint number %{N:d} Breakpoint address %{ADDR2}. Fields:

ADDR1 unsigned int

Address.

ADDR2 unsigned int

Breakpoint address.

BAS unsigned int

BAS field.

IS_ADDRESS_MISMATCH_BREAKPOINT bool

Is a Mismatch breakpoint, not a Match breakpoint.

N unsigned int

Breakpoint number.

OFFSET unsigned int

Indicates which halfword of a 32-bit Thumb instruction the breakpoint is set on. 0 for the first halfword or if the instruction is 16 bits wide, or 1 for the second halfword.

ArchMsg.Warning.unpredictable_t32_vector_catch

TAGS **UNPREDICTABLE** DISPLAY Unpredictable Vector catch exception on a T32 instruction at %{ADDR} Offset %{OFFSET}. Fields:

ADDR unsigned int

Address.

OFFSET unsigned int

Indicates which halfword of a 32-bit Thumb instruction the vector catch exception is set on. 0 for the first halfword or if the instruction is 16 bits wide, or 1 for the second halfword.

ArchMsg.Warning.unpredictable_watchpoint_far

TAGS **UNPREDICTABLE** DISPLAY The value for FAR/EDWAR for hitting this watchpoint may be anywhere in the range %{LOWER_BOUND} to %{UPPER_BOUND}. Fields:

LOWER_BOUND unsigned int

the lowest address accessed by the instruction that triggered the watchpoint.

UPPER_BOUND unsigned int

the highest watchpointed address accessed by that instruction.

ArchMsg.Warning.unpredictable_watchpoint_far_isscmp

DISPLAY The value for FAR/EDWAR for hitting this watchpoint may be anywhere in the range %{LOWER_BOUND} to %{UPPER_BOUND}. Fields:

LOWER_BOUND unsigned int

the lowest address accessed by the instruction that triggered the watchpoint.

UPPER_BOUND unsigned int

the highest watchpointed address accessed by that instruction.

ArchMsg.Warning.warning_AdvSIMDExpandImmUnexpectedZero

TAGS UNPREDICTABLE DISPLAY AdvSIMDExpandImm may treat this immediate value as UNPREDICTABLE.

ArchMsg.Warning.warning_ConditionalSMC

TAGS UNPREDICTABLE DISPLAY SMC instruction has UNPREDICTABLE effects when conditional and when combined with traps.

ArchMsg.Warning.warning_TLBID_out_of_range

TLBID out of range in TLBI according to the configuration of NIS, NVIS, NOS and NVOS. Fields:

NIS unsigned int

NIS.

NOS unsigned int

NOS.

NVIS unsigned int

NVIS.

NVOS unsigned int

NVOS.

TLBID unsigned int

The TLBID value used in the TLBI instruction.

ArchMsg.Warning.warning_TTL_or_TG_mismatches_current_translation_granule

TTL or TG mismatches current translation granule in TLBI VA/RVA/IPA/RIPA. Fields:

CURRENT_TG unsigned int

Current TG as indicated by TCR_ELx or VTCR_EL2.

TLBI_TG unsigned int

TG (or TTL[3:2]) used in TLBI instruction.

ArchMsg.Warning.warning_access_crosses_page_boundary

TAGS UNPREDICTABLE DISPLAY Access crosses page boundary.

ArchMsg.Warning.warning_access_crosses_page_boundary_has_fault

TAGS UNPREDICTABLE DISPLAY Access that crosses page boundary has fault(s).

ArchMsg.Warning.warning_access_crosses_page_boundary_has_fault_on_both_pages

TAGS **UNPREDICTABLE** DISPLAY Access that crosses page boundary has a fault on each page.

Fields:

ADDR1 unsigned int

FAR address on first page.

ADDR2 unsigned int

FAR address on second page.

FSTATUS1 unsigned int

Fault status on first page.

FSTATUS2 unsigned int

Fault status on second page.

ArchMsg.Warning.warning_access_crosses_page_boundary_spanning_different_memory

TAGS **UNPREDICTABLE** DISPLAY Access at address %{ADDR} crosses page from MEM_TYPE %{MEMTYPE_PAGE1} to %{MEMTYPE_PAGE2}. Fields:

ADDR unsigned int

address of access crossing page.

MEMTYPE_PAGE1 signed int

First Page's memory type (0-Normal, 1-Non-Normal).

MEMTYPE_PAGE2 signed int

Second Page's memory type (0-Normal, 1-Non-Normal).

ArchMsg.Warning.warning_access_wraps_around_memory

TAGS **UNPREDICTABLE** DISPLAY Access wraps around memory in %{WIDTH:d}bit mode.

Fields:

WIDTH unsigned int

Address width.

ArchMsg.Warning.warning_bcr_linking_status

TAGS **UNPREDICTABLE** DISPLAY DBGBCR.{BT2,BT} combination gives unpredictable behaviour.

ArchMsg.Warning.warning_bcr_mask_reserved

TAGS **UNPREDICTABLE** DISPLAY DBGBCR_ELn.MASK is set to a reserved value.

ArchMsg.Warning.warning_bvr_masked_bit_not_zero

TAGS **UNPREDICTABLE** DISPLAY DBGBVRn_EL1.VA contains a bit masked by DBGBCRn_EL1.MASK that is not zero.

ArchMsg.Warning.warning_ccsidr_unimplemented_level

TAGS **UNPREDICTABLE** DISPLAY CCSIDR read while CSSELR %{CSSELR:x} points to an unimplemented cache level. Fields:

CSSELR unsigned int

current effective value of CSSELR.

ArchMsg.Warning.warning_change_to_ns_when_tge_set

TAGS **UNPREDICTABLE** DISPLAY Attempting to change NS to 1 when HCR.TGE = 1.

ArchMsg.Warning.warning_contiguous_bit_check_abort

TAGS **UNPREDICTABLE** DISPLAY An abort occurred whilst attempting to check TLB entry is contiguous at %{ENTRY_ADDR}. Fields:

ENTRY_ADDR unsigned int

address of conflicting TLB entry.

ArchMsg.Warning.warning_contiguous_bit_error

TAGS **UNPREDICTABLE** DISPLAY TLB entry at %{ENTRY_ADDR} was not contiguous with entry at %{CONTIG_ADDR} Contents are %{ENTRY_DATA} but expected %{CONTIG_DATA}. Fields:

CONTIG_ADDR unsigned int

address of the TLB first read, with which this entry is expected to be contiguous.

CONTIG_DATA unsigned int

expected contents based on the entry at CONTIG_ADDR.

ENTRY_ADDR unsigned int

address of conflicting TLB entry.

ENTRY_DATA unsigned int

contents of conflicting TLB entry.

ArchMsg.Warning.warning_contiguous_bit_error_D128

TAGS **UNPREDICTABLE** DISPLAY TLB entry at %{ENTRY_ADDR} was not contiguous with entry at %{CONTIG_ADDR} Contents are %{ENTRY_DATA_HI} %{ENTRY_DATA_LO} but expected %{CONTIG_DATA}. Fields:

CONTIG_ADDR unsigned int

address of the TLB first read, with which this entry is expected to be contiguous.

CONTIG_DATA unsigned int

expected contents based on the entry at CONTIG_ADDR.

ENTRY_ADDR unsigned int

address of conflicting TLB entry.

ENTRY_DATA_HI unsigned int

Higher bit contents of conflicting TLB entry.

ENTRY_DATA_LO unsigned int

Lower bit contents of conflicting TLB entry.

ArchMsg.Warning.warning_cp10_cp11_mismatch

TAGS **UNPREDICTABLE** DISPLAY Unpredictable configuration of CP10 and CP11 controls at EL %{EL:d}. Fields:

EL unsigned int

EL owning the control.

ArchMsg.Warning.warning_cp10_cp11_reserved_value

TAGS **UNPREDICTABLE** DISPLAY Unpredictable configuration of CPACR CP10 ({CP10}) or CP11 ({CP11}). Fields:

CP10 unsigned int

CP10 access.

CP11 unsigned int

CP11 access.

ArchMsg.Warning.warning_csselr_level_out_of_range

TAGS **UNPREDICTABLE** DISPLAY CSSELR.Level out of range, written value {LEVEL} with only {IMPLEMENTED:d} levels implemented. Fields:

IMPLEMENTED unsigned int

number of cache levels implemented.

LEVEL unsigned int

written value of CSSELR.Level.

TYPE enum

The type of cache selected by CSSELR.InD and TnD.

ArchMsg.Warning.warning_debug_flow_control_bits_not_obeyed

TAGS **UNPREDICTABLE** DISPLAY Flow control bits for {REGISTER} were not obeyed, causing {ERROR}. Fields:

ERROR enum

type of error resulting.

REGISTER enum

register under consideration.

ArchMsg.Warning.warning_debug_register_access_during_reset

TAGS **UNPREDICTABLE** DISPLAY Unpredictable {IS_WRITE:(read|write)} access to debug register offset {OFFSET} during reset. Fields:

IS_WRITE unsigned int

Write Not Read.

OFFSET unsigned int

Register Offset.

ArchMsg.Warning.warning_decode_cps_inconsistent_fields

TAGS **UNPREDICTABLE** DISPLAY The mode setting fields in the CPS instruction are inconsistent.

ArchMsg.Warning.warning_decode_invalid_state

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable in current exception-level/security state.

ArchMsg.Warning.warning_default_cacheable_mmu_on

TAGS **UNPREDICTABLE** DISPLAY Default cacheable is enabled (HCR.DC = 1) while MMU is enabled.

ArchMsg.Warning.warning_default_cacheable_vmmu_off

TAGS **UNPREDICTABLE** DISPLAY Default cacheable is enabled (HCR.DC = 1) while HCR.VM = 0.

ArchMsg.Warning.warning_deprecated_wvr_bit_2_set

TAGS **UNPREDICTABLE** DISPLAY DBGWVR%{N:d}_EL1=%{WVR:h} has bit 2 set, deprecated in ARMv8. Fields:

N unsigned int

index.

WVR unsigned int

DBGWVR.

ArchMsg.Warning.warning_eret_while_software_step_active_pending

Missing ISB between setting MDSCR_EL1.SS and ERET TAGS **UNPREDICTABLE**.

ArchMsg.Warning.warning_exclusive_to_non_normal

TAGS **UNPREDICTABLE** DISPLAY Exclusive Access to Non-Normal Memory %{ADDR}. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_exclusive_to_non_writeback

TAGS **UNPREDICTABLE** DISPLAY Exclusive Access to Non-Writeback-Cacheable Memory %{ADDR}. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_execute_from_device_memory

TAGS **UNPREDICTABLE** DISPLAY An attempt was made to execute from Device memory.

ArchMsg.Warning.warning_illegal_cpsr_mode

TAGS **UNPREDICTABLE** DISPLAY Writing an illegal or unimplement mode (%{NEW_MODE:h}) to CPSR was unpredictable. Fields:

NEW_MODE unsigned int

New value of CPSR.M.

ArchMsg.Warning.warning_illegal_srs_mode

TAGS **UNPREDICTABLE** DISPLAY Illegal or **UNPREDICTABLE** mode (%{MODE:h}) used for SRS instruction. Fields:

MODE unsigned int

mode for Banked SP.

ArchMsg.Warning.warning_implementation_defined_read_debug_register_in_SCS

Read Access to Debug registers from the processor is IMPLEMENTATION DEFINED TAGS IMP_DEF. Fields:

PPB_OFFSET unsigned int
debug register offset.

ArchMsg.Warning.warning_implementation_defined_sequential_security_transitions_supported

The behavior of sequential instruction fetches that cross from non-secure to secure memory and contain an SG instruction is CONSTRAINED UNPREDICTABLE TAGS UNPREDICTABLE.

ArchMsg.Warning.warning_implementation_defined_stack_limit_check_supported

It is IMPLEMENTATION DEFINED whether stack pointer limit checking is performed for this instruction TAGS IMP_DEF. Fields:

ADDRESS unsigned int
Address of instruction.

OPCODE unsigned int
instruction opcode.

ArchMsg.Warning.warning_implementation_defined_taggednonsharedmemorytype

TAGS IMP_DEF DISPLAY Memory attribute has Tagged, non shared memory type; it is IMPLEMENTATION DEFINED whether the memory location is treated as Tagged or non-Tagged.

ArchMsg.Warning.warning_implementation_defined_write_debug_register_in_SCS

Write Access to Debug registers from the processor is IMPLEMENTATION DEFINED TAGS IMP_DEF. Fields:

DATA unsigned int
data attempted to be written.

PPB_OFFSET unsigned int
debug register offset.

ArchMsg.Warning.warning_invalid_tcr_granule

TAGS UNPREDICTABLE DISPLAY TCR.TG%{TG_ID:(0|1)} bits have been set to a granule size %{{REQUEST:d}}K not implemented - using %{{SUBSTITUTE:d}}K. Fields:

REQUEST signed int
page size requested (or 0 for reserved).

SUBSTITUTE signed int
best guess available page size.

TG_ID bool
bits TG0 or TG1.

ArchMsg.Warning.warning_load_multiple_user_registers_from_user_mode

TAGS UNPREDICTABLE DISPLAY An LDM(user registers) instruction executed from user mode.

ArchMsg.Warning.warning_mte_in_mem_attr_not_supported

TAGS UNPREDICTABLE DISPLAY Memory attribute %{ATTR} has Tagged memory type but Armv8.5-MTE is not implemented. Fields:

ATTR unsigned int

Memory attribute.

ArchMsg.Warning.warning_reserved_breakpoint_state_match

TAGS UNPREDICTABLE DISPLAY %{IS_BREAKPOINT:(Watchpoint|Breakpoint)} programmed with a reserved combination of HMC, SSC and %{IS_BREAKPOINT:(PAC|PMC)}. Fields:

IS_BREAKPOINT unsigned int

Is a breakpoint, not a watchpoint.

ArchMsg.Warning.warning_shareability

TAGS UNPREDICTABLE DISPLAY Unpredictable: combination of the 1st and 2nd stages of translation is Normal Inner Non-Cacheable, Outer Non-Cacheable.

ArchMsg.Warning.warning_software_step_set_while_enabled

TAGS UNPREDICTABLE DISPLAY MDSCR_EL1.SS set to 1 while software step debug exceptions are enabled.

ArchMsg.Warning.warning_thumb_instruction_wraps_around_memory

TAGS UNPREDICTABLE DISPLAY Thumb instruction wraps around memory.

ArchMsg.Warning.warning_ttbr_sbz_bits_are_not_zero

TAGS UNPREDICTABLE DISPLAY %{TTBR}[%{MSB:d}:0] should be zero, but are not. Fields:

MSB unsigned int

x-1.

TTBR enum

which TTBR register.

ArchMsg.Warning.warning_unaligned_address_dbgdtrrx_write

TAGS UNPREDICTABLE DISPLAY Unaligned address %{ADDR} in X0/R0 when DBGDTRRX written in memory access mode. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unaligned_address_dbgdtrtx_read

TAGS UNPREDICTABLE DISPLAY Unaligned address %{ADDR} in X0/R0 when DBGDTRTX read in memory access mode. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unaligned_to_device

TAGS UNPREDICTABLE DISPLAY Unaligned Access to Device Memory %{ADDR}. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unaligned_to_strongly_ordered

TAGS **UNPREDICTABLE** DISPLAY Unaligned Access to Strongly Ordered Memory %{ADDR}.

Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unknown_sau_rnr

TAGS **UNPREDICTABLE** DISPLAY SAU_RNR was set to an unsupported value.

ArchMsg.Warning.warning_unpredictable_AIRCR_PRIS_and_BFHFNMINs

TAGS **UNPREDICTABLE** DISPLAY The effect of setting both AIRCR.BFHFNMINs and AIRCR.PRIS to 1 is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_AIRCR_VECTCLRACTIVE_when_not_in_debug

TAGS **UNPREDICTABLE** DISPLAY The effect of writing a 1 to AIRCR.VECTCLRACTIVE if the processor is not halted in Debug state is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_AIRCR_VECTRESET_and_SYSRESETREQ

TAGS **UNPREDICTABLE** DISPLAY When the processor is halted in Debug state, if a write to the register writes a 1 to both VECTRESET and SYSRESETREQ, the behavior is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_AIRCR_VECTRESET_when_not_in_debug

TAGS **UNPREDICTABLE** DISPLAY The effect of writing a 1 to AIRCR.VECTRESET if the processor is not halted in Debug state is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_AIRCR_incorrect_VKEY

TAGS **UNPREDICTABLE** DISPLAY The value 0x05FA must be written to AIRCR.VKEY, otherwise the register write is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_EXC_RETURN_Reserved_Bit

EXC_RETURN[23:7] are reserved with the special condition that all bits should be written as one. Values other than all 1s are **UNPREDICTABLE**. TAGS **UNPREDICTABLE**. Fields:

EXC_RETURN unsigned int

Special PC value.

SBO_MASK unsigned int

Which bits Should Be One to not UNPRED.

ArchMsg.Warning.warning_unpredictable_PMEVTYPEPn_ELO_VS_reserved_value

TAGS **UNPREDICTABLE** DISPLAY Unpredictable reserved value of PMEVTYPEN<n>_ELO.VS.

ArchMsg.Warning.warning_unpredictable_change_to_DEMCR_MON_STEP_at_insufficient_priority

TAGS **UNPREDICTABLE** DISPLAY The effect of changing DEMCR.MON_STEP at an execution priority that is lower than the priority of the DebugMonitor exception is **CONSTRAINED UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_change_to_priority_of_active_exception

TAGS UNPREDICTABLE DISPLAY Changing the priority of an active exception is UNPREDICTABLE.

ArchMsg.Warning.warning_unpredictable_clear_lspact

TAGS UNPREDICTABLE DISPLAY UNPREDICTABLE state on context switch. Fields:

DRVEXC_TAKEN unsigned int

Derived exception taken, not pended.

REACHED_LAST_STORE unsigned int

Only the last faulted.

ArchMsg.Warning.warning_unpredictable_dbg_exit_unaligned_dlr

TAGS UNPREDICTABLE DISPLAY Debug exit to AArch32 with DLR[0] set to 1. PC[0] can be set to 0 or to DLR[0].

ArchMsg.Warning.warning_unpredictable_exception_catch

TAGS UNPREDICTABLE DISPLAY Generation of an exception catch event is unpredictable.

ArchMsg.Warning.warning_unpredictable_exception_return_inconsistent_state

TAGS UNPREDICTABLE DISPLAY state on exception return is unpredictable.

ArchMsg.Warning.warning_unpredictable_exception_return_instruction

TAGS UNPREDICTABLE DISPLAY exception return instruction is unpredictable.

ArchMsg.Warning.warning_unpredictable_in_debug_state

TAGS UNPREDICTABLE DISPLAY Instruction is UNPREDICTABLE when executed in debug state.

ArchMsg.Warning.warning_unpredictable_mair_encoding

TAGS UNPREDICTABLE DISPLAY Memory attribute %{ATTR} has invalid MAIR encoding; this may lead to unpredictable results. Fields:

ATTR unsigned int

Memory attribute.

ArchMsg.Warning.warning_unpredictable_pmu_counter_access

TAGS UNPREDICTABLE DISPLAY Access to PMU counters from non-secure EL0 or EL1 is UNPREDICTABLE with MDCR_EL2.HPMN set to 0.

ArchMsg.Warning.warning_unpredictable_prioritization_breakpoint_match_vector_catch

TAGS UNPREDICTABLE DISPLAY Two events with the same priority occurred at the same instruction: Address Matching Vector Catch debug event, and Breakpoint debug event. It is constrained unpredictable which is taken.

ArchMsg.Warning.warning_unpredictable_stack_selection

TAGS UNPREDICTABLE DISPLAY UNPREDICTABLE state on context switch.

ArchMsg.Warning.warning_unpredictable_tsize_out_of_range

TAGS UNPREDICTABLE DISPLAY %{TCR}.T%{TTBR:d}SZ (%{TSIZE:d} bits) is out of range. Must be between 25 and %{TSIZE_MAX:d}. Fields:

TCR enum

Which TCR.

TSize signed int

Value of TSize.

TSize_MAX signed int

The maximum allowed value.

TTBR unsigned int

Which TTBR region.

ArchMsg.Warning.warning_unpredictable_unaligned_pc_as_base_register

TAGS **UNPREDICTABLE** DISPLAY The PC value must be word-aligned, otherwise the behavior of the instruction is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_unaligned_pop_stack

TAGS **UNPREDICTABLE** DISPLAY **CONSTRAINED UNPREDICTABLE** unaligned stack frame `%{FRAME_ADDR:x}` in exception-return. Fields:

FRAME_ADDR unsigned int

unaligned stack frame address.

ArchMsg.Warning.warning_unpredictable_unaligned_to_non_writeback

TAGS **UNPREDICTABLE** DISPLAY Unaligned Access to Non-Writeback-Cacheable Memory.

ArchMsg.Warning.warning_unpredictable_vmsa_memattrib

TAGS **UNPREDICTABLE** DISPLAY Unpredictable vmsa memory attribute with texcb value programmed to `%{TEXCB}` when tex remap is `%{TEX_REMAP}`. Fields:

TEXCB unsigned int

texcb value.

TEX_REMAP bool

use tex remap.

ArchMsg.Warning.warning_unpredictable_vtcr_t0sz_sl0

TAGS **UNPREDICTABLE** DISPLAY The combination of VTCR.TOSZ=`%{TOSZ:d}` and VTCR.SL0=`%{SL0:d}` is unpredictable. Fields:

SL0 signed int

VTCR.SL0, starting level for stage 2 translation table walks.

TOSZ signed int

VTCR.TOSZ, required input address range.

ArchMsg.Warning.warning_unpredictable_write_DHCSR

TAGS **UNPREDICTABLE** DISPLAY **UNPREDICTABLE** set of bit-changes in DHCSR. Fields:

NEW unsigned int

data attempted to be written.

OLD unsigned int

previous value.

ArchMsg.Warning.warning_unpredictable_write_SHCSR

TAGS UNPREDICTABLE DISPLAY UNPREDICTABLE setting a pending bit to 1 for an exception with priority \geq execution priority.

ArchMsg.Warning.warning_unsupported_access_to_memory_mapped_register

TAGS UNPREDICTABLE DISPLAY Access to memory mapped register is unsupported.

ArchMsg.Warning.warning_user_jmcr_access

TAGS UNPREDICTABLE DISPLAY User %{WRITE:(write|read)} access to JMCR is unpredictable. Fields:

WRITE unsigned int
access is write.

ArchMsg.Warning.warning_wcr_mask_and_bas

TAGS UNPREDICTABLE DISPLAY DBGWCR_EL1.MASK is non-zero and BAS is not 0xff.

ArchMsg.Warning.warning_wcr_mask_reserved

TAGS UNPREDICTABLE DISPLAY DBGWCR_ELn.MASK is set to a reserved value.

ArchMsg.Warning.warning_wcr_non_configuous_bas

TAGS UNPREDICTABLE DISPLAY DBGWCR_EL1.BAS has non-contiguous set of ones.

ArchMsg.Warning.warning_wvr_masked_bit_not_zero

TAGS UNPREDICTABLE DISPLAY DBGWVRn_EL1.VA contains a bit masked by DBGWCRn_EL1.MASK that is not zero.

ArchMsg.Why.why_branch_target_exception

A branch target exception has occurred. DISPLAY branch target exception: target pc=%{PC}, target opcode=%{OPCODE}. PRIMARY KEY PC. Fields:

BTYPE unsigned int
pstate.BTYPE at time branch taken.

OPCODE unsigned int
opcode at branch target.

PC unsigned int
Virtual Address of branch target.

ArchMsg.Why.why_illegal_state

An illegal mode change has occurred. DISPLAY illegal mode change: %{MESSAGE}. Fields:

MESSAGE string
The message.

ArchMsg.warning_invalid_interval_counter_reload_value

Invalid interval counter reload value in PMSIRR_EL1.INTERVAL: software should set this to a value greater than or equal to the minimum indicated by PMSIDR_EL1.Interval.

ArchMsg.warning_invalid_spe_buffer_write_pointer

Invalid value programmed in PMBPTR_EL1.PTR or PMBLIMITR_EL1.LIMIT: Current write pointer must be at least one sample record below the write limit pointer and PMBPTR_EL1.PTR[63:56] must be equal to PMBLIMITR_EL1.LIMIT[63:56].

BRANCH_MISPREDICT

Simulating branch mispredict. Fields:

PC unsigned int

Origin address (or 0 if unavailable).

BRA_DIR

Direct branches, to immediate address. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISET enum

The instruction set of the branch instruction.

IS_COND bool

Indicates if this is a conditional branch.

IS_HINTED bool

Whether the conditional instruction is hinted.

IS_ISB bool

Whether the instruction is ISB.

IS_LINK bool

Indicates if this is a branch with link.

PC unsigned int

The address of the branch instruction.

TARGET_ISET enum

The instruction set after the branch.

TARGET_PC unsigned int

The address the instruction branches to.

BRA_INDIR

Indirect branches, perhaps to a register. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISSET enum

The instruction set of the branch instruction.

IS_COND bool

Indicates if this is a conditional branch.

IS_HINTED bool

Whether the conditional instruction is hinted.

IS_ISB bool

Whether the instruction is ISB.

IS_LINK bool

Indicates if this is a branch with link.

PC unsigned int

The address of the branch instruction.

TARGET_ISET enum

The instruction set after the branch.

TARGET_PC unsigned int

The address the instruction branches to.

BRA_INDIR_CCFAIL

Indirect branches fail, perhaps to a register.

BRA_RET_CCFAIL

Branch return fail, perhaps to a register.

CACHE_MAINTENANCE_OP

Cache Maintenance Operation. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

DATA unsigned int

Specified MVA or Set/way.

FUNCTION enum

Clean / Invalidate.

SCOPE enum

Affected region.

SIDE enum

Inst / Data.

CCFAIL

Conditional instruction condition check fail. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CCFAIL_UNC

Unconditional instruction condition check fail. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CCPASS

Conditional instruction condition check pass. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CCPASS_UNC

Unconditional instruction condition check pass. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CODE_CACHE_MAINT

Code cache maintenance. Fields:

END_INCL unsigned int

Inclusive end address for VA or PA requests.

NS bool

Non-secure world for PA.

START unsigned int

Start address for VA or PA requests.

TYPE enum

Request type.

COMPILE_BLOCK_END

Last instruction of basic block translated. Fields:

VADDR unsigned int

Address of next instruction after this basic block.

COMPILE_BLOCK_START

First instruction of basic block translated. Fields:

VADDR unsigned int

Address of first instruction in basic block.

COMPILE_INST

ARM instruction compiled. Fields:

DISASS string

Disassembly of the instruction.

ISSET enum

The instruction set of this instruction.

ITSTATE unsigned int

The ITSTATE current for the instruction.

OPCODE unsigned int

The opcode of the instruction.

PC unsigned int

The address of the instruction.

SIZE unsigned int

The size of the instruction in bytes.

CONTEXTIDR

Context ID Register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

NS bool

Secure or nonsecure banked register is accessed.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The new value written.

CONTEXT_SYNC

Called for every context synchronization event. Fields:

ARCH_CSE bool

Architectural Context Synchronization Event.

PC unsigned int

The address of the synchronization instruction.

SYSREG_SYNC bool

System register synchronization event.

CORE_ENDIAN

Core BE8 Big-Endian state changed. Fields:

BE8 bool

Core BE8 Big-Endian state.

CORE_INFO

Static processor attributes. Only triggered by a call to DumpState(). Fields:

ARCH_PROFILE enum

The architecture profile of the core.

CLUSTER_ID unsigned int

The cluster ID of this processor.

CORE_NUM unsigned int

The number of this core in an MP processor.

DETAILED_CORE_FEATURES enum

Detailed core features as a bitfield. Enum values are bitmasks to extract the info from the field.

FPU_VERSION enum

The VFP version implemented by the core.

MEM_ARCH enum

The memory architecture of the core.

MIDR_EL1 unsigned int

MIDR_EL1 value.

NUM_CORES unsigned int

The number of cores in this MP processor.

QUANTUM_SIZE unsigned int

The default quantum size of the core (in ticks).

SECURITY_FEATURES bool

Does the core have security features?.

CORE_LOADS

Processor load accesses. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ACQREL enum

Is this an acquire/release.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DATA unsigned int

The data read or written.

ELEMENT_SIZE unsigned int

Width of each element.

ENDIANNESS enum

Byte ordering of this access.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

LOCK enum

Normal, exclusive or locked access.

MECID unsigned int

Memory Encryption Context ID, if applicable.

MECID2 unsigned int

Memory Encryption Context ID of the second page, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PADDR_TAG unsigned int

If tag checked, the physical address tag for the access.

PAGE_SIZE unsigned int

Page size as $\log_2(\text{size in bytes})$.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK, 2=Exclusive Failed, 3=Unsupported.

SIZE unsigned int

Width of the access in bytes. Only required if DATA is not traced.

TAG_CHECKED bool

Is this access tag checked.

TAG_CHECKED2 bool

Is the second page access tag checked.

TRANS bool

Is this a translated access.

VADDR unsigned int

The virtual address of the access.

CORE_REGS

Changes of the core registers R0 to R14. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number, 0 to 14.

MODE enum

Bank of the register accessed.

OLD_VALUE unsigned int

The old value overwritten.

PHYS_ID enum

The physical register accessed.

VALUE **unsigned int**

The new value written to the register.

CORE_REGS64

Changes of the core registers X0..X30, SP_ELn. Fields:

CORE_NUM **unsigned int**

Core number in a multi processor.

ID **enum**

The register number, 0-30 for X0-X30, >=32 for SP_ELn.

OLD_VALUE **unsigned int**

The old value overwritten.

VALUE **unsigned int**

The new value written to the register.

CORE_REGS64_READ

General purpose AArch64 register has been read X0..X30, SP_ELn. Fields:

ID **enum**

The register number, 0-30 for X0-X30, >=32 for SP_ELn.

VALUE **unsigned int**

The value read from the register.

CORE_STALLED_TRANSITION

Detect when a core is prevented from progressing PChannel states. Fields:

CURRENT_STATE **enum**

Current power mode of core.

REASON **string**

The reason a transition was stalled, and what must be done to unblock it.

TARGET_STATE **enum**

Desired power mode of core after transition.

CORE_STORES

Processor store accesses. Fields:

ACCESS_TYPE **enum**

The type of instruction performing the access.

ACQREL **enum**

Is this an acquire/release.

ATTR **unsigned int**

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DATA unsigned int

The data read or written.

ELEMENT_SIZE unsigned int

Width of each element.

ENDIANNESS enum

Byte ordering of this access.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

LOCK enum

Normal, exclusive or locked access.

MECID unsigned int

Memory Encryption Context ID, if applicable.

MECID2 unsigned int

Memory Encryption Context ID of the second page, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PADDR_TAG unsigned int

If tag checked, the physical address tag for the access.

PAGE_SIZE unsigned int

Page size as $\log_2(\text{size in bytes})$.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK, 2=Exclusive Failed, 3=Unsupported.

SIZE unsigned int

Width of the access in bytes. Only required if DATA is not traced.

TAG_CHECKED bool

Is this access tag checked.

TAG_CHECKED2 bool

Is the second page access tag checked.

TRANS bool

Is this a translated access.

VADDR unsigned int

The virtual address of the access.

CP14_READ

System Coprocessor register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP14 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc1 unsigned int

opcode 1.

opc2 unsigned int

opcode 2.

CP14_WRITE

System Coprocessor register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP14 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc1 unsigned int

opcode 1.

opc2 unsigned int

opcode 2.

CP15_READ

System Coprocessor register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP15 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc1 unsigned int

opcode 1.

opc2 unsigned int

opcode 2.

CP15_READ64

System Coprocessor register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP15 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc unsigned int

opcode 1.

CP15_WRITE

System Control Coprocessor register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP15 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc1 unsigned int

opcode 1.

opc2 unsigned int

opcode 2.

CP15_WRITE64

System Coprocessor register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP15 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc unsigned int

opcode 1.

CPSR

CPSR change. (In AArch64 PSTATE represented in SPSR format). Fields:

CORE_NUM unsigned int

Core number in a multi processor.

OLD_VALUE unsigned int

The old CPSR value.

UNKNOWN unsigned int

Bits within the register that have unknown value.

VALUE unsigned int

The new CPSR value.

CRYPTO_SPEC

Every crypto instruction speculatively executed.

CorePowerStateChange

Core power-up state has changed. DISPLAY Core powered %{PU:(up|down)}. Fields:

PU bool

Whether the core powered up.

DATA_CACHE_SET_ALLOC_TAGS

Set allocation tags from DC GVA instruction. Fields:

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

NSDESC unsigned int

The security state of the access.

PADDR unsigned int

The physical address of the access.

PAS enum

The physical address space of the page.

RESPONSE enum

0=Aborted, 1=OK.

SIZE unsigned int

The size of memory in bytes.

TAG unsigned int

The value of the tag being set.

TAGGING_TYPE enum

Type of Memory tagging (Physical or Virtual).

TAG_VADDR unsigned int

The virtual address of the Virtual Tag if Virtual tagging is used; otherwise 0.

VADDR unsigned int

Virtual address used for setting allocation tags.

DATA_CACHE_ZERO

Zero and invalidate cache line from DC ZVA instruction. Fields:

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

NSDESC unsigned int

The security state of the access.

PADDR unsigned int

The physical address of the access.

PADDR_TAG unsigned int

If tag checked, the physical address tag for the access.

PAS enum

The physical address space of the page.

RESPONSE enum

0=Aborted, 1=OK.

SIZE unsigned int

The size of zero'd memory in bytes.

TAG_CHECKED bool

Is this access tag checked.

VADDR unsigned int

Virtual address of the zero'd cache line.

DEBUG_EVENT

Hardware debug support event. Fields:

EVENT enum

Description of event.

VALUE unsigned int
data value.

DEBUG_TARGET_ISET

Hardware debug target ISA. Fields:

TARGET_ISET enum
The target instruction set of the debug state.

DMI_ALLOCATE

DMI allocated. Fields:

ACCESS_DIRECTION enum
READ / WRITE.

PADDR unsigned int
The physical address of the DMI.

PTR unsigned int
DMI range pointer.

SIDE enum
DSIDE / ISIDE.

SIZE unsigned int
The size of DMI.

VADDR unsigned int
The virtual address of the DMI..

DMI_HIT

DMI hit, access used DMI (only fire for small DMI regions i.e. less than a full page). Fields:

ACCESS_DIRECTION enum
READ / WRITE.

PADDR unsigned int
The physical address of the DMI.

SIDE enum
DSIDE / ISIDE.

SIZE unsigned int
The size of the DMI.

VADDR unsigned int
The virtual address of the DMI..

DMI_REVOKE

DMI revoked. Fields:

ACCESS_DIRECTION enum
READ / WRITE.

PADDR unsigned int

The physical address of the DMI.

SIDE enum

DSIDE / ISIDE.

SIZE unsigned int

The size of DMI.

VADDR unsigned int

The virtual address of the DMI..

END_COMPILE

Compilation end. Fields:

END_OF_PAGE_COUNT unsigned int

Number of basic blocks exited due to page boundary since START_COMPILE.

FETCHFAIL_COUNT unsigned int

Number of basic blocks exited due to ifetch failure START_COMPILE.

INST_COUNT unsigned int

Number of instructions compiled since START_COMPILE.

NONSEQ_COUNT unsigned int

Number of basic blocks exited due to non-sequential instructions since START_COMPILE.

PAGE_STRADDLE_COUNT unsigned int

Number of basic blocks exited due to unaligned instructions crossing page since START_COMPILE.

ESB

Error Synchronization Barrier hit. Fields:

ASYNC bool

ESB is async.

IMPLICIT bool

ESB is implicit.

EXCEPTION

Exception taken. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ESR unsigned int

The value assigned to ESR when exception is taken.

ESR_EC unsigned int

The Value assigned to ESR.EC when exception is taken.

IS_PHYSICAL bool

Physical or Virtual exception.

IS_TRAP bool

Exception coming from a trap.

LR unsigned int

The value assigned to the link register.

PC unsigned int

The location where the exception occurred.

PREFERRED_RETURN unsigned int

The preferred return address for the exception.

TAKEN_LOCALLY bool

Whether exception is taken locally(same exception level).

TARGET_ISET enum

The instruction set of the exception handler code.

TARGET_PC unsigned int

The address the exception branches to.

VECTOR enum

The exception vector.

VECTOR_OFFSET unsigned int

The offset of an exception vector from the base address.

EXCEPTION_END

Every exception completed.

EXCEPTION_INFO

Trace human-readable reason of the current exception for data/instruction aborts, floating point exceptions and unknown exceptions. Fields:

DESCRIPTION string

Reason for current exception.

ESR_EC unsigned int

The value of ESR.EC when exception is taken.

EXCEPTION_RAISE

Every exception raised.

EXCEPTION_RETURN

Branches on leaving exception. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISSET enum

The instruction set of the branch instruction.

IS_COND bool

Indicates if this is a conditional branch.

IS_HINTED bool

Whether the conditional instruction is hinted.

IS_ISB bool

Whether the instruction is ISB.

IS_LINK bool

Indicates if this is a branch with link.

PC unsigned int

The address of the branch instruction.

TARGET_ISSET enum

The instruction set after the branch.

TARGET_PC unsigned int

The address the instruction branches to.

EXCEPTION_RETURN_PREBRANCH

Exception return event that must occur before PC and processor state have updated.

EXCEPTION_START

Every exception started.

FIQ_TAKEN

FIQ taken exception.

FP_EXCEPTIONS_SIGNALLED

Trace which FP exceptions were signalled as a part of the FP operations performed by an instruction. Fields:

FP_EXCEPTIONS_SIGNALLED_MASK unsigned int

Indicates which FP exceptions were signalled by an instruction. This uses the same encoding as the cumulative floating-point exception bitfields in FPSR.

FP_STATE

Floating point state. Fields:

FPSR unsigned int

FPSR register value.

MASK unsigned int

The mask indicating updated cumulative bits.

FREQ_CHANGED

Core frequency changed. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The instruction count of this core.

NEW_FREQ signed int

The new frequency of this core (expressed in Hz).

OLD_FREQ signed int

The old frequency of this core (expressed in Hz).

HLT

HLT instruction occurred. Fields:

IMM16 unsigned int

The 16-bit immediate value encoded in HLT instruction.

INST

Every instruction executed. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DEBUG_STATE bool

The instruction is executed in debug state.

DISASS string

Disassembly of instruction.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISSET enum

The current instruction set.

ITSTATE unsigned int

The current ITSTATE.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

MODE enum

The mode the core is in.

NS unsigned int

The core's non-secure bit.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

OPCODE unsigned int

The opcode of the instruction.

PADDR unsigned int

The physical address of the instruction.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the instruction.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the page.

PC unsigned int

The address of the instruction.

SECURITY_STATE enum

The core's security state.

SIZE unsigned int

The size of the instruction in bytes.

INST_END

Every instruction completed.

INST_START

Every instruction started. Fields:

DEBUG_STATE bool

The instruction is executed in debug state.

ISET enum

The current instruction set.

MODE enum

The mode the core is in.

NS enum

The current Secure State.

PC unsigned int

The address of the conditional instruction.

SECURITY_STATE enum

The current Security State.

INST_STRADDLE

Instruction straddles boundary. Fields:

DEBUG_STATE bool

The instruction is executed in debug state.

ISET enum

The current instruction set.

MODE enum

The mode the core is in.

NS enum

The current Secure State.

PC unsigned int

The address of the conditional instruction.

SECURITY_STATE enum

The current Security State.

IRQ_TAKEN

IRQ taken exception.

ISB

ISB instruction occurred. Fields:

PC unsigned int

The address of the ISB instruction.

LOAD_MULTIPLE_REGS_LIST

List of destination registers (not source) of a multiple registers load instruction. Fields:

REG string

Register to add into the list.

LOCAL_MONITOR

Local monitor activity. Fields:

PADDR unsigned int

Local Monitor Address.

State enum

State of the monitor (Open/Exclusive).

MAPPING_CREATE

A memory mapping has been created. This is an implementation detail of the model and so may change without notice. Fields:

ASID unsigned int

The ASID used during translation, or 0 if no ASID was used.

IS_VALID bool

Whether the mapping is valid or not.

PADDR unsigned int

Base physical address of region, or 0 if not known.

SIDE enum

I-side or D-side.

SIZE unsigned int

Size of the page, or 0 if not known.

VADDR unsigned int

Base virtual address of region.

VMID unsigned int

The VMID used during translation, or 0 if no VMID was used.

MAPPING_DESTROY

A memory mapping has been destroyed. This is an implementation detail of the model and so may change without notice. Fields:

ASID unsigned int

The ASID used during translation.

CAUSE enum

The cause of invalidation.

PADDR unsigned int

Base physical address of page.

SIDE enum

I-side or D-side.

SIZE unsigned int

Size of the page.

VADDR unsigned int

Base virtual address of page.

VMID unsigned int

The VMID used during translation.

MEMMAP_DEBUG_READ

Memory mapped reads to the debug registers. Fields:

ADDR unsigned int

address.

CORE_NUM unsigned int

Core number in a multi processor.

EXT bool

Whether access is from an external device (such as the DAP).

NS enum

Secure state of the access.

REG_NAME string

Name of the debug register accessed.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

MEMMAP_DEBUG_WRITE

Memory mapped writes to the debug registers. Fields:

ADDR unsigned int

address.

CORE_NUM unsigned int

Core number in a multi processor.

EXT bool

Whether access is from an external device (such as the DAP).

NS enum

Secure state of the access.

REG_NAME string

Name of the debug register accessed.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The new value written.

MEMTAG_LOADS

Allocation tag memory read; when attributes and system configuration allow tag accesses. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

NUM_GRANULES unsigned int

Number of tags to be Loaded/stored.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK.

TAGGING_TYPE enum

Type of Memory tagging (Physical or Virtual).

TAGS unsigned int

The tags read or written. Tags for subsequent granules would be found at subsequent bytes in this field (One byte for each tag).

TAG_VADDR unsigned int

The virtual address of the Virtual Tag if Virtual tagging is used; otherwise 0.

VADDR unsigned int

The virtual address of the access.

MENTAG_LOAD_INST

Allocation tag load instruction; regardless of tags being actually accessed. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate

(Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

NUM_GRANULES unsigned int

Number of tags to be Loaded/stored.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK.

TAGGING_TYPE enum

Type of Memory tagging (Physical or Virtual).

TAGS unsigned int

The tags read or written. Tags for subsequent granules would be found at subsequent bytes in this field (One byte for each tag).

TAG_VADDR unsigned int

The virtual address of the Virtual Tag if Virtual tagging is used; otherwise 0.

VADDR unsigned int

The virtual address of the access.

MEMTAG_STORES

Allocation tag memory write; when attributes and system configuration allow tag accesses. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

NUM_GRANULES unsigned int

Number of tags to be Loaded/stored.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK.

TAGGING_TYPE enum

Type of Memory tagging (Physical or Virtual).

TAGS unsigned int

The tags read or written. Tags for subsequent granules would be found at subsequent bytes in this field (One byte for each tag).

TAG_VADDR unsigned int

The virtual address of the Virtual Tag if Virtual tagging is used; otherwise 0.

VADDR unsigned int

The virtual address of the access.

MEMTAG_STORE_INST

Allocation tag store instruction; regardless of tags being actually accessed. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes

([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

NUM_GRANULES unsigned int

Number of tags to be Loaded/stored.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK.

TAGGING_TYPE enum

Type of Memory tagging (Physical or Virtual).

TAGS unsigned int

The tags read or written. Tags for subsequent granules would be found at subsequent bytes in this field (One byte for each tag).

TAG_VADDR unsigned int

The virtual address of the Virtual Tag if Virtual tagging is used; otherwise 0.

VADDR unsigned int

The virtual address of the access.

MMU_D128_TTB_READ

This event is triggered by reads caused by a D128 translation table walk. Fields:

ABORTED bool

Was the walk successful.

DATA_HIGH unsigned int

The data upper bits read.

DATA_LOW unsigned int

The data lower bits read.

IPA unsigned int

For stage 1, the IPA of the read.

LEVEL unsigned int

Translation table level.

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

OUTADDR unsigned int

The output physical address.

PADDR unsigned int

The physical address of the read.

PASmem enum

Physical Address Space of the accessed memory.

PASreq enum

Physical Address Space of the requested lookup.

REGIME_EL enum

Entry matches in this translation regime.

REQUESTER enum

The requester of this table walk.

SIDE enum

Inst / Data.

STAGE unsigned int

Translation stage.

MMU_D128_TTB_WRITE

This event is triggered by writes caused by a D128 translation table walk. Fields:

ABORTED bool

Was the walk successful.

DATA_HIGH unsigned int

The data upper bits written.

DATA_LOW unsigned int

The data lower bits written.

IPA unsigned int

For stage 1, the IPA of the write.

IS_HACDBS_ACCESS bool

If the access was due to HACDBS_CLEAN.

LEVEL unsigned int

Translation table level.

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

OUTADDR unsigned int

The output physical address.

PADDR unsigned int

The physical address of the write.

PASmem enum

Physical Address Space of the accessed memory.

PASreq enum

Physical Address Space of the requested lookup.

REGIME_EL enum

Entry matches in this translation regime.

SIDE enum

Inst / Data.

STAGE unsigned int

Translation stage.

MMU_TRANS

Address translation information. Fields:

ASID unsigned int

Address space identifier.

CORE_NUM unsigned int

Core number in a multi processor.

Hyp bool

Entry matches in Hyp state only.

INNERCACHE_RA bool

Is the inner cache allocate on read.

INNERCACHE_TRANSIENT bool

Is the inner write-through transient.

INNERCACHE_TYPE enum

Inner Caching scheme (NC/MB/WA).

INNERCACHE_WA bool

Is the inner cache allocate on write.

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MEMTYPE enum

Memory type.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC enum

Is secure side supposed to access secure or nonsecure memory.

OUTERCACHE_RA bool

Is the outer cache allocate on read.

OUTERCACHE_TRANSIENT bool

Is the outer write-through transient.

OUTERCACHE_TYPE enum

Outer Caching scheme (NC/MB/WA).

OUTERCACHE_WA bool

Is the outer cache allocate on write.

PADDR unsigned int

Physical address of the access.

PAGESIZE unsigned int

Page size as $\log_2(\text{size})$.

PAS enum

PAS supposed to be accessed is secure or nonsecure memory or Root/Realm if RME is implemented.

SH enum

Shareability.

SIDE enum

Inst / Data.

STAGE1_PERM unsigned int

Stage 1 permissions, [5:3] Privileged access (XWR) [2:0] User access (XWR).

STAGE2_PERM unsigned int

Stage 2 permission mask [5:3] Privileged access (XWR) [2:0] User access (XWR).

VADDR unsigned int

Virtual address of the access.

VMID unsigned int

Virtual machine identifier.

XS bool

XS Attribute.

nG enum

Flag indicating whether ASID will be matched.

MMU_TTB_READ

This event is triggered by reads caused by a translation table walk. Fields:

ABORTED bool

Was the walk successful.

DATA unsigned int

The data read.

DESC_KIND enum

The encoding scheme of the descriptors used for this walk.

IPA unsigned int

For stage 1, the IPA of the read.

LEVEL unsigned int

Translation table level.

LPAE bool

Is this for an LPAE translation (*DEPRECATED*, use DESC_KIND).

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSmem bool

Non secure memory access.

NSreq bool

Non secure request.

OUTADDR unsigned int

The output physical address.

PADDR unsigned int

The physical address of the read.

PASmem enum

Physical Address Space of the accessed memory.

PASreq enum

Physical Address Space of the requested lookup.

REGIME_EL enum

Entry matches in this translation regime.

REQUESTER enum

The requester of this table walk.

SIDE enum

Inst / Data.

STAGE unsigned int

Translation stage.

MMU_TTB_WRITE

This event is triggered by writes caused by a translation table walk. Fields:

ABORTED bool

Was the walk successful.

DATA unsigned int

The data written.

DESC_KIND enum

The encoding scheme of the descriptors used for this walk.

IPA unsigned int

For stage 1, the IPA of the write.

IS_HACDBS_ACCESS bool

If the access was due to HACDBS_CLEAN.

LEVEL unsigned int

Translation table level.

LPAE bool

Is this for an LPAE translation (*DEPRECATED*, use DESC_KIND).

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSmem bool

Non secure memory access.

NSreq bool

Non secure request.

OUTADDR unsigned int

The output physical address.

PADDR unsigned int

The physical address of the write.

PASmem enum

Physical Address Space of the accessed memory.

PASreq enum

Physical Address Space of the requested lookup.

REGIME_EL enum

Entry matches in this translation regime.

SIDE enum

Inst / Data.

STAGE unsigned int

Translation stage.

MODE_CHANGE

Mode change. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

MODE enum

The new mode.

NON_SECURE enum

The core's new non-secure bit.

OLD_MODE enum

The old mode.

SECURITY_STATE enum

The new security state.

NOT_TAG_CHECKED_INST

MTE Tag not checked for current instruction.

PERIODIC

Called for every quantum. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The instruction count of this CPU.

PC unsigned int

The address of the next instruction to be executed on this CPU.

PMU_COUNTER_OVERFLOW

PMU counter overflow. Fields:

COUNTER_GROUP enum

Counter group.

COUNTER_RANGE enum

Counter range.

EVENT_ID unsigned int

Event Identifier, field is valid only for Event counter group.

INDEX unsigned int

Counter index (as selected by PMSELR), field is valid only for Event/Cycle counter groups.

INTERRUPT bool

Is interrupt enabled on overflow for this counter.

POINTER_AUTH

Called when execution hits a pointer authentication check. Fields:

AUTH_STATE bool

Authentication result.

KEY_USED enum

Pointer authentication key.

POTINDEX_UPDATE

Value of POT index for current pc changed. Fields:

NEW_POTINDEX unsigned int

New POTIndex.

OLD_POTINDEX unsigned int

Old POTIndex.

PREFETCH_MEMORY64

Prefetch from PRFM or PRFUM instructions. Fields:

PRFOP enum

Prefetch hint.

VADDR unsigned int

Virtual address of the location that should be prefetched.

PRELOAD64_REQUEST

Preload requests from PRFM or PRFUM instructions. Fields:

PRFOP enum

Prefetch hint.

VADDR unsigned int

Virtual address of the location that should be prefetched.

PRELOAD_DATA

Data preload from PLD instruction. Fields:

VADDR unsigned int

Virtual address of the data that should be preloaded.

PRELOAD_INST

Instruction preload from PLI instruction. Fields:

VADDR unsigned int

Virtual address of the instruction that should be preloaded.

PRE_CORE_LOAD

Trace just before a core load. Fields:

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

PRE_CORE_STORE

Trace just before a core store. Fields:

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

PRE_TTB_READ

This event is triggered before reads caused by a translation table walk. Fields:

ADDR unsigned int

The physical address of the read.

PSTATE

PSTATE fields update. Fields:

ALLINT unsigned int

All interrupt mask bit.

BTYPE unsigned int

Branch target identification bit.

CORE_NUM unsigned int

Core number in a multi processor.

DAIF unsigned int

Exception mask bits.

DIT unsigned int

Data Independent Timing bit.

EL unsigned int

Current Exception level field.

EXLOCK unsigned int

Exception return state lock bit.

IL unsigned int

Illegal Execution state bit.

NZCV unsigned int

Condition flags.

PACM unsigned int

PACM bit.

PAN unsigned int

Privileged Access Never state bit.

PM unsigned int

PMU exception mask bit.

PPEND unsigned int

PMU exception pending bit.

SM unsigned int

StreamingMode status bit.

SP unsigned int

Stack pointer register selection bit.

SS unsigned int

Software Step bit.

SSBS unsigned int

Speculative Store Bypass Safe bit.

TCO unsigned int

Tag Check Override bit.

TOUCHED unsigned int

Mask of the fields that have been touched since this trace has last been emitted. Note that touched does not necessarily mean the value changed. The index of the field is the bit index in the mask plus the index of this field plus one.

UAO unsigned int

User Access Override bit.

UINJ unsigned int

Undefined Instruction exceptions bit.

ZA unsigned int

ZA storage and ZT0 register enable bit.

nRW unsigned int

Current Execution state bit.

PSTATE_PACTIVE_OPMODE_TRANSITION

Trace pactive and Opmode mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Operating mode after the transition.

OLD_STATE enum

Operating mode before the transition.

PSTATE_PACTIVE_OPMODE_TRANSITION_DENIED

Trace the denied pactive and Opmode mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Operating mode after the transition.

OLD_STATE enum

Operating mode before the transition.

REASON string

Reason for the transition being denied.

PSTATE_PACTIVE_POWER_TRANSITION

Trace pactive and power mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Power mode after the transition.

OLD_STATE enum

Power mode before the transition.

PSTATE_PACTIVE_POWER_TRANSITION_DENIED

Trace the denied pactive and power mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Power mode after the transition.

OLD_STATE enum

Power mode before the transition.

REASON string

Reason for the transition being denied.

PSTATE_PACTIVE_THREAD_OPMODE_TRANSITION

Trace pactive and Opmode mode transitions for threads on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Operating mode after the transition.

OLD_STATE enum

Operating mode before the transition.

QUANTUM_START

Called at the start of every quantum.

RANGE_PRELOAD_REQUEST

Preload requests from RPRFM. Fields:

COUNT unsigned int

Number of blocks of data to access.

LENGTH unsigned int

Number of contiguous bytes to be accessed in each block.

OP unsigned int

Range prefetch operation .

REUSE unsigned int

Maximum number of bytes to be accessed before next RPRFM .

STRIDE signed int

Stride to advance after accessing LENGTH bytes.

VADDR unsigned int

Base virtual address of range prefetch request.

RUN_STATE

Run state transition. Fields:

INST_COUNT unsigned int

Ticks count at point of transition.

NEW enum

New run state.

OLD enum

Old run state.

SBP_SAMPLE_COLLISION

DISPLAY Statistical profiling samples collided .

SBP_SAMPLE_DISCARDED

DISPLAY Operation discarded by statistical profiling unit when configured in discard mode.
Fields:

BUFFER_POINTER unsigned int

Value of buffer pointer.

DATA unsigned int

Data discarded.

SIZE unsigned int

Size of sample, in bytes.

SBP_SAMPLE_END

DISPLAY Statistical profiling of an operation ended.

SBP_SAMPLE_POP

DISPLAY Operation which might be sampled by statistical profiling.

SBP_SAMPLE_RECORDED

DISPLAY Operation sampled by statistical profiling has been written to the buffer at
%{BUFFER_ADDRESS} with size %{SIZE} and data %{DATA}. Fields:

BUFFER_ADDRESS unsigned int

Virtual Address of start of buffer write.

DATA unsigned int

Data written to the buffer.

SIZE unsigned int

Size of sample written to buffer, in bytes.

SBP_SAMPLE_START

DISPLAY Statistical profiling of an operation started.

SBP_SAMPLE_TAKEN

DISPLAY Operation sampled by statistical profiling after filtering.

SEMIHOSTING_CALL

Call of a semihost function occurred. Fields:

PC unsigned int

The program counter after the semihosting call.

SEMIHOSTING_PRECALL

About to call semihost. Fields:

EL2 bool

Use EL2 translation regime.

NS bool

Is Non-Secure.

REG_WIDTH unsigned int

The current register width in bytes.

SIGNAL

External signal state change. Fields:

SIGNAL enum

Signal that changed.

STATE bool

Signal asserted state.

SOFTWARE_STEP

Return the Debug Software Step state. Fields:

STATE enum

Software Step state.

SPSR

SPSR change. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

MODE enum

Which of the banked SPSR registers is written.

OLD_VALUE unsigned int

The old SPSR value.

VALUE unsigned int

The new SPSR value.

START_COMPILE

Compilation started. Fields:

VADDR unsigned int

Instruction where compilation begins.

STG_MULTIPLE_MICRO_OPS_ADDRESS_RANGE

Address range for MTE stores with multiple uops. Fields:

ADDR_END unsigned int

Final address.

ADDR_START unsigned int

Starting address.

STORE_MULTIPLE_REGS_ADDRESS_RANGE

Address range for stores with multiple registers. Fields:

ADDR_END unsigned int

Final address.

ADDR_START unsigned int

Starting address.

SUBOPTIMAL_LDST_RETIRE

suboptimal load/store. Fields:

ACCESS_SIZE unsigned int

Log2 of the access width in bytes used for alignment checking (1: half-word, 2: word ...).

OPTIMAL_ALIGN_SELECTOR unsigned int

Selector to decide the optimal alignment size (1: access width, 2: 4-bytes, 3: 8-bytes, ..., 12: 4-kbytes).

VADDR unsigned int

The virtual address of the access.

SVE_INST_SPEC

SVE operations speculatively executed.

SVE_LD_RETIRE

SVE load instructions architecturally executed.

SVE_LOADS

SVE load accesses. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ACQREL enum

Is this an acquire/release.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DATA unsigned int

The data read or written.

ELEMENT_SIZE unsigned int

Width of each element.

ENDIANNESS enum

Byte ordering of this access.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

LOCK enum

Normal, exclusive or locked access.

MECID unsigned int

Memory Encryption Context ID, if applicable.

MECID2 unsigned int

Memory Encryption Context ID of the second page, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PADDR_TAG unsigned int

If tag checked, the physical address tag for the access.

PAGE_SIZE unsigned int

Page size as $\log_2(\text{size in bytes})$.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK, 2=Exclusive Failed, 3=Unsupported.

SIZE unsigned int

Width of the access in bytes. Only required if DATA is not traced.

TAG_CHECKED bool

Is this access tag checked.

TAG_CHECKED2 bool

Is the second page access tag checked.

TRANS bool

Is this a translated access.

VADDR unsigned int

The virtual address of the access.

SVE_MEM_PREFETCH

Scalable Vector Extension memory prefetch. Fields:

CACHE_LEVEL enum

The preferred cache level.

HINT enum

What the memory is expected to be used for.

SIZE unsigned int

The prefetch size in bytes.

TEMPORAL bool

Whether the data is temporal.

VADDR unsigned int

The virtual address of the prefetch.

SVE_REG_READ

Scalable Vector Extension register read. Fields:

ID enum

The register identifier.

SM bool

Whether the PE is in Streaming Mode.

VALUE unsigned int

The read data.

SVE_REG_UPDATE

Scalable Vector Extension register update. Fields:

ID enum

The register identifier.

SM bool

Whether the PE is in Streaming Mode.

VALUE unsigned int

The written data.

SVE_REG_WRITE

Scalable Vector Extension register write. Fields:

ID enum

The register identifier.

SM bool

Whether the PE is in Streaming Mode.

VALUE unsigned int

The written data.

SVE_STORES

SVE store accesses. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ACQREL enum

Is this an acquire/release.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DATA unsigned int

The data read or written.

ELEMENT_SIZE unsigned int

Width of each element.

ENDIANNESS enum

Byte ordering of this access.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

LOCK enum

Normal, exclusive or locked access.

MECID unsigned int

Memory Encryption Context ID, if applicable.

MECID2 unsigned int

Memory Encryption Context ID of the second page, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PADDR_TAG unsigned int

If tag checked, the physical address tag for the access.

PAGE_SIZE unsigned int

Page size as $\log_2(\text{size in bytes})$.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK, 2=Exclusive Failed, 3=Unsupported.

SIZE unsigned int

Width of the access in bytes. Only required if DATA is not traced.

TAG_CHECKED bool

Is this access tag checked.

TAG_CHECKED2 bool

Is the second page access tag checked.

TRANS bool

Is this a translated access.

VADDR unsigned int

The virtual address of the access.

SVE_ST_RETIRED

SVE store instructions architecturally executed.

SVE_UNALIGNED_LDST_RETIRED

SVE load or store instructions architecturally executed that would cause alignment fault if A bit is set.

SVE_Z_PRED_MEM_ACCESS_INFO

Information about SVE/SME predicated accesses to Z registers (non Gather/Scatter). Fields:

MEM_ELEM_SIZE unsigned int

Width of elements accessed in memory (in bytes).

RnW bool

Whether this is a Read not a Write.

VA unsigned int

Virtual base address computed for accesses.

SYNC

Called for every synchronization, that is every quantum break. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The instruction count of this CPU.

LOCAL_QUANTUM unsigned int

The local quantum of this CPU.

LOCAL_TIME unsigned int

The local time of this CPU.

SYNC_REASON enum

Reason for sync.

SYSCALL

System call instruction executed. Fields:

IMM unsigned int

Immediate value of the system call instruction.

TYPE enum

System call type.

VADDR unsigned int

Instruction that caused the system call.

SYSREG_READ128

System Coprocessor 128 bit register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

REGNUM enum

Internal register number.

REGNUM_PHYS enum

Internal register number physically accessed (e.g. in presense of VHE redirection).

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_READ64

System Coprocessor register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REGNUM enum

Internal register number.

REGNUM_OPERAND enum

Register number corresponding to the instruction operand.

REGNUM_PHYS enum

Internal register number physically accessed (e.g. in presense of VHE redirection).

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_UPDATE128

Triggers when the system updates a register. Fields:

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG enum

Register number.

UNKNOWN unsigned int

Bits of the register which became unknown.

VALUE unsigned int

Value written to the register.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_UPDATE32

Triggers when the system updates a register. Fields:

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG enum

Register number.

UNKNOWN unsigned int

Bits of the register which became unknown.

VALUE unsigned int

Value written to the register.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_UPDATE64

Triggers when the system updates a register. Fields:

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG enum

Register number.

UNKNOWN unsigned int

Bits of the register which became unknown.

VALUE unsigned int

Value written to the register.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_WRITE128

System Coprocessor 128 bit register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

REGNUM enum

Internal register number.

REGNUM_PHYS enum

Internal register number physically accessed (e.g. in presense of VHE redirection).

UNDEF bool

The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_WRITE64

System Coprocessor register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REGNUM enum

Internal register number.

REGNUM_OPERAND enum

Register number corresponding to the instruction operand.

REGNUM_PHYS enum

Internal register number physically accessed (e.g. in presense of VHE redirection).

UNDEF bool

The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

TRBU_BUFFER_WRAP

DISPLAY Trace buffer current write pointer wrapped.

TRBU_TRIGGER_EVENT

DISPLAY Trace buffer Trigger Event.

TRBU_WRITE

DISPLAY PETU trace data has been written to the TRBU memory buffer at %{BUFFER_ADDRESS} with size %{SIZE} and data %{DATA}. Fields:

BUFFER_ADDRESS unsigned int

Virtual Address of start of buffer write.

DATA unsigned int

Data written to the buffer.

SIZE unsigned int

Size of sample written to buffer, in bytes.

UNALIGNED_LDST_RETIRED

Processor unaligned load/store. Fields:

ACCESS_SIZE unsigned int

Log2 of the access width in bytes used for alignment checking (1=>half-word, 2->word ...).

NUMBER_OF_BEATS unsigned int

Number of accesses (beats) in this burst (Total size in bytes is ACCESS_SIZE * NUMBER_OF_BEATS).

VADDR unsigned int

The virtual address of the access.

VFP_D_REGS

VFP/NEON D 64 bit register write. Fields:

ALIASING enum

Type of register aliasing used.

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a write to the corresponding bit in the register. For example, 0x1 would indicate a write to VALUE[0].

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

VFP_D_REGS_READ

VFP/NEON D 64 bit register read. Fields:

ID unsigned int

The register number.

VALUE unsigned int

The current value of the register.

VFP_Q_REGS

VFP/NEON Q 128 bit register write. Fields:

ALIASING enum

Type of register aliasing used.

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a write to the corresponding byte in the register. For example, 0x1 would indicate a write to VALUE[7:0].

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

VFP_Q_REGS_READ

VFP/NEON Q 128 bit register read. Fields:

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a read to the corresponding byte in the register. For example, 0x1 would indicate a read to VALUE[7:0].

VALUE unsigned int

The current value of the register.

VFP_SYS_REGS

Writes to the VFP/NEON units system registers. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ID enum

Which VFP system register is written.

MASK unsigned int

The mask indicating updated cumulative bits.

OLD_VALUE unsigned int

The register's old value overwritten.

VALUE unsigned int

The new value written to the VFP system register.

VFP_S_REGS

VFP/NEON S 32 bit register write. Fields:

ALIASING enum

Type of register aliasing used.

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a write to the corresponding bit in the register. For example, 0x1 would indicate a write to VALUE[0].

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

VFP_S_REGS_READ

VFP/NEON S 32 bit register read. Fields:

ID unsigned int

The register number.

VALUE unsigned int

The current value of the register.

VIRTUAL_TAG_ALLOCATED

Virtual tag information when allocated. Fields:

TAG_VADDR unsigned int

Virtual tag virtual address.

VADDR unsigned int

Data Virtual address.

VIRTUAL_TAG unsigned int

Virtual tag value.

WAYPOINT

Signals end of basic block execution. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ISET enum

Origin instruction set.

IS_COND bool

Indicates if this is a conditional waypoint.

PC unsigned int

Origin address (or 0 if unavailable).

TAKEN bool

Indicates if this waypoint was taken.

TARGET unsigned int

Destination address.

TARGET_ISET enum

Destination instruction set.

WFE_END

WFE ended. Fields:

INST_COUNT unsigned int

Ticks count when leaving WFE.

WFE_EVENT_REGISTER

WFE event register status: set/clear, reason. Fields:

INST_COUNT unsigned int

Ticks count.

REASON enum

Reason for set/clear. Only REASON==0 (Cleared by WFE) clears the bit, all other reasons set it.

WFE_IGNORED

WFE ignored. Fields:

EVENT bool

This WFE was ignored because the event register was set.

INST_COUNT unsigned int

Ticks count when ignoring WFE.

TRAPPED bool

This WFE was trapped.

WFE_START

WFE entered. Fields:

INST_COUNT **unsigned int**

Ticks count when entering WFE.

WFI_END

WFI ended. Fields:

INST_COUNT **unsigned int**

Ticks count when leaving WFI.

WFI_IGNORED

WFI ignored. Fields:

DISABLED **bool**

This WFI was ignored because WFI is disabled.

INST_COUNT **unsigned int**

Ticks count when ignoring WFI.

REASON **enum**

specifies reason why WFI trace was ignored.

TRAPPED **bool**

This WFI was trapped.

WFI_START

WFI entered. Fields:

INST_COUNT **unsigned int**

Ticks count when entering WFI.

WFI_WAKEUP

WFI wakeup. Fields:

INST_COUNT **unsigned int**

Ticks count when WFI wakeup occurred.

REASON **enum**

Reason for wakeup.

2.61 ARM_Cortex-X925

This section describes the trace sources.

AA64_ASE_SVE_REGS

Modification of AArch64 Advanced SIMD or SVE Vector Registers (via SIMD or Scalar operations), SVE Predicate Registers and ZT0. Fields:

ID enum

The register identifier.

MASK unsigned int

Bitmask of the register to signal the modified bits in the VALUE field.

SM bool

Whether the PE is in Streaming Mode.

VALUE unsigned int

Value of AArch64 vector register.

ASYNC_MEMORY_FAULT

Context ID Register write. Fields:

FAULT unsigned int

Fault status present in ESR.ISS encoding for Serror in ARCH64 and present in DFSR encoding for external abort format in ARCH32.

PADDR unsigned int

Physical Address (or 0 if unavailable).

VADDR signed int

Virtual Address (or 0 if unavailable).

ATOMIC_END_ACCESS

Bus trace access for atomics. Fields:

ACCESS_FAIL bool

Memory access failed.

ADDR unsigned int

The virtual address of the access.

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

COMPARE_VALUE unsigned int

Compare value for CAS.

LOAD_VALUE unsigned int

Loaded values.

NSDESC unsigned int

The security state of the access.

OPERAND_VALUE unsigned int

Operation's operand.

OPERATION enum

Operation type.

PADDR unsigned int

The physical address of the access.

PRIV bool

Is this a privileged access?.

ATOMIC_START_ACCESS

Bus trace access for atomics. Fields:

ADDR unsigned int

The virtual address of the access.

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

COMPARE_VALUE unsigned int

Compare value for CAS.

NSDESC unsigned int

The security state of the access.

OPERAND_VALUE unsigned int

Operation's operand.

OPERATION enum

Operation type.

PADDR unsigned int

The physical address of the access.

PRIV bool

Is this a privileged access?.

ArchMsg.Error.exit_code

DISPLAY exit code: %{EXITCODE} component kind: %{KIND} reason: %{REASON}. Fields:

EXITCODE signed int

Exit code to be returned.

KIND string

Component kind that invoked the exit code trace.

REASON string

Reason why the exit code trace was invoked (optional).

ArchMsg.Warning.IgnoredWarmResetRequest

Event fires when software attempts to request warm reset through write to deprecated DBGPRCR.CWRR.

ArchMsg.Warning.RAS_ErrselrUnpredictableState

DISPLAY ERRSELR is in unpredictable state due to value greater than number_of_error_records written to it.

ArchMsg.Warning.RAS_UnpredictableStateWrite

DISPLAY %{REG_NAME} write ignored due to unpredictable ERRSELR state. Fields:

REG_NAME string

Name of register being written.

ArchMsg.Warning.ReportedRevisionRegisterRead

DISPLAY Reported revision and patch do not match that of the cpu. Fields:

ORIGINAL_VALUE unsigned int

The value the register would have based on the modelled revision.

REGISTER string

The name of the register that has ben read.

REPORTED_VALUE unsigned int

The value the register has been overridden to.

ArchMsg.Warning.Unpredictably Indexed PM Event Register

TAGS **UNPREDICTABLE** DISPLAY Event %{IsDirect:(Selector value|Register)} %{SEL} is >= %{N}. Fields:

IsDirect unsigned int

Direct Access rather than indirect.

N unsigned int

Number of Accessible Registers.

SEL unsigned int

Selector.

ArchMsg.Warning.cache_contents_unknown

A cache has been activated without being invalidated and may contain **UNKNOWN** content. DISPLAY %{SIDE:(D|I)}-Cache was enabled in %{REGIME:(SECURE|NON_SECURE|HYP)} regime but was not invalidated since power-on: cache lines could contain **UNKNOWN** content %{SIDE:(Note: D-side caches must invalidate SECURE lines even if the cache is only used by NON_SECURE code in order to ensure that unexpected dirty lines tagged SECURE are not naturally evicted)}. Fields:

REGIME unsigned int

Current translation regime.

SIDE unsigned int

0 d-side cache or unified cache, 1 i-side.

ArchMsg.Warning.dcimvac_matches_watchpoint

TAGS **UNPREDICTABLE** DISPLAY Watchpoints matching an AArch32 DCIMVA are implementation defined.

ArchMsg.Warning.decode_fieldmismatch

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with different fields %{FIELD1} and %{FIELD2}. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_fracbitsnegative

TAGS **UNPREDICTABLE** DISPLAY VCVT Instruction is unpredictable with a negative number of fraction bits.

ArchMsg.Warning.decode_initblock

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable in an IT block.

ArchMsg.Warning.decode_initblocknotlast

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable in an IT block when not the last.

ArchMsg.Warning.decode_invalid_condition

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable when the condition is not AL.

ArchMsg.Warning.decode_invalidfieldcombination

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with the values in %{FIELD1} and %{FIELD2}. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_invalidvalue

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with the value in %{FIELD}. Fields:

FIELD enum

Field name.

ArchMsg.Warning.decode_mem_hint_unallocated

TAGS **UNPREDICTABLE** DISPLAY There is no memory hint allocated to this bit pattern.

ArchMsg.Warning.decode_registermatch

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with %{FIELD1} and %{FIELD2} as the same register. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_registermismatch

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with %{FIELD1} and %{FIELD2} as different registers. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_registeroutofrange

TAGS **UNPREDICTABLE** DISPLAY Floating-point instruction is unpredictable indexing beyond the end of the register bank.

ArchMsg.Warning.decode_sbzsbo

TAGS **UNPREDICTABLE** DISPLAY Reserved bits in the instruction are not canonical.

ArchMsg.Warning.decode_transactiontoobig

TAGS **UNPREDICTABLE** DISPLAY Attempt to perform a memory transaction of more than 128 bytes.

ArchMsg.Warning.decode_unpred_other

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable.

ArchMsg.Warning.decode_unpreduseofpc

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable when %{FIELD} is PC. Fields:

FIELD enum

Field name.

ArchMsg.Warning.decode_unpreduseofr13

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable when %{FIELD} is R13. Fields:

FIELD enum

Field name.

ArchMsg.Warning.decode_unpreduseofr13inreglist

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with R13 in register-list.

ArchMsg.Warning.decode_useofd16orhigher

TAGS **UNPREDICTABLE** DISPLAY Attempt to use register D16 or higher where it is not allowed.

ArchMsg.Warning.decode_writebackandbaseinlist

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with write back when the base register is in the transfer list.

ArchMsg.Warning.decode_zeroregistersinlist

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable when no registers are to be transferred.

ArchMsg.Warning.misaligned_ptrs

DISPLAY TRBPTR_EL1(%{TRBPTR_EL1}) or TRBTRG_EL1(%{TRBTRG_EL1}) are not aligned as per TRBIDR_EL1.Align(%{TRBIDR_EL1}). Fields:

TRBIDR_EL1 unsigned int

TRBIDR_EL1.Align value.

TRBPTR_EL1 unsigned int

Current write pointer.

TRBTRG_EL1 unsigned int

Trigger pointer.

ArchMsg.Warning.out_of_range_ptrs

DISPLAY TRBPTR_EL1(%{TRBPTR_EL1}) is out of range as per TRBBASER_EL1(%{TRBBASER_EL1}) or TRBLIMITR_EL1.LIMIT(%{TRBLIMITR_EL1}). Fields:

TRBBASER_EL1 unsigned int

Base pointer.

TRBLIMITR_EL1 unsigned int

Limit pointer.

TRBPTR_EL1 unsigned int

Current write pointer.

ArchMsg.Warning.recursive_branch

An instruction is performing a branch that targets the same instruction. If this was intended then a WFI might be more effective. DISPLAY recursive branch detected at PC=%{PC}. PRIMARY KEY PC. Fields:

PC unsigned int

Address of instruction causing the branch.

ArchMsg.Warning.recursive_exception

An instruction has generated an exception that targets the same instruction. DISPLAY recursive %{TYPE} exception detected at PC=%{PC} (CPSR=%{CPSR}). PRIMARY KEY PC. Fields:

CPSR unsigned int

Processor state.

PC unsigned int

Address of instruction causing the exception.

TYPE enum

Type of exception.

ArchMsg.Warning.reentrant_vector_catch

TAGS **UNPREDICTABLE** DISPLAY Vector catch debug event on Prefetch or Data abort vector %{ADDR}. Fields:

ADDR unsigned int
vector address.

ArchMsg.Warning.reserved_it_state

TAGS **UNPREDICTABLE** DISPLAY Execution with reserved IT state %{ITSTATE}. Fields:

ITSTATE unsigned int
reserved IT state.

ArchMsg.Warning.tlb_contents_unknown

Translation has been enabled without invalidating the TLB. DISPLAY Translation was enabled in the %{PAS:(Secure|Non-secure|Root|Realm)} EL%{EL:d} translation regime (vmid=%{VMID:x}) without invalidating %{INVALIDITY:(|d-side |i-side |)}entries since power-on: %{INVALIDITY:(|D||)}TLB could contain **UNKNOWN** content. Fields:

EL unsigned int
Exception level of translation regime.

INVALIDITY unsigned int
Which TLB may be invalid (if TLBs are separate).

NON_SECURE bool
Whether the translation regime is non-secure.

PAS unsigned int
Physical address space of translation regime.

VMID unsigned int
Current VMID, if applicable.

ArchMsg.Warning.unknown_DLR_DSPSR

Exiting debug state with unknown DLR or DSPSR.

ArchMsg.Warning.unknown_ELR_SPSR

Returning from debug exception with unknown ELR or SPSR.

ArchMsg.Warning.unpred_FAR_on_LRCPC3_non_atomic_mem_access

TAGS **UNPREDICTABLE** Unpredictable FAR/FSTATUS on LRCPC3 pair instructions resulting in two separate memory accessesDISPLAY Unpredictable FAR on LRCPC3 non atomic memory access at %{ADDR1} with FSTATUS %{FSTATUS1} or %{ADDR2} with FSTATUS %{FSTATUS2}. Fields:

ADDR1 unsigned int
Possible FAR address 1.

ADDR2 unsigned int
Possible FAR address 2.

FSTATUS1 unsigned int
Possible fault status value 1.

FSTATUS2 unsigned int
Possible fault status value 2.

ArchMsg.Warning.unpred_async_tag_check_fail_on_sve_speculative_load_do_update_ffr

Unpredictable set to FALSE of the FFR element associated to SVE speculative load with an async tag check fail TAGS **UNPREDICTABLE**.

ArchMsg.Warning.unpredictable_a32_breakpoint

TAGS **UNPREDICTABLE** DISPLAY Unpredictable breakpoint on a A32 instruction at %{ADDR1} Offset %{OFFSET} BAS %{BAS} for a %{IS_ADDRESS_MISMATCH_BREAKPOINT:(Mismatch|Match)} Breakpoint number %{N:d} Breakpoint address %{ADDR2}. Fields:

ADDR1 unsigned int

Address.

ADDR2 unsigned int

Breakpoint address.

BAS unsigned int

BAS field.

IS_ADDRESS_MISMATCH_BREAKPOINT bool

Is a Mismatch breakpoint, not a Match breakpoint.

N unsigned int

Breakpoint number.

OFFSET unsigned int

Always has a value of zero.

ArchMsg.Warning.unpredictable_banked_register_access

TAGS **UNPREDICTABLE** DISPLAY Access to banked register (R=%{R:d} SYSm=%{SYSM:2x}) is unpredictable. Fields:

R unsigned int

R bit.

SYSM unsigned int

SYSm.

ArchMsg.Warning.unpredictable_ccfail_undef

TAGS **UNPREDICTABLE** DISPLAY An instruction %{OPCODE} at VA=%{VADDR} failed its condition codes check but would otherwise have caused a non-data-dependent trap or undefined exception. Behaviour is implementation-defined choice of exception or nop. Fields:

OPCODE unsigned int

opcode of the instruction.

VADDR unsigned int

Virtual address of the instruction being executed.

ArchMsg.Warning.unpredictable_ici_bits

TAGS **UNPREDICTABLE** DISPLAY ICI bits do not apply or are architecturally unpredictable for instruction.

ArchMsg.Warning.unpredictable_t32_breakpoint

TAGS **UNPREDICTABLE** DISPLAY Unpredictable breakpoint on a T32 instruction at %{ADDR1} Offset %{OFFSET} BAS %{BAS} for a %{IS_ADDRESS_MISMATCH_BREAKPOINT:(Mismatch|Match)} Breakpoint number %{N:d} Breakpoint address %{ADDR2}. Fields:

ADDR1 unsigned int

Address.

ADDR2 unsigned int

Breakpoint address.

BAS unsigned int

BAS field.

IS_ADDRESS_MISMATCH_BREAKPOINT bool

Is a Mismatch breakpoint, not a Match breakpoint.

N unsigned int

Breakpoint number.

OFFSET unsigned int

Indicates which halfword of a 32-bit Thumb instruction the breakpoint is set on. 0 for the first halfword or if the instruction is 16 bits wide, or 1 for the second halfword.

ArchMsg.Warning.unpredictable_t32_vector_catch

TAGS **UNPREDICTABLE** DISPLAY Unpredictable Vector catch exception on a T32 instruction at %{ADDR} Offset %{OFFSET}. Fields:

ADDR unsigned int

Address.

OFFSET unsigned int

Indicates which halfword of a 32-bit Thumb instruction the vector catch exception is set on. 0 for the first halfword or if the instruction is 16 bits wide, or 1 for the second halfword.

ArchMsg.Warning.unpredictable_watchpoint_far

TAGS **UNPREDICTABLE** DISPLAY The value for FAR/EDWAR for hitting this watchpoint may be anywhere in the range %{LOWER_BOUND} to %{UPPER_BOUND}. Fields:

LOWER_BOUND unsigned int

the lowest address accessed by the instruction that triggered the watchpoint.

UPPER_BOUND unsigned int

the highest watchpointed address accessed by that instruction.

ArchMsg.Warning.unpredictable_watchpoint_far_isscmp

DISPLAY The value for FAR/EDWAR for hitting this watchpoint may be anywhere in the range %{LOWER_BOUND} to %{UPPER_BOUND}. Fields:

LOWER_BOUND unsigned int

the lowest address accessed by the instruction that triggered the watchpoint.

UPPER_BOUND unsigned int

the highest watchpointed address accessed by that instruction.

ArchMsg.Warning.warning_AdvSIMDEExpandImmUnexpectedZero

TAGS **UNPREDICTABLE** DISPLAY AdvSIMDEExpandImm may treat this immediate value as **UNPREDICTABLE**.

ArchMsg.Warning.warning_ConditionalSMC

TAGS **UNPREDICTABLE** DISPLAY SMC instruction has **UNPREDICTABLE** effects when conditional and when combined with traps.

ArchMsg.Warning.warning_TLBID_out_of_range

TLBID out of range in TLBI according to the configuration of NIS, NVIS, NOS and NVOS.

Fields:

NIS unsigned int

NIS.

NOS unsigned int

NOS.

NVIS unsigned int

NVIS.

NVOS unsigned int

NVOS.

TLBID unsigned int

The TLBID value used in the TLBI instruction.

ArchMsg.Warning.warning_TTL_or_TG_mismatches_current_translation_granule

TTL or TG mismatches current translation granule in TLBI VA/RVA/IPA/RIPA. Fields:

CURRENT_TG unsigned int

Current TG as indicated by TCR_ELx or VTCR_EL2.

TLBI_TG unsigned int

TG (or TTL[3:2]) used in TLBI instruction.

ArchMsg.Warning.warning_access_crosses_page_boundary

TAGS **UNPREDICTABLE** DISPLAY Access crosses page boundary.

ArchMsg.Warning.warning_access_crosses_page_boundary_has_fault

TAGS **UNPREDICTABLE** DISPLAY Access that crosses page boundary has fault(s).

ArchMsg.Warning.warning_access_crosses_page_boundary_has_fault_on_both_pages

TAGS **UNPREDICTABLE** DISPLAY Access that crosses page boundary has a fault on each page. Fields:

ADDR1 unsigned int

FAR address on first page.

ADDR2 unsigned int

FAR address on second page.

FSTATUS1 unsigned int

Fault status on first page.

FSTATUS2 unsigned int

Fault status on second page.

ArchMsg.Warning.warning_access_crosses_page_boundary_spanning_different_memory

TAGS **UNPREDICTABLE** DISPLAY Access at address %{ADDR} crosses page from MEM_TYPE %{MEMTYPE_PAGE1} to %{MEMTYPE_PAGE2}. Fields:

ADDR unsigned int

address of access crossing page.

MEMTYPE_PAGE1 signed int

First Page's memory type (0-Normal, 1-Non-Normal).

MEMTYPE_PAGE2 signed int

Second Page's memory type (0-Normal, 1-Non-Normal).

ArchMsg.Warning.warning_access_wraps_around_memory

TAGS **UNPREDICTABLE** DISPLAY Access wraps around memory in %{WIDTH:d}bit mode. Fields:

WIDTH unsigned int

Address width.

ArchMsg.Warning.warning_bcr_linking_status

TAGS **UNPREDICTABLE** DISPLAY DBGBCR.{BT2,BT} combination gives unpredictable behaviour.

ArchMsg.Warning.warning_bcr_mask_reserved

TAGS **UNPREDICTABLE** DISPLAY DBGBCR_ELn.MASK is set to a reserved value.

ArchMsg.Warning.warning_bvr_masked_bit_not_zero

TAGS **UNPREDICTABLE** DISPLAY DBGBVRn_EL1.VA contains a bit masked by DBGBCRn_EL1.MASK that is not zero.

ArchMsg.Warning.warning_ccsidr_unimplemented_level

TAGS **UNPREDICTABLE** DISPLAY CCSIDR read while CSSELR %{CSSELR:x} points to an unimplemented cache level. Fields:

CSSELR unsigned int

current effective value of CSSELR.

ArchMsg.Warning.warning_change_to_ns_when_tge_set

TAGS **UNPREDICTABLE** DISPLAY Attempting to change NS to 1 when HCR.TGE = 1.

ArchMsg.Warning.warning_contiguous_bit_check_abort

TAGS **UNPREDICTABLE** DISPLAY An abort occurred whilst attempting to check TLB entry is contiguous at %{ENTRY_ADDR}. Fields:

ENTRY_ADDR unsigned int

address of conflicting TLB entry.

ArchMsg.Warning.warning_contiguous_bit_error

TAGS **UNPREDICTABLE** DISPLAY TLB entry at %{ENTRY_ADDR} was not contiguous with entry at %{CONTIG_ADDR} Contents are %{ENTRY_DATA} but expected %{CONTIG_DATA}. Fields:

CONTIG_ADDR unsigned int

address of the TLB first read, with which this entry is expected to be contiguous.

CONTIG_DATA unsigned int

expected contents based on the entry at CONTIG_ADDR.

ENTRY_ADDR unsigned int

address of conflicting TLB entry.

ENTRY_DATA unsigned int

contents of conflicting TLB entry.

ArchMsg.Warning.warning_contiguous_bit_error_D128

TAGS **UNPREDICTABLE** DISPLAY TLB entry at %{ENTRY_ADDR} was not contiguous with entry at %{CONTIG_ADDR} Contents are %{ENTRY_DATA_HI} %{ENTRY_DATA_LO} but expected %{CONTIG_DATA}. Fields:

CONTIG_ADDR unsigned int

address of the TLB first read, with which this entry is expected to be contiguous.

CONTIG_DATA unsigned int

expected contents based on the entry at CONTIG_ADDR.

ENTRY_ADDR unsigned int

address of conflicting TLB entry.

ENTRY_DATA_HI unsigned int

Higher bit contents of conflicting TLB entry.

ENTRY_DATA_LO unsigned int

Lower bit contents of conflicting TLB entry.

ArchMsg.Warning.warning_cp10_cp11_mismatch

TAGS **UNPREDICTABLE** DISPLAY Unpredictable configuration of CP10 and CP11 controls at EL %{EL:d}. Fields:

EL unsigned int

EL owning the control.

ArchMsg.Warning.warning_cp10_cp11_reserved_value

TAGS **UNPREDICTABLE** DISPLAY Unpredictable configuration of CPACR CP10 ({CP10}) or CP11 ({CP11}). Fields:

CP10 unsigned int

CP10 access.

CP11 unsigned int

CP11 access.

ArchMsg.Warning.warning_csselr_level_out_of_range

TAGS **UNPREDICTABLE** DISPLAY CSSELR.Level out of range, written value {LEVEL} with only {IMPLEMENTED:d} levels implemented. Fields:

IMPLEMENTED unsigned int

number of cache levels implemented.

LEVEL unsigned int

written value of CSSELR.Level.

TYPE enum

The type of cache selected by CSSELR.InD and TnD.

ArchMsg.Warning.warning_debug_flow_control_bits_not_obeyed

TAGS **UNPREDICTABLE** DISPLAY Flow control bits for {REGISTER} were not obeyed, causing {ERROR}. Fields:

ERROR enum

type of error resulting.

REGISTER enum

register under consideration.

ArchMsg.Warning.warning_debug_register_access_during_reset

TAGS **UNPREDICTABLE** DISPLAY Unpredictable {IS_WRITE:(read|write)} access to debug register offset {OFFSET} during reset. Fields:

IS_WRITE unsigned int

Write Not Read.

OFFSET unsigned int

Register Offset.

ArchMsg.Warning.warning_decode_cps_inconsistent_fields

TAGS **UNPREDICTABLE** DISPLAY The mode setting fields in the CPS instruction are inconsistent.

ArchMsg.Warning.warning_decode_invalid_state

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable in current exception-level/security state.

ArchMsg.Warning.warning_default_cacheable_mmu_on

TAGS **UNPREDICTABLE** DISPLAY Default cacheable is enabled (HCR.DC = 1) while MMU is enabled.

ArchMsg.Warning.warning_default_cacheable_vmmu_off

TAGS **UNPREDICTABLE** DISPLAY Default cacheable is enabled (HCR.DC = 1) while HCR.VM = 0.

ArchMsg.Warning.warning_deprecated_wvr_bit_2_set

TAGS **UNPREDICTABLE** DISPLAY DBGWVR%{N:d}_EL1=%{WVR:h} has bit 2 set, deprecated in ARMv8. Fields:

N unsigned int

index.

WVR unsigned int

DBGWVR.

ArchMsg.Warning.warning_eret_while_software_step_active_pending

Missing ISB between setting MDSCR_EL1.SS and ERET TAGS **UNPREDICTABLE**.

ArchMsg.Warning.warning_exclusive_to_non_normal

TAGS **UNPREDICTABLE** DISPLAY Exclusive Access to Non-Normal Memory %{ADDR}. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_exclusive_to_non_writeback

TAGS **UNPREDICTABLE** DISPLAY Exclusive Access to Non-Writeback-Cacheable Memory %{ADDR}. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_execute_from_device_memory

TAGS **UNPREDICTABLE** DISPLAY An attempt was made to execute from Device memory.

ArchMsg.Warning.warning_illegal_cpsr_mode

TAGS **UNPREDICTABLE** DISPLAY Writing an illegal or unimplement mode (%{NEW_MODE:h}) to CPSR was unpredictable. Fields:

NEW_MODE unsigned int

New value of CPSR.M.

ArchMsg.Warning.warning_illegal_srs_mode

TAGS **UNPREDICTABLE** DISPLAY Illegal or **UNPREDICTABLE** mode (%{MODE:h}) used for SRS instruction. Fields:

MODE unsigned int

mode for Banked SP.

ArchMsg.Warning.warning_implementation_defined_read_debug_register_in_SCS

Read Access to Debug registers from the processor is IMPLEMENTATION DEFINED TAGS IMP_DEF. Fields:

PPB_OFFSET unsigned int
debug register offset.

ArchMsg.Warning.warning_implementation_defined_sequential_security_transitions_supported

The behavior of sequential instruction fetches that cross from non-secure to secure memory and contain an SG instruction is CONSTRAINED UNPREDICTABLE TAGS UNPREDICTABLE.

ArchMsg.Warning.warning_implementation_defined_stack_limit_check_supported

It is IMPLEMENTATION DEFINED whether stack pointer limit checking is performed for this instruction TAGS IMP_DEF. Fields:

ADDRESS unsigned int
Address of instruction.

OPCODE unsigned int
instruction opcode.

ArchMsg.Warning.warning_implementation_defined_taggednonsharedmemorytype

TAGS IMP_DEF DISPLAY Memory attribute has Tagged, non shared memory type; it is IMPLEMENTATION DEFINED whether the memory location is treated as Tagged or non-Tagged.

ArchMsg.Warning.warning_implementation_defined_write_debug_register_in_SCS

Write Access to Debug registers from the processor is IMPLEMENTATION DEFINED TAGS IMP_DEF. Fields:

DATA unsigned int
data attempted to be written.

PPB_OFFSET unsigned int
debug register offset.

ArchMsg.Warning.warning_invalid_tcr_granule

TAGS UNPREDICTABLE DISPLAY TCR.TG%{TG_ID:(0|1)} bits have been set to a granule size %{{REQUEST:d}}K not implemented - using %{{SUBSTITUTE:d}}K. Fields:

REQUEST signed int
page size requested (or 0 for reserved).

SUBSTITUTE signed int
best guess available page size.

TG_ID bool
bits TG0 or TG1.

ArchMsg.Warning.warning_load_multiple_user_registers_from_user_mode

TAGS UNPREDICTABLE DISPLAY An LDM(user registers) instruction executed from user mode.

ArchMsg.Warning.warning_mte_in_mem_attr_not_supported

TAGS UNPREDICTABLE DISPLAY Memory attribute %{ATTR} has Tagged memory type but Armv8.5-MTE is not implemented. Fields:

ATTR unsigned int

Memory attribute.

ArchMsg.Warning.warning_reserved_breakpoint_state_match

TAGS UNPREDICTABLE DISPLAY %{IS_BREAKPOINT:(Watchpoint|Breakpoint)} programmed with a reserved combination of HMC, SSC and %{IS_BREAKPOINT:(PAC|PMC)}. Fields:

IS_BREAKPOINT unsigned int

Is a breakpoint, not a watchpoint.

ArchMsg.Warning.warning_shareability

TAGS UNPREDICTABLE DISPLAY Unpredictable: combination of the 1st and 2nd stages of translation is Normal Inner Non-Cacheable, Outer Non-Cacheable.

ArchMsg.Warning.warning_software_step_set_while_enabled

TAGS UNPREDICTABLE DISPLAY MDSCR_EL1.SS set to 1 while software step debug exceptions are enabled.

ArchMsg.Warning.warning_thumb_instruction_wraps_around_memory

TAGS UNPREDICTABLE DISPLAY Thumb instruction wraps around memory.

ArchMsg.Warning.warning_ttbr_sbz_bits_are_not_zero

TAGS UNPREDICTABLE DISPLAY %{TTBR}[%{MSB:d};0] should be zero, but are not. Fields:

MSB unsigned int

x-1.

TTBR enum

which TTBR register.

ArchMsg.Warning.warning_unaligned_address_dbgdtrrx_write

TAGS UNPREDICTABLE DISPLAY Unaligned address %{ADDR} in X0/R0 when DBGDTRRX written in memory access mode. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unaligned_address_dbgdtrtx_read

TAGS UNPREDICTABLE DISPLAY Unaligned address %{ADDR} in X0/R0 when DBGDTRTX read in memory access mode. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unaligned_to_device

TAGS UNPREDICTABLE DISPLAY Unaligned Access to Device Memory %{ADDR}. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unaligned_to_strongly_ordered

TAGS **UNPREDICTABLE** DISPLAY Unaligned Access to Strongly Ordered Memory %{ADDR}.

Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unknown_sau_rnr

TAGS **UNPREDICTABLE** DISPLAY SAU_RNR was set to an unsupported value.

ArchMsg.Warning.warning_unpredictable_AIRCR_PRIS_and_BFHFNMINs

TAGS **UNPREDICTABLE** DISPLAY The effect of setting both AIRCR.BFHFNMINs and AIRCR.PRIS to 1 is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_AIRCR_VECTCLRACTIVE_when_not_in_debug

TAGS **UNPREDICTABLE** DISPLAY The effect of writing a 1 to AIRCR.VECTCLRACTIVE if the processor is not halted in Debug state is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_AIRCR_VECTRESET_and_SYSRESETREQ

TAGS **UNPREDICTABLE** DISPLAY When the processor is halted in Debug state, if a write to the register writes a 1 to both VECTRESET and SYSRESETREQ, the behavior is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_AIRCR_VECTRESET_when_not_in_debug

TAGS **UNPREDICTABLE** DISPLAY The effect of writing a 1 to AIRCR.VECTRESET if the processor is not halted in Debug state is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_AIRCR_incorrect_VKEY

TAGS **UNPREDICTABLE** DISPLAY The value 0x05FA must be written to AIRCR.VKEY, otherwise the register write is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_EXC_RETURN_Reserved_Bit

EXC_RETURN[23:7] are reserved with the special condition that all bits should be written as one. Values other than all 1s are **UNPREDICTABLE**. TAGS **UNPREDICTABLE**. Fields:

EXC_RETURN unsigned int

Special PC value.

SBO_MASK unsigned int

Which bits Should Be One to not UNPRED.

ArchMsg.Warning.warning_unpredictable_PMEVTYPEPERn_ELO_VS_reserved_value

TAGS **UNPREDICTABLE** DISPLAY Unpredictable reserved value of PMEVTYPER<n>_ELO.VS.

ArchMsg.Warning.warning_unpredictable_change_to_DEMCR_MON_STEP_at_insufficient_priority

TAGS **UNPREDICTABLE** DISPLAY The effect of changing DEMCR.MON_STEP at an execution priority that is lower than the priority of the DebugMonitor exception is **CONSTRAINED UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_change_to_priority_of_active_exception

TAGS UNPREDICTABLE DISPLAY Changing the priority of an active exception is UNPREDICTABLE.

ArchMsg.Warning.warning_unpredictable_clear_lspact

TAGS UNPREDICTABLE DISPLAY UNPREDICTABLE state on context switch. Fields:

DRVEXC_TAKEN unsigned int

Derived exception taken, not pended.

REACHED_LAST_STORE unsigned int

Only the last faulted.

ArchMsg.Warning.warning_unpredictable_dbg_exit_unaligned_dlr

TAGS UNPREDICTABLE DISPLAY Debug exit to AArch32 with DLR[0] set to 1. PC[0] can be set to 0 or to DLR[0].

ArchMsg.Warning.warning_unpredictable_exception_catch

TAGS UNPREDICTABLE DISPLAY Generation of an exception catch event is unpredictable.

ArchMsg.Warning.warning_unpredictable_exception_return_inconsistent_state

TAGS UNPREDICTABLE DISPLAY state on exception return is unpredictable.

ArchMsg.Warning.warning_unpredictable_exception_return_instruction

TAGS UNPREDICTABLE DISPLAY exception return instruction is unpredictable.

ArchMsg.Warning.warning_unpredictable_in_debug_state

TAGS UNPREDICTABLE DISPLAY Instruction is UNPREDICTABLE when executed in debug state.

ArchMsg.Warning.warning_unpredictable_mair_encoding

TAGS UNPREDICTABLE DISPLAY Memory attribute %{ATTR} has invalid MAIR encoding; this may lead to unpredictable results. Fields:

ATTR unsigned int

Memory attribute.

ArchMsg.Warning.warning_unpredictable_pmu_counter_access

TAGS UNPREDICTABLE DISPLAY Access to PMU counters from non-secure EL0 or EL1 is UNPREDICTABLE with MDCR_EL2.HPMN set to 0.

ArchMsg.Warning.warning_unpredictable_prioritization_breakpoint_match_vector_catch

TAGS UNPREDICTABLE DISPLAY Two events with the same priority occurred at the same instruction: Address Matching Vector Catch debug event, and Breakpoint debug event. It is constrained unpredictable which is taken.

ArchMsg.Warning.warning_unpredictable_stack_selection

TAGS UNPREDICTABLE DISPLAY UNPREDICTABLE state on context switch.

ArchMsg.Warning.warning_unpredictable_tsize_out_of_range

TAGS UNPREDICTABLE DISPLAY %{TCR}.T%{TTBR:d}SZ (%{TSIZE:d} bits) is out of range. Must be between 25 and %{TSIZE_MAX:d}. Fields:

TCR enum

Which TCR.

TSize signed int

Value of TSize.

TSize_MAX signed int

The maximum allowed value.

TTBR unsigned int

Which TTBR region.

ArchMsg.Warning.warning_unpredictable_unaligned_pc_as_base_register

TAGS **UNPREDICTABLE** DISPLAY The PC value must be word-aligned, otherwise the behavior of the instruction is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_unaligned_pop_stack

TAGS **UNPREDICTABLE** DISPLAY **CONSTRAINED UNPREDICTABLE** unaligned stack frame `%{FRAME_ADDR:x}` in exception-return. Fields:

FRAME_ADDR unsigned int

unaligned stack frame address.

ArchMsg.Warning.warning_unpredictable_unaligned_to_non_writeback

TAGS **UNPREDICTABLE** DISPLAY Unaligned Access to Non-Writeback-Cacheable Memory.

ArchMsg.Warning.warning_unpredictable_vmsa_memattrib

TAGS **UNPREDICTABLE** DISPLAY Unpredictable vmsa memory attribute with texcb value programmed to `%{TEXCB}` when tex remap is `%{TEX_REMAP}`. Fields:

TEXCB unsigned int

texcb value.

TEX_REMAP bool

use tex remap.

ArchMsg.Warning.warning_unpredictable_vtcr_t0sz_sl0

TAGS **UNPREDICTABLE** DISPLAY The combination of VTCR.TOSZ=`%{TOSZ:d}` and VTCR.SL0=`%{SL0:d}` is unpredictable. Fields:

SL0 signed int

VTCR.SL0, starting level for stage 2 translation table walks.

TOSZ signed int

VTCR.TOSZ, required input address range.

ArchMsg.Warning.warning_unpredictable_write_DHCSR

TAGS **UNPREDICTABLE** DISPLAY **UNPREDICTABLE** set of bit-changes in DHCSR. Fields:

NEW unsigned int

data attempted to be written.

OLD unsigned int

previous value.

ArchMsg.Warning.warning_unpredictable_write_SHCSR

TAGS UNPREDICTABLE DISPLAY UNPREDICTABLE setting a pending bit to 1 for an exception with priority \geq execution priority.

ArchMsg.Warning.warning_unsupported_access_to_memory_mapped_register

TAGS UNPREDICTABLE DISPLAY Access to memory mapped register is unsupported.

ArchMsg.Warning.warning_user_jmcr_access

TAGS UNPREDICTABLE DISPLAY User %{WRITE:(write|read)} access to JMCR is unpredictable. Fields:

WRITE unsigned int
access is write.

ArchMsg.Warning.warning_wcr_mask_and_bas

TAGS UNPREDICTABLE DISPLAY DBGWCR_EL1.MASK is non-zero and BAS is not 0xff.

ArchMsg.Warning.warning_wcr_mask_reserved

TAGS UNPREDICTABLE DISPLAY DBGWCR_ELn.MASK is set to a reserved value.

ArchMsg.Warning.warning_wcr_non_configuous_bas

TAGS UNPREDICTABLE DISPLAY DBGWCR_EL1.BAS has non-contiguous set of ones.

ArchMsg.Warning.warning_wvr_masked_bit_not_zero

TAGS UNPREDICTABLE DISPLAY DBGWVRn_EL1.VA contains a bit masked by DBGWCRn_EL1.MASK that is not zero.

ArchMsg.Why.why_branch_target_exception

A branch target exception has occurred. DISPLAY branch target exception: target pc=%{PC}, target opcode=%{OPCODE}. PRIMARY KEY PC. Fields:

BTYPE unsigned int
pstate.BTYPE at time branch taken.

OPCODE unsigned int
opcode at branch target.

PC unsigned int
Virtual Address of branch target.

ArchMsg.Why.why_illegal_state

An illegal mode change has occurred. DISPLAY illegal mode change: %{MESSAGE}. Fields:

MESSAGE string
The message.

ArchMsg.warning_invalid_interval_counter_reload_value

Invalid interval counter reload value in PMSIRR_EL1.INTERVAL: software should set this to a value greater than or equal to the minimum indicated by PMSIDR_EL1.Interval.

ArchMsg.warning_invalid_spe_buffer_write_pointer

Invalid value programmed in PMBPTR_EL1.PTR or PMBLIMITR_EL1.LIMIT: Current write pointer must be at least one sample record below the write limit pointer and PMBPTR_EL1.PTR[63:56] must be equal to PMBLIMITR_EL1.LIMIT[63:56].

BRANCH_MISPREDICT

Simulating branch mispredict. Fields:

PC unsigned int

Origin address (or 0 if unavailable).

BRA_DIR

Direct branches, to immediate address. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISET enum

The instruction set of the branch instruction.

IS_COND bool

Indicates if this is a conditional branch.

IS_HINTED bool

Whether the conditional instruction is hinted.

IS_ISB bool

Whether the instruction is ISB.

IS_LINK bool

Indicates if this is a branch with link.

PC unsigned int

The address of the branch instruction.

TARGET_ISET enum

The instruction set after the branch.

TARGET_PC unsigned int

The address the instruction branches to.

BRA_INDIR

Indirect branches, perhaps to a register. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISSET enum

The instruction set of the branch instruction.

IS_COND bool

Indicates if this is a conditional branch.

IS_HINTED bool

Whether the conditional instruction is hinted.

IS_ISB bool

Whether the instruction is ISB.

IS_LINK bool

Indicates if this is a branch with link.

PC unsigned int

The address of the branch instruction.

TARGET_ISET enum

The instruction set after the branch.

TARGET_PC unsigned int

The address the instruction branches to.

BRA_INDIR_CCFAIL

Indirect branches fail, perhaps to a register.

BRA_RET_CCFAIL

Branch return fail, perhaps to a register.

CACHE_MAINTENANCE_OP

Cache Maintenance Operation. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

DATA unsigned int

Specified MVA or Set/way.

FUNCTION enum

Clean / Invalidate.

SCOPE enum

Affected region.

SIDE enum

Inst / Data.

CCFAIL

Conditional instruction condition check fail. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CCFAIL_UNC

Unconditional instruction condition check fail. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CCPASS

Conditional instruction condition check pass. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CCPASS_UNC

Unconditional instruction condition check pass. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CODE_CACHE_MAINT

Code cache maintenance. Fields:

END_INCL unsigned int

Inclusive end address for VA or PA requests.

NS bool

Non-secure world for PA.

START unsigned int

Start address for VA or PA requests.

TYPE **enum**

Request type.

COMPILE_BLOCK_END

Last instruction of basic block translated. Fields:

VADDR **unsigned int**

Address of next instruction after this basic block.

COMPILE_BLOCK_START

First instruction of basic block translated. Fields:

VADDR **unsigned int**

Address of first instruction in basic block.

COMPILE_INST

ARM instruction compiled. Fields:

DISASS **string**

Disassembly of the instruction.

ISSET **enum**

The instruction set of this instruction.

ITSTATE **unsigned int**

The ITSTATE current for the instruction.

OPCODE **unsigned int**

The opcode of the instruction.

PC **unsigned int**

The address of the instruction.

SIZE **unsigned int**

The size of the instruction in bytes.

CONTEXTIDR

Context ID Register write. Fields:

CORE_NUM **unsigned int**

Core number in a multi processor.

NS **bool**

Secure or nonsecure banked register is accessed.

UNDEF **bool**

The register accessed is undefined.

VALUE **unsigned int**

The new value written.

CONTEXT_SYNC

Called for every context synchronization event. Fields:

ARCH_CSE bool

Architectural Context Synchronization Event.

PC unsigned int

The address of the synchronization instruction.

SYSREG_SYNC bool

System register synchronization event.

CORE_ENDIAN

Core BE8 Big-Endian state changed. Fields:

BE8 bool

Core BE8 Big-Endian state.

CORE_INFO

Static processor attributes. Only triggered by a call to DumpState(). Fields:

ARCH_PROFILE enum

The architecture profile of the core.

CLUSTER_ID unsigned int

The cluster ID of this processor.

CORE_NUM unsigned int

The number of this core in an MP processor.

DETAILED_CORE_FEATURES enum

Detailed core features as a bitfield. Enum values are bitmasks to extract the info from the field.

FPU_VERSION enum

The VFP version implemented by the core.

MEM_ARCH enum

The memory architecture of the core.

MIDR_EL1 unsigned int

MIDR_EL1 value.

NUM_CORES unsigned int

The number of cores in this MP processor.

QUANTUM_SIZE unsigned int

The default quantum size of the core (in ticks).

SECURITY_FEATURES bool

Does the core have security features?.

CORE_LOADS

Processor load accesses. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ACQREL enum

Is this an acquire/release.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DATA unsigned int

The data read or written.

ELEMENT_SIZE unsigned int

Width of each element.

ENDIANNESS enum

Byte ordering of this access.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

LOCK enum

Normal, exclusive or locked access.

MECID unsigned int

Memory Encryption Context ID, if applicable.

MECID2 unsigned int

Memory Encryption Context ID of the second page, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PADDR_TAG unsigned int

If tag checked, the physical address tag for the access.

PAGE_SIZE unsigned int

Page size as $\log_2(\text{size in bytes})$.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK, 2=Exclusive Failed, 3=Unsupported.

SIZE unsigned int

Width of the access in bytes. Only required if DATA is not traced.

TAG_CHECKED bool

Is this access tag checked.

TAG_CHECKED2 bool

Is the second page access tag checked.

TRANS bool

Is this a translated access.

VADDR unsigned int

The virtual address of the access.

CORE_REGS

Changes of the core registers R0 to R14. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number, 0 to 14.

MODE enum

Bank of the register accessed.

OLD_VALUE unsigned int

The old value overwritten.

PHYS_ID enum

The physical register accessed.

VALUE unsigned int

The new value written to the register.

CORE_REGS64

Changes of the core registers X0..X30, SP_ELn. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ID enum

The register number, 0-30 for X0-X30, >=32 for SP_ELn.

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

CORE_REGS64_READ

General purpose AArch64 register has been read X0..X30, SP_ELn. Fields:

ID enum

The register number, 0-30 for X0-X30, >=32 for SP_ELn.

VALUE unsigned int

The value read from the register.

CORE_STALLED_TRANSITION

Detect when a core is prevented from progressing PChannel states. Fields:

CURRENT_STATE enum

Current power mode of core.

REASON string

The reason a transition was stalled, and what must be done to unblock it.

TARGET_STATE enum

Desired power mode of core after transition.

CORE_STORES

Processor store accesses. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ACQREL enum

Is this an acquire/release.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DATA unsigned int

The data read or written.

ELEMENT_SIZE unsigned int

Width of each element.

ENDIANNESS enum

Byte ordering of this access.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

LOCK enum

Normal, exclusive or locked access.

MECID unsigned int

Memory Encryption Context ID, if applicable.

MECID2 unsigned int

Memory Encryption Context ID of the second page, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PADDR_TAG unsigned int

If tag checked, the physical address tag for the access.

PAGE_SIZE unsigned int

Page size as $\log_2(\text{size in bytes})$.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK, 2=Exclusive Failed, 3=Unsupported.

SIZE unsigned int

Width of the access in bytes. Only required if DATA is not traced.

TAG_CHECKED bool

Is this access tag checked.

TAG_CHECKED2 bool

Is the second page access tag checked.

TRANS bool

Is this a translated access.

VADDR unsigned int

The virtual address of the access.

CP14_READ

System Coprocessor register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP14 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc1 unsigned int

opcode 1.

opc2 unsigned int

opcode 2.

CP14_WRITE

System Coprocessor register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP14 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc1 unsigned int

opcode 1.

opc2 unsigned int

opcode 2.

CP15_READ

System Coprocessor register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP15 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc1 unsigned int

opcode 1.

opc2 unsigned int

opcode 2.

CP15_READ64

System Coprocessor register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP15 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc unsigned int

opcode 1.

CP15_WRITE

System Control Coprocessor register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP15 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc1 unsigned int

opcode 1.

opc2 unsigned int

opcode 2.

CP15_WRITE64

System Coprocessor register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP15 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc unsigned int

opcode 1.

CPSR

CPSR change. (In AArch64 PSTATE represented in SPSR format). Fields:

CORE_NUM unsigned int

Core number in a multi processor.

OLD_VALUE unsigned int

The old CPSR value.

UNKNOWN unsigned int

Bits within the register that have unknown value.

VALUE unsigned int

The new CPSR value.

CRYPTO_SPEC

Every crypto instruction speculatively executed.

CorePowerStateChange

Core power-up state has changed. DISPLAY Core powered %{PU:(up|down)}. Fields:

PU bool

Whether the core powered up.

DATA_CACHE_SET_ALLOC_TAGS

Set allocation tags from DC GVA instruction. Fields:

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

NSDESC unsigned int

The security state of the access.

PADDR unsigned int

The physical address of the access.

PAS enum

The physical address space of the page.

RESPONSE enum

0=Aborted, 1=OK.

SIZE unsigned int

The size of memory in bytes.

TAG unsigned int

The value of the tag being set.

TAGGING_TYPE enum

Type of Memory tagging (Physical or Virtual).

TAG_VADDR unsigned int

The virtual address of the Virtual Tag if Virtual tagging is used; otherwise 0.

VADDR unsigned int

Virtual address used for setting allocation tags.

DATA_CACHE_ZERO

Zero and invalidate cache line from DC ZVA instruction. Fields:

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

NSDESC unsigned int

The security state of the access.

PADDR unsigned int

The physical address of the access.

PADDR_TAG unsigned int

If tag checked, the physical address tag for the access.

PAS enum

The physical address space of the page.

RESPONSE enum

0=Aborted, 1=OK.

SIZE unsigned int

The size of zero'd memory in bytes.

TAG_CHECKED bool

Is this access tag checked.

VADDR unsigned int

Virtual address of the zero'd cache line.

DEBUG_EVENT

Hardware debug support event. Fields:

EVENT enum

Description of event.

VALUE unsigned int
data value.

DEBUG_TARGET_ISET

Hardware debug target ISA. Fields:

TARGET_ISET enum
The target instruction set of the debug state.

DMI_ALLOCATE

DMI allocated. Fields:

ACCESS_DIRECTION enum
READ / WRITE.

PADDR unsigned int
The physical address of the DMI.

PTR unsigned int
DMI range pointer.

SIDE enum
DSIDE / ISIDE.

SIZE unsigned int
The size of DMI.

VADDR unsigned int
The virtual address of the DMI..

DMI_HIT

DMI hit, access used DMI (only fire for small DMI regions i.e. less than a full page). Fields:

ACCESS_DIRECTION enum
READ / WRITE.

PADDR unsigned int
The physical address of the DMI.

SIDE enum
DSIDE / ISIDE.

SIZE unsigned int
The size of the DMI.

VADDR unsigned int
The virtual address of the DMI..

DMI_REVOKE

DMI revoked. Fields:

ACCESS_DIRECTION enum
READ / WRITE.

PADDR unsigned int

The physical address of the DMI.

SIDE enum

DSIDE / ISIDE.

SIZE unsigned int

The size of DMI.

VADDR unsigned int

The virtual address of the DMI..

END_COMPILE

Compilation end. Fields:

END_OF_PAGE_COUNT unsigned int

Number of basic blocks exited due to page boundary since START_COMPILE.

FETCHFAIL_COUNT unsigned int

Number of basic blocks exited due to ifetch failure START_COMPILE.

INST_COUNT unsigned int

Number of instructions compiled since START_COMPILE.

NONSEQ_COUNT unsigned int

Number of basic blocks exited due to non-sequential instructions since START_COMPILE.

PAGE_STRADDLE_COUNT unsigned int

Number of basic blocks exited due to unaligned instructions crossing page since START_COMPILE.

ESB

Error Synchronization Barrier hit. Fields:

ASYNC bool

ESB is async.

IMPLICIT bool

ESB is implicit.

EXCEPTION

Exception taken. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ESR unsigned int

The value assigned to ESR when exception is taken.

ESR_EC unsigned int

The Value assigned to ESR.EC when exception is taken.

IS_PHYSICAL bool

Physical or Virtual exception.

IS_TRAP bool

Exception coming from a trap.

LR unsigned int

The value assigned to the link register.

PC unsigned int

The location where the exception occurred.

PREFERRED_RETURN unsigned int

The preferred return address for the exception.

TAKEN_LOCALLY bool

Whether exception is taken locally(same exception level).

TARGET_ISET enum

The instruction set of the exception handler code.

TARGET_PC unsigned int

The address the exception branches to.

VECTOR enum

The exception vector.

VECTOR_OFFSET unsigned int

The offset of an exception vector from the base address.

EXCEPTION_END

Every exception completed.

EXCEPTION_INFO

Trace human-readable reason of the current exception for data/instruction aborts, floating point exceptions and unknown exceptions. Fields:

DESCRIPTION string

Reason for current exception.

ESR_EC unsigned int

The value of ESR.EC when exception is taken.

EXCEPTION_RAISE

Every exception raised.

EXCEPTION_RETURN

Branches on leaving exception. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISSET enum

The instruction set of the branch instruction.

IS_COND bool

Indicates if this is a conditional branch.

IS_HINTED bool

Whether the conditional instruction is hinted.

IS_ISB bool

Whether the instruction is ISB.

IS_LINK bool

Indicates if this is a branch with link.

PC unsigned int

The address of the branch instruction.

TARGET_ISSET enum

The instruction set after the branch.

TARGET_PC unsigned int

The address the instruction branches to.

EXCEPTION_RETURN_PREBRANCH

Exception return event that must occur before PC and processor state have updated.

EXCEPTION_START

Every exception started.

FIQ_TAKEN

FIQ taken exception.

FP_EXCEPTIONS_SIGNALLED

Trace which FP exceptions were signalled as a part of the FP operations performed by an instruction. Fields:

FP_EXCEPTIONS_SIGNALLED_MASK unsigned int

Indicates which FP exceptions were signalled by an instruction. This uses the same encoding as the cumulative floating-point exception bitfields in FPSR.

FP_STATE

Floating point state. Fields:

FPSR unsigned int

FPSR register value.

MASK unsigned int

The mask indicating updated cumulative bits.

FREQ_CHANGED

Core frequency changed. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The instruction count of this core.

NEW_FREQ signed int

The new frequency of this core (expressed in Hz).

OLD_FREQ signed int

The old frequency of this core (expressed in Hz).

HLT

HLT instruction occurred. Fields:

IMM16 unsigned int

The 16-bit immediate value encoded in HLT instruction.

INST

Every instruction executed. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DEBUG_STATE bool

The instruction is executed in debug state.

DISASS string

Disassembly of instruction.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISSET enum

The current instruction set.

ITSTATE unsigned int

The current ITSTATE.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

MODE enum

The mode the core is in.

NS unsigned int

The core's non-secure bit.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

OPCODE unsigned int

The opcode of the instruction.

PADDR unsigned int

The physical address of the instruction.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the instruction.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the page.

PC unsigned int

The address of the instruction.

SECURITY_STATE enum

The core's security state.

SIZE unsigned int

The size of the instruction in bytes.

INST_END

Every instruction completed.

INST_START

Every instruction started. Fields:

DEBUG_STATE bool

The instruction is executed in debug state.

ISET enum

The current instruction set.

MODE enum

The mode the core is in.

NS enum

The current Secure State.

PC unsigned int

The address of the conditional instruction.

SECURITY_STATE enum

The current Security State.

INST_STRADDLE

Instruction straddles boundary. Fields:

DEBUG_STATE bool

The instruction is executed in debug state.

ISET enum

The current instruction set.

MODE enum

The mode the core is in.

NS enum

The current Secure State.

PC unsigned int

The address of the conditional instruction.

SECURITY_STATE enum

The current Security State.

IRQ_TAKEN

IRQ taken exception.

ISB

ISB instruction occurred. Fields:

PC unsigned int

The address of the ISB instruction.

LOAD_MULTIPLE_REGS_LIST

List of destination registers (not source) of a multiple registers load instruction. Fields:

REG string

Register to add into the list.

LOCAL_MONITOR

Local monitor activity. Fields:

PADDR unsigned int

Local Monitor Address.

State enum

State of the monitor (Open/Exclusive).

MAPPING_CREATE

A memory mapping has been created. This is an implementation detail of the model and so may change without notice. Fields:

ASID unsigned int

The ASID used during translation, or 0 if no ASID was used.

IS_VALID bool

Whether the mapping is valid or not.

PADDR unsigned int

Base physical address of region, or 0 if not known.

SIDE enum

I-side or D-side.

SIZE unsigned int

Size of the page, or 0 if not known.

VADDR unsigned int

Base virtual address of region.

VMID unsigned int

The VMID used during translation, or 0 if no VMID was used.

MAPPING_DESTROY

A memory mapping has been destroyed. This is an implementation detail of the model and so may change without notice. Fields:

ASID unsigned int

The ASID used during translation.

CAUSE enum

The cause of invalidation.

PADDR unsigned int

Base physical address of page.

SIDE enum

I-side or D-side.

SIZE unsigned int

Size of the page.

VADDR unsigned int

Base virtual address of page.

VMID unsigned int

The VMID used during translation.

MEMMAP_DEBUG_READ

Memory mapped reads to the debug registers. Fields:

ADDR unsigned int

address.

CORE_NUM unsigned int

Core number in a multi processor.

EXT bool

Whether access is from an external device (such as the DAP).

NS enum

Secure state of the access.

REG_NAME string

Name of the debug register accessed.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

MEMMAP_DEBUG_WRITE

Memory mapped writes to the debug registers. Fields:

ADDR unsigned int

address.

CORE_NUM unsigned int

Core number in a multi processor.

EXT bool

Whether access is from an external device (such as the DAP).

NS enum

Secure state of the access.

REG_NAME string

Name of the debug register accessed.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The new value written.

MEMTAG_LOADS

Allocation tag memory read; when attributes and system configuration allow tag accesses. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

NUM_GRANULES unsigned int

Number of tags to be Loaded/stored.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK.

TAGGING_TYPE enum

Type of Memory tagging (Physical or Virtual).

TAGS unsigned int

The tags read or written. Tags for subsequent granules would be found at subsequent bytes in this field (One byte for each tag).

TAG_VADDR unsigned int

The virtual address of the Virtual Tag if Virtual tagging is used; otherwise 0.

VADDR unsigned int

The virtual address of the access.

MENTAG_LOAD_INST

Allocation tag load instruction; regardless of tags being actually accessed. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate

(Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

NUM_GRANULES unsigned int

Number of tags to be Loaded/stored.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK.

TAGGING_TYPE enum

Type of Memory tagging (Physical or Virtual).

TAGS unsigned int

The tags read or written. Tags for subsequent granules would be found at subsequent bytes in this field (One byte for each tag).

TAG_VADDR unsigned int

The virtual address of the Virtual Tag if Virtual tagging is used; otherwise 0.

VADDR unsigned int

The virtual address of the access.

MEMTAG_STORES

Allocation tag memory write; when attributes and system configuration allow tag accesses. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

NUM_GRANULES unsigned int

Number of tags to be Loaded/stored.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK.

TAGGING_TYPE enum

Type of Memory tagging (Physical or Virtual).

TAGS unsigned int

The tags read or written. Tags for subsequent granules would be found at subsequent bytes in this field (One byte for each tag).

TAG_VADDR unsigned int

The virtual address of the Virtual Tag if Virtual tagging is used; otherwise 0.

VADDR unsigned int

The virtual address of the access.

MEMTAG_STORE_INST

Allocation tag store instruction; regardless of tags being actually accessed. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes

([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

NUM_GRANULES unsigned int

Number of tags to be Loaded/stored.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK.

TAGGING_TYPE enum

Type of Memory tagging (Physical or Virtual).

TAGS unsigned int

The tags read or written. Tags for subsequent granules would be found at subsequent bytes in this field (One byte for each tag).

TAG_VADDR unsigned int

The virtual address of the Virtual Tag if Virtual tagging is used; otherwise 0.

VADDR unsigned int

The virtual address of the access.

MMU_D128_TTB_READ

This event is triggered by reads caused by a D128 translation table walk. Fields:

ABORTED bool

Was the walk successful.

DATA_HIGH unsigned int

The data upper bits read.

DATA_LOW unsigned int

The data lower bits read.

IPA unsigned int

For stage 1, the IPA of the read.

LEVEL unsigned int

Translation table level.

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

OUTADDR unsigned int

The output physical address.

PADDR unsigned int

The physical address of the read.

PASmem enum

Physical Address Space of the accessed memory.

PASreq enum

Physical Address Space of the requested lookup.

REGIME_EL enum

Entry matches in this translation regime.

REQUESTER enum

The requester of this table walk.

SIDE enum

Inst / Data.

STAGE unsigned int

Translation stage.

MMU_D128_TTB_WRITE

This event is triggered by writes caused by a D128 translation table walk. Fields:

ABORTED bool

Was the walk successful.

DATA_HIGH unsigned int

The data upper bits written.

DATA_LOW unsigned int

The data lower bits written.

IPA unsigned int

For stage 1, the IPA of the write.

IS_HACDBS_ACCESS bool

If the access was due to HACDBS_CLEAN.

LEVEL unsigned int

Translation table level.

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

OUTADDR unsigned int

The output physical address.

PADDR unsigned int

The physical address of the write.

PASmem enum

Physical Address Space of the accessed memory.

PASreq enum

Physical Address Space of the requested lookup.

REGIME_EL enum

Entry matches in this translation regime.

SIDE enum

Inst / Data.

STAGE unsigned int

Translation stage.

MMU_TRANS

Address translation information. Fields:

ASID unsigned int

Address space identifier.

CORE_NUM unsigned int

Core number in a multi processor.

Hyp bool

Entry matches in Hyp state only.

INNERCACHE_RA bool

Is the inner cache allocate on read.

INNERCACHE_TRANSIENT bool

Is the inner write-through transient.

INNERCACHE_TYPE enum

Inner Caching scheme (NC/MB/WA).

INNERCACHE_WA bool

Is the inner cache allocate on write.

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MEMTYPE enum

Memory type.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC enum

Is secure side supposed to access secure or nonsecure memory.

OUTERCACHE_RA bool

Is the outer cache allocate on read.

OUTERCACHE_TRANSIENT bool

Is the outer write-through transient.

OUTERCACHE_TYPE enum

Outer Caching scheme (NC/MB/WA).

OUTERCACHE_WA bool

Is the outer cache allocate on write.

PADDR unsigned int

Physical address of the access.

PAGESIZE unsigned int

Page size as $\log_2(\text{size})$.

PAS enum

PAS supposed to be accessed is secure or nonsecure memory or Root/Realm if RME is implemented.

SH enum

Shareability.

SIDE enum

Inst / Data.

STAGE1_PERM unsigned int

Stage 1 permissions, [5:3] Privileged access (XWR) [2:0] User access (XWR).

STAGE2_PERM unsigned int

Stage 2 permission mask [5:3] Privileged access (XWR) [2:0] User access (XWR).

VADDR unsigned int

Virtual address of the access.

VMID unsigned int

Virtual machine identifier.

XS bool

XS Attribute.

nG enum

Flag indicating whether ASID will be matched.

MMU_TTB_READ

This event is triggered by reads caused by a translation table walk. Fields:

ABORTED bool

Was the walk successful.

DATA unsigned int

The data read.

DESC_KIND enum

The encoding scheme of the descriptors used for this walk.

IPA unsigned int

For stage 1, the IPA of the read.

LEVEL unsigned int

Translation table level.

LPAAE bool

Is this for an LPAAE translation (*DEPRECATED*, use DESC_KIND).

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSmem bool

Non secure memory access.

NSreq bool

Non secure request.

OUTADDR unsigned int

The output physical address.

PADDR unsigned int

The physical address of the read.

PASmem enum

Physical Address Space of the accessed memory.

PASreq enum

Physical Address Space of the requested lookup.

REGIME_EL enum

Entry matches in this translation regime.

REQUESTER enum

The requester of this table walk.

SIDE enum

Inst / Data.

STAGE unsigned int

Translation stage.

MMU_TTB_WRITE

This event is triggered by writes caused by a translation table walk. Fields:

ABORTED bool

Was the walk successful.

DATA unsigned int

The data written.

DESC_KIND enum

The encoding scheme of the descriptors used for this walk.

IPA unsigned int

For stage 1, the IPA of the write.

IS_HACDBS_ACCESS bool

If the access was due to HACDBS_CLEAN.

LEVEL unsigned int

Translation table level.

LPAE bool

Is this for an LPAE translation (*DEPRECATED*, use DESC_KIND).

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSmem bool

Non secure memory access.

NSreq bool

Non secure request.

OUTADDR unsigned int

The output physical address.

PADDR unsigned int

The physical address of the write.

PASmem enum

Physical Address Space of the accessed memory.

PASreq enum

Physical Address Space of the requested lookup.

REGIME_EL enum

Entry matches in this translation regime.

SIDE enum

Inst / Data.

STAGE unsigned int

Translation stage.

MODE_CHANGE

Mode change. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

MODE enum

The new mode.

NON_SECURE enum

The core's new non-secure bit.

OLD_MODE enum

The old mode.

SECURITY_STATE enum

The new security state.

NOT_TAG_CHECKED_INST

MTE Tag not checked for current instruction.

PERIODIC

Called for every quantum. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The instruction count of this CPU.

PC unsigned int

The address of the next instruction to be executed on this CPU.

PMU_COUNTER_OVERFLOW

PMU counter overflow. Fields:

COUNTER_GROUP enum

Counter group.

COUNTER_RANGE enum

Counter range.

EVENT_ID unsigned int

Event Identifier, field is valid only for Event counter group.

INDEX unsigned int

Counter index (as selected by PMSELR), field is valid only for Event/Cycle counter groups.

INTERRUPT bool

Is interrupt enabled on overflow for this counter.

POINTER_AUTH

Called when execution hits a pointer authentication check. Fields:

AUTH_STATE bool

Authentication result.

KEY_USED enum

Pointer authentication key.

POTINDEX_UPDATE

Value of POT index for current pc changed. Fields:

NEW_POTINDEX unsigned int

New POTIndex.

OLD_POTINDEX unsigned int

Old POTIndex.

PREFETCH_MEMORY64

Prefetch from PRFM or PRFUM instructions. Fields:

PRFOP enum

Prefetch hint.

VADDR unsigned int

Virtual address of the location that should be prefetched.

PRELOAD64_REQUEST

Preload requests from PRFM or PRFUM instructions. Fields:

PRFOP enum

Prefetch hint.

VADDR unsigned int

Virtual address of the location that should be prefetched.

PRELOAD_DATA

Data preload from PLD instruction. Fields:

VADDR unsigned int

Virtual address of the data that should be preloaded.

PRELOAD_INST

Instruction preload from PLI instruction. Fields:

VADDR unsigned int

Virtual address of the instruction that should be preloaded.

PRE_CORE_LOAD

Trace just before a core load. Fields:

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

PRE_CORE_STORE

Trace just before a core store. Fields:

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

PRE_TTB_READ

This event is triggered before reads caused by a translation table walk. Fields:

ADDR unsigned int

The physical address of the read.

PSTATE

PSTATE fields update. Fields:

ALLINT unsigned int

All interrupt mask bit.

BTYPE unsigned int

Branch target identification bit.

CORE_NUM unsigned int

Core number in a multi processor.

DAIF unsigned int

Exception mask bits.

DIT unsigned int

Data Independent Timing bit.

EL unsigned int

Current Exception level field.

EXLOCK unsigned int

Exception return state lock bit.

IL unsigned int

Illegal Execution state bit.

NZCV unsigned int

Condition flags.

PACM unsigned int

PACM bit.

PAN unsigned int

Privileged Access Never state bit.

PM unsigned int

PMU exception mask bit.

PPEND unsigned int

PMU exception pending bit.

SM unsigned int

StreamingMode status bit.

SP unsigned int

Stack pointer register selection bit.

SS unsigned int

Software Step bit.

SSBS unsigned int

Speculative Store Bypass Safe bit.

TCO unsigned int

Tag Check Override bit.

TOUCHED unsigned int

Mask of the fields that have been touched since this trace has last been emitted. Note that touched does not necessarily mean the value changed. The index of the field is the bit index in the mask plus the index of this field plus one.

UAO unsigned int

User Access Override bit.

UINJ unsigned int

Undefined Instruction exceptions bit.

ZA unsigned int

ZA storage and ZT0 register enable bit.

nRW unsigned int

Current Execution state bit.

PSTATE_PACTIVE_OPMODE_TRANSITION

Trace pactive and Opmode mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Operating mode after the transition.

OLD_STATE enum

Operating mode before the transition.

PSTATE_PACTIVE_OPMODE_TRANSITION_DENIED

Trace the denied pactive and Opmode mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Operating mode after the transition.

OLD_STATE enum

Operating mode before the transition.

REASON string

Reason for the transition being denied.

PSTATE_PACTIVE_POWER_TRANSITION

Trace pactive and power mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Power mode after the transition.

OLD_STATE enum

Power mode before the transition.

PSTATE_PACTIVE_POWER_TRANSITION_DENIED

Trace the denied pactive and power mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Power mode after the transition.

OLD_STATE enum

Power mode before the transition.

REASON string

Reason for the transition being denied.

PSTATE_PACTIVE_THREAD_OPMODE_TRANSITION

Trace pactive and Opmode mode transitions for threads on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Operating mode after the transition.

OLD_STATE enum

Operating mode before the transition.

QUANTUM_START

Called at the start of every quantum.

RANGE_PRELOAD_REQUEST

Preload requests from RPRFM. Fields:

COUNT unsigned int

Number of blocks of data to access.

LENGTH unsigned int

Number of contiguous bytes to be accessed in each block.

OP unsigned int

Range prefetch operation .

REUSE unsigned int

Maximum number of bytes to be accessed before next RPRFM .

STRIDE signed int

Stride to advance after accessing LENGTH bytes.

VADDR unsigned int

Base virtual address of range prefetch request.

RUN_STATE

Run state transition. Fields:

INST_COUNT unsigned int

Ticks count at point of transition.

NEW enum

New run state.

OLD enum

Old run state.

SBP_SAMPLE_COLLISION

DISPLAY Statistical profiling samples collided .

SBP_SAMPLE_DISCARDED

DISPLAY Operation discarded by statistical profiling unit when configured in discard mode.
Fields:

BUFFER_POINTER unsigned int

Value of buffer pointer.

DATA unsigned int

Data discarded.

SIZE unsigned int

Size of sample, in bytes.

SBP_SAMPLE_END

DISPLAY Statistical profiling of an operation ended.

SBP_SAMPLE_POP

DISPLAY Operation which might be sampled by statistical profiling.

SBP_SAMPLE_RECORDED

DISPLAY Operation sampled by statistical profiling has been written to the buffer at
%{BUFFER_ADDRESS} with size %{SIZE} and data %{DATA}. Fields:

BUFFER_ADDRESS unsigned int

Virtual Address of start of buffer write.

DATA unsigned int

Data written to the buffer.

SIZE unsigned int

Size of sample written to buffer, in bytes.

SBP_SAMPLE_START

DISPLAY Statistical profiling of an operation started.

SBP_SAMPLE_TAKEN

DISPLAY Operation sampled by statistical profiling after filtering.

SEMIHOSTING_CALL

Call of a semihost function occurred. Fields:

PC unsigned int

The program counter after the semihosting call.

SEMIHOSTING_PRECALL

About to call semihost. Fields:

EL2 bool

Use EL2 translation regime.

NS bool

Is Non-Secure.

REG_WIDTH unsigned int

The current register width in bytes.

SIGNAL

External signal state change. Fields:

SIGNAL enum

Signal that changed.

STATE bool

Signal asserted state.

SOFTWARE_STEP

Return the Debug Software Step state. Fields:

STATE enum

Software Step state.

SPSR

SPSR change. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

MODE enum

Which of the banked SPSR registers is written.

OLD_VALUE unsigned int

The old SPSR value.

VALUE unsigned int

The new SPSR value.

START_COMPILE

Compilation started. Fields:

VADDR unsigned int

Instruction where compilation begins.

STG_MULTIPLE_MICRO_OPS_ADDRESS_RANGE

Address range for MTE stores with multiple uops. Fields:

ADDR_END unsigned int

Final address.

ADDR_START unsigned int

Starting address.

STORE_MULTIPLE_REGS_ADDRESS_RANGE

Address range for stores with multiple registers. Fields:

ADDR_END unsigned int

Final address.

ADDR_START unsigned int

Starting address.

SUBOPTIMAL_LDST_RETIRED

suboptimal load/store. Fields:

ACCESS_SIZE unsigned int

Log2 of the access width in bytes used for alignment checking (1: half-word, 2: word ...).

OPTIMAL_ALIGN_SELECTOR unsigned int

Selector to decide the optimal alignment size (1: access width, 2: 4-bytes, 3: 8-bytes, ..., 12: 4-kbytes).

VADDR unsigned int

The virtual address of the access.

SVE_INST_SPEC

SVE operations speculatively executed.

SVE_LD_RETIRED

SVE load instructions architecturally executed.

SVE_LOADS

SVE load accesses. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ACQREL enum

Is this an acquire/release.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DATA unsigned int

The data read or written.

ELEMENT_SIZE unsigned int

Width of each element.

ENDIANNESS enum

Byte ordering of this access.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

LOCK enum

Normal, exclusive or locked access.

MECID unsigned int

Memory Encryption Context ID, if applicable.

MECID2 unsigned int

Memory Encryption Context ID of the second page, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PADDR_TAG unsigned int

If tag checked, the physical address tag for the access.

PAGE_SIZE unsigned int

Page size as $\log_2(\text{size in bytes})$.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK, 2=Exclusive Failed, 3=Unsupported.

SIZE unsigned int

Width of the access in bytes. Only required if DATA is not traced.

TAG_CHECKED bool

Is this access tag checked.

TAG_CHECKED2 bool

Is the second page access tag checked.

TRANS bool

Is this a translated access.

VADDR unsigned int

The virtual address of the access.

SVE_MEM_PREFETCH

Scalable Vector Extension memory prefetch. Fields:

CACHE_LEVEL enum

The preferred cache level.

HINT enum

What the memory is expected to be used for.

SIZE unsigned int

The prefetch size in bytes.

TEMPORAL bool

Whether the data is temporal.

VADDR unsigned int

The virtual address of the prefetch.

SVE_REG_READ

Scalable Vector Extension register read. Fields:

ID enum

The register identifier.

SM bool

Whether the PE is in Streaming Mode.

VALUE unsigned int

The read data.

SVE_REG_UPDATE

Scalable Vector Extension register update. Fields:

ID enum

The register identifier.

SM bool

Whether the PE is in Streaming Mode.

VALUE unsigned int

The written data.

SVE_REG_WRITE

Scalable Vector Extension register write. Fields:

ID enum

The register identifier.

SM bool

Whether the PE is in Streaming Mode.

VALUE unsigned int

The written data.

SVE_STORES

SVE store accesses. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ACQREL enum

Is this an acquire/release.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DATA unsigned int

The data read or written.

ELEMENT_SIZE unsigned int

Width of each element.

ENDIANNESS enum

Byte ordering of this access.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

LOCK enum

Normal, exclusive or locked access.

MECID unsigned int

Memory Encryption Context ID, if applicable.

MECID2 unsigned int

Memory Encryption Context ID of the second page, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PADDR_TAG unsigned int

If tag checked, the physical address tag for the access.

PAGE_SIZE unsigned int

Page size as $\log_2(\text{size in bytes})$.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK, 2=Exclusive Failed, 3=Unsupported.

SIZE unsigned int

Width of the access in bytes. Only required if DATA is not traced.

TAG_CHECKED bool

Is this access tag checked.

TAG_CHECKED2 bool

Is the second page access tag checked.

TRANS bool

Is this a translated access.

VADDR unsigned int

The virtual address of the access.

SVE_ST_RETIRED

SVE store instructions architecturally executed.

SVE_UNALIGNED_LDST_RETIRED

SVE load or store instructions architecturally executed that would cause alignment fault if A bit is set.

SVE_Z_PRED_MEM_ACCESS_INFO

Information about SVE/SME predicated accesses to Z registers (non Gather/Scatter). Fields:

MEM_ELEM_SIZE unsigned int

Width of elements accessed in memory (in bytes).

RnW bool

Whether this is a Read not a Write.

VA unsigned int

Virtual base address computed for accesses.

SYNC

Called for every synchronization, that is every quantum break. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The instruction count of this CPU.

LOCAL_QUANTUM unsigned int

The local quantum of this CPU.

LOCAL_TIME unsigned int

The local time of this CPU.

SYNC_REASON enum

Reason for sync.

SYSCALL

System call instruction executed. Fields:

IMM unsigned int

Immediate value of the system call instruction.

TYPE enum

System call type.

VADDR unsigned int

Instruction that caused the system call.

SYSREG_READ128

System Coprocessor 128 bit register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

REGNUM enum

Internal register number.

REGNUM_PHYS enum

Internal register number physically accessed (e.g. in presense of VHE redirection).

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_READ64

System Coprocessor register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REGNUM enum

Internal register number.

REGNUM_OPERAND enum

Register number corresponding to the instruction operand.

REGNUM_PHYS enum

Internal register number physically accessed (e.g. in presense of VHE redirection).

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_UPDATE128

Triggers when the system updates a register. Fields:

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG enum

Register number.

UNKNOWN unsigned int

Bits of the register which became unknown.

VALUE unsigned int

Value written to the register.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_UPDATE32

Triggers when the system updates a register. Fields:

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG enum

Register number.

UNKNOWN unsigned int

Bits of the register which became unknown.

VALUE unsigned int

Value written to the register.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_UPDATE64

Triggers when the system updates a register. Fields:

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG enum

Register number.

UNKNOWN unsigned int

Bits of the register which became unknown.

VALUE unsigned int

Value written to the register.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_WRITE128

System Coprocessor 128 bit register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

REGNUM enum

Internal register number.

REGNUM_PHYS enum

Internal register number physically accessed (e.g. in presense of VHE redirection).

UNDEF bool

The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_WRITE64

System Coprocessor register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REGNUM enum

Internal register number.

REGNUM_OPERAND enum

Register number corresponding to the instruction operand.

REGNUM_PHYS enum

Internal register number physically accessed (e.g. in presense of VHE redirection).

UNDEF bool

The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

TRBU_BUFFER_WRAP

DISPLAY Trace buffer current write pointer wrapped.

TRBU_TRIGGER_EVENT

DISPLAY Trace buffer Trigger Event.

TRBU_WRITE

DISPLAY PETU trace data has been written to the TRBU memory buffer at %{BUFFER_ADDRESS} with size %{SIZE} and data %{DATA}. Fields:

BUFFER_ADDRESS unsigned int

Virtual Address of start of buffer write.

DATA unsigned int

Data written to the buffer.

SIZE unsigned int

Size of sample written to buffer, in bytes.

UNALIGNED_LDST_RETIRED

Processor unaligned load/store. Fields:

ACCESS_SIZE unsigned int

Log2 of the access width in bytes used for alignment checking (1=>half-word, 2->word ...).

NUMBER_OF_BEATS unsigned int

Number of accesses (beats) in this burst (Total size in bytes is ACCESS_SIZE * NUMBER_OF_BEATS).

VADDR unsigned int

The virtual address of the access.

VFP_D_REGS

VFP/NEON D 64 bit register write. Fields:

ALIASING enum

Type of register aliasing used.

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a write to the corresponding bit in the register. For example, 0x1 would indicate a write to VALUE[0].

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

VFP_D_REGS_READ

VFP/NEON D 64 bit register read. Fields:

ID unsigned int

The register number.

VALUE unsigned int

The current value of the register.

VFP_Q_REGS

VFP/NEON Q 128 bit register write. Fields:

ALIASING enum

Type of register aliasing used.

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a write to the corresponding byte in the register. For example, 0x1 would indicate a write to VALUE[7:0].

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

VFP_Q_REGS_READ

VFP/NEON Q 128 bit register read. Fields:

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a read to the corresponding byte in the register. For example, 0x1 would indicate a read to VALUE[7:0].

VALUE unsigned int

The current value of the register.

VFP_SYS_REGS

Writes to the VFP/NEON units system registers. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ID enum

Which VFP system register is written.

MASK unsigned int

The mask indicating updated cumulative bits.

OLD_VALUE unsigned int

The register's old value overwritten.

VALUE unsigned int

The new value written to the VFP system register.

VFP_S_REGS

VFP/NEON S 32 bit register write. Fields:

ALIASING enum

Type of register aliasing used.

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a write to the corresponding bit in the register. For example, 0x1 would indicate a write to VALUE[0].

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

VFP_S_REGS_READ

VFP/NEON S 32 bit register read. Fields:

ID unsigned int

The register number.

VALUE unsigned int

The current value of the register.

VIRTUAL_TAG_ALLOCATED

Virtual tag information when allocated. Fields:

TAG_VADDR unsigned int

Virtual tag virtual address.

VADDR unsigned int

Data Virtual address.

VIRTUAL_TAG unsigned int

Virtual tag value.

WAYPOINT

Signals end of basic block execution. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ISET enum

Origin instruction set.

IS_COND bool

Indicates if this is a conditional waypoint.

PC unsigned int

Origin address (or 0 if unavailable).

TAKEN bool

Indicates if this waypoint was taken.

TARGET unsigned int

Destination address.

TARGET_ISET enum

Destination instruction set.

WFE_END

WFE ended. Fields:

INST_COUNT unsigned int

Ticks count when leaving WFE.

WFE_EVENT_REGISTER

WFE event register status: set/clear, reason. Fields:

INST_COUNT unsigned int

Ticks count.

REASON enum

Reason for set/clear. Only REASON==0 (Cleared by WFE) clears the bit, all other reasons set it.

WFE_IGNORED

WFE ignored. Fields:

EVENT bool

This WFE was ignored because the event register was set.

INST_COUNT unsigned int

Ticks count when ignoring WFE.

TRAPPED bool

This WFE was trapped.

WFE_START

WFE entered. Fields:

INST_COUNT **unsigned int**

Ticks count when entering WFE.

WFI_END

WFI ended. Fields:

INST_COUNT **unsigned int**

Ticks count when leaving WFI.

WFI_IGNORED

WFI ignored. Fields:

DISABLED **bool**

This WFI was ignored because WFI is disabled.

INST_COUNT **unsigned int**

Ticks count when ignoring WFI.

REASON **enum**

specifies reason why WFI trace was ignored.

TRAPPED **bool**

This WFI was trapped.

WFI_START

WFI entered. Fields:

INST_COUNT **unsigned int**

Ticks count when entering WFI.

WFI_WAKEUP

WFI wakeup. Fields:

INST_COUNT **unsigned int**

Ticks count when WFI wakeup occurred.

REASON **enum**

Reason for wakeup.

2.62 ARM_CortexR52

This section describes the trace sources.

AA64_ASE_SVE_REGS

Modification of AArch64 Advanced SIMD or SVE Vector Registers (via SIMD or Scalar operations), SVE Predicate Registers and ZT0. Fields:

ID enum

The register identifier.

MASK unsigned int

Bitmask of the register to signal the modified bits in the VALUE field.

SM bool

Whether the PE is in Streaming Mode.

VALUE unsigned int

Value of AArch64 vector register.

ASYNC_MEMORY_FAULT

Context ID Register write. Fields:

FAULT unsigned int

Fault status present in ESR.ISS encoding for Serror in ARCH64 and present in DFSR encoding for external abort format in ARCH32.

PADDR unsigned int

Physical Address (or 0 if unavailable).

VADDR signed int

Virtual Address (or 0 if unavailable).

ATOMIC_END_ACCESS

Bus trace access for atomics. Fields:

ACCESS_FAIL bool

Memory access failed.

ADDR unsigned int

The virtual address of the access.

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

COMPARE_VALUE unsigned int

Compare value for CAS.

LOAD_VALUE unsigned int

Loaded values.

NSDESC unsigned int

The security state of the access.

OPERAND_VALUE unsigned int

Operation's operand.

OPERATION enum

Operation type.

PADDR unsigned int

The physical address of the access.

PRIV bool

Is this a privileged access?.

ATOMIC_SLAVE_ACCESS

Information on the atomic operation at slave side. Fields:

ACCESS_RESULT unsigned int

Result of atomic access.

COMPARE_VALUE unsigned int

Value to be compared with memory by CAS.

LOAD_VALUE unsigned int

Value to be loaded from memory as a result of atomic operation.

MANAGER unsigned int

ID of manager which initiated the transaction.

NS bool

Transaction is Non-Secure.

OPERAND_VALUE unsigned int

Incoming operand value provided by core.

OPERATION enum

Type of atomic operation.

PADDR unsigned int

Full physical address of TCM.

STORE_VALUE unsigned int

Value to be stored to memory as a result of atomic operation.

ATOMIC_START_ACCESS

Bus trace access for atomics. Fields:

ADDR unsigned int

The virtual address of the access.

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

COMPARE_VALUE unsigned int

Compare value for CAS.

NSDESC unsigned int

The security state of the access.

OPERAND_VALUE unsigned int

Operation's operand.

OPERATION enum

Operation type.

PADDR unsigned int

The physical address of the access.

PRIV bool

Is this a privileged access?.

ArchMsg.Error.exit_code

DISPLAY exit code: %{EXITCODE} component kind: %{KIND} reason: %{REASON}. Fields:

EXITCODE signed int

Exit code to be returned.

KIND string

Component kind that invoked the exit code trace.

REASON string

Reason why the exit code trace was invoked (optional).

ArchMsg.Info.invalid_region_programmed

DISPLAY MPU Stage %{STAGE:d}; Region ID %{REGION_ID:d} has base address greater than end address Base Addr=%{BASE_ADDR:x}, End Addr=%{END_ADDR:x}. Fields:

BASE_ADDR unsigned int

Base Address.

END_ADDR unsigned int

End Address.

REGION_ID unsigned int

Region ID.

STAGE unsigned int

MPU Stage.

ArchMsg.Warning.IgnoredWarmResetRequest

Event fires when software attempts to request warm reset through write to deprecated DBGPRCR.CWRR.

ArchMsg.Warning.ReportedRevisionRegisterRead

DISPLAY Reported revision and patch do not match that of the cpu. Fields:

ORIGINAL_VALUE unsigned int

The value the register would have based on the modelled revision.

REGISTER string

The name of the register that has been read.

REPORTED_VALUE unsigned int

The value the register has been overridden to.

ArchMsg.Warning.Unpredictably Indexed PM Event Register

TAGS **UNPREDICTABLE** DISPLAY Event %{IsDirect:(Selector value|Register)} %{SEL} is >= %{N}.
Fields:

IsDirect unsigned int

Direct Access rather than indirect.

N unsigned int

Number of Accessible Registers.

SEL unsigned int

Selector.

ArchMsg.Warning.cache_contents_unknown

DISPLAY %{SIDE:(D|I)}-Cache was enabled but was not invalidated since power-on: cache lines could contain **UNKNOWN** content. Fields:

REGIME unsigned int

Current translation regime.

SIDE unsigned int

0 d-side cache or unified cache, 1 i-side.

ArchMsg.Warning.dcimvac_matches_watchpoint

TAGS **UNPREDICTABLE** DISPLAY Watchpoints matching an AArch32 DCIMVA are implementation defined.

ArchMsg.Warning.decode_fieldmismatch

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with different fields %{FIELD1} and %{FIELD2}. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_frachitsnegative

TAGS **UNPREDICTABLE** DISPLAY VCVT Instruction is unpredictable with a negative number of fraction bits.

ArchMsg.Warning.decode_initblock

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable in an IT block.

ArchMsg.Warning.decode_initblocknotlast

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable in an IT block when not the last.

ArchMsg.Warning.decode_invalid_condition

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable when the condition is not AL.

ArchMsg.Warning.decode_invalidfieldcombination

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with the values in %{FIELD1} and %{FIELD2}. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_invalidvalue

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with the value in %{FIELD}. Fields:

FIELD enum

Field name.

ArchMsg.Warning.decode_mem_hint_unallocated

TAGS **UNPREDICTABLE** DISPLAY There is no memory hint allocated to this bit pattern.

ArchMsg.Warning.decode_registermatch

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with %{FIELD1} and %{FIELD2} as the same register. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_registermismatch

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with %{FIELD1} and %{FIELD2} as different registers. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_registeroutofrange

TAGS **UNPREDICTABLE** DISPLAY Floating-point instruction is unpredictable indexing beyond the end of the register bank.

ArchMsg.Warning.decode_sbzsbo

TAGS **UNPREDICTABLE** DISPLAY Reserved bits in the instruction are not canonical.

ArchMsg.Warning.decode_transactiontoobig

TAGS **UNPREDICTABLE** DISPLAY Attempt to perform a memory transaction of more than 128 bytes.

ArchMsg.Warning.decode_unpred_other

TAGS UNPREDICTABLE DISPLAY Instruction is unpredictable.

ArchMsg.Warning.decode_unpreduseofpc

TAGS UNPREDICTABLE DISPLAY Instruction is unpredictable when %{FIELD} is PC. Fields:

FIELD enum

Field name.

ArchMsg.Warning.decode_unpreduseofr13

TAGS UNPREDICTABLE DISPLAY Instruction is unpredictable when %{FIELD} is R13. Fields:

FIELD enum

Field name.

ArchMsg.Warning.decode_unpreduseofr13inreglist

TAGS UNPREDICTABLE DISPLAY Instruction is unpredictable with R13 in register-list.

ArchMsg.Warning.decode_useofd16orhigher

TAGS UNPREDICTABLE DISPLAY Attempt to use register D16 or higher where it is not allowed.

ArchMsg.Warning.decode_writebackandbaseinlist

TAGS UNPREDICTABLE DISPLAY Instruction is unpredictable with write back when the base register is in the transfer list.

ArchMsg.Warning.decode_zeroregistersinlist

TAGS UNPREDICTABLE DISPLAY Instruction is unpredictable when no registers are to be transferred.

ArchMsg.Warning.recursive_branch

An instruction is performing a branch that targets the same instruction. If this was intended then a WFI might be more effective DISPLAY recursive branch detected at PC=%{PC}.

PRIMARY KEY PC. Fields:

PC unsigned int

Address of instruction causing the branch.

ArchMsg.Warning.recursive_exception

An instruction has generated an exception that targets the same instruction. DISPLAY recursive %{TYPE} exception detected at PC=%{PC} (CPSR=%{CPSR}). PRIMARY KEY PC.

Fields:

CPSR unsigned int

Processor state.

PC unsigned int

Address of instruction causing the exception.

TYPE enum

Type of exception.

ArchMsg.Warning.reentrant_vector_catch

TAGS **UNPREDICTABLE** DISPLAY Vector catch debug event on Prefetch or Data abort vector %**{ADDR}**. Fields:

ADDR **unsigned int**
vector address.

ArchMsg.Warning.reserved_it_state

TAGS **UNPREDICTABLE** DISPLAY Execution with reserved IT state %**{ITSTATE}**. Fields:

ITSTATE **unsigned int**
reserved IT state.

ArchMsg.Warning.tlb_contents_unknown

Translation has been enabled without invalidating the TLB. DISPLAY Translation was enabled in the %**{PAS:(Secure|Non-secure|Root|Realm)}** EL%**{EL:d}** translation regime (vmid=%**{VMID:x}**) without invalidating %**{INVALIDITY:(|d-side |i-side |)}** entries since power-on: %**{|INVALIDITY:(|D||)}** TLB could contain **UNKNOWN** content. Fields:

EL **unsigned int**
Exception level of translation regime.

INVALIDITY **unsigned int**
Which TLB may be invalid (if TLBs are separate).

NON_SECURE **bool**
Whether the translation regime is non-secure.

PAS **unsigned int**
Physical address space of translation regime.

VMID **unsigned int**
Current VMID, if applicable.

ArchMsg.Warning.unknown_DLR_DSPSR

Exiting debug state with unknown DLR or DSPSR.

ArchMsg.Warning.unknown_ELR_SPSR

Returning from debug exception with unknown ELR or SPSR.

ArchMsg.Warning.unpred_FAR_on_LRCPC3_non_atomic_mem_access

TAGS **UNPREDICTABLE** Unpredictable FAR/FSTATUS on LRCPC3 pair instructions resulting in two separate memory accessesDISPLAY Unpredictable FAR on LRCPC3 non atomic memory access at %**{ADDR1}** with FSTATUS %**{FSTATUS1}** or %**{ADDR2}** with FSTATUS %**{FSTATUS2}**. Fields:

ADDR1 **unsigned int**
Possible FAR address 1.

ADDR2 **unsigned int**
Possible FAR address 2.

FSTATUS1 unsigned int

Possible fault status value 1.

FSTATUS2 unsigned int

Possible fault status value 2.

ArchMsg.Warning.unpred_async_tag_check_fail_on_sve_speculative_load_do_update_ffr

Unpredictable set to FALSE of the FFR element associated to SVE speculative load with an async tag check fail TAGS **UNPREDICTABLE**.

ArchMsg.Warning.unpredictable_a32_breakpoint

TAGS **UNPREDICTABLE** DISPLAY Unpredictable breakpoint on a A32 instruction at %{ADDR1} Offset %{OFFSET} BAS %{BAS} for a %{IS_ADDRESS_MISMATCH_BREAKPOINT:(Mismatch|Match)} Breakpoint number %{N:d} Breakpoint address %{ADDR2}. Fields:

ADDR1 unsigned int

Address.

ADDR2 unsigned int

Breakpoint address.

BAS unsigned int

BAS field.

IS_ADDRESS_MISMATCH_BREAKPOINT bool

Is a Mismatch breakpoint, not a Match breakpoint.

N unsigned int

Breakpoint number.

OFFSET unsigned int

Always has a value of zero.

ArchMsg.Warning.unpredictable_banked_register_access

TAGS **UNPREDICTABLE** DISPLAY Access to banked register (R=%{R:d} SYSm=%{SYSM:2x}) is unpredictable. Fields:

R unsigned int

R bit.

SYSM unsigned int

SYSm.

ArchMsg.Warning.unpredictable_ccfail_undef

TAGS **UNPREDICTABLE** DISPLAY An instruction %{OPCODE} at VA=%{VADDR} failed its condition codes check but would otherwise have caused a non-data-dependent trap or undefined exception. Behaviour is implementation-defined choice of exception or nop. Fields:

OPCODE unsigned int

opcode of the instruction.

VADDR unsigned int

Virtual address of the instruction being executed.

ArchMsg.Warning.unpredictable_ici_bits

TAGS **UNPREDICTABLE** DISPLAY ICI bits do not apply or are architecturally unpredictable for instruction.

ArchMsg.Warning.unpredictable_t32_breakpoint

TAGS **UNPREDICTABLE** DISPLAY Unpredictable breakpoint on a T32 instruction at %{ADDR1} Offset %{OFFSET} BAS %{BAS} for a %{IS_ADDRESS_MISMATCH_BREAKPOINT:(Mismatch|Match)} Breakpoint number %{N:d} Breakpoint address %{ADDR2}. Fields:

ADDR1 unsigned int

Address.

ADDR2 unsigned int

Breakpoint address.

BAS unsigned int

BAS field.

IS_ADDRESS_MISMATCH_BREAKPOINT bool

Is a Mismatch breakpoint, not a Match breakpoint.

N unsigned int

Breakpoint number.

OFFSET unsigned int

Indicates which halfword of a 32-bit Thumb instruction the breakpoint is set on. 0 for the first halfword or if the instruction is 16 bits wide, or 1 for the second halfword.

ArchMsg.Warning.unpredictable_t32_vector_catch

TAGS **UNPREDICTABLE** DISPLAY Unpredictable Vector catch exception on a T32 instruction at %{ADDR} Offset %{OFFSET}. Fields:

ADDR unsigned int

Address.

OFFSET unsigned int

Indicates which halfword of a 32-bit Thumb instruction the vector catch exception is set on. 0 for the first halfword or if the instruction is 16 bits wide, or 1 for the second halfword.

ArchMsg.Warning.unpredictable_watchpoint_far

TAGS **UNPREDICTABLE** DISPLAY The value for FAR/EDWAR for hitting this watchpoint may be anywhere in the range %{LOWER_BOUND} to %{UPPER_BOUND}. Fields:

LOWER_BOUND unsigned int

the lowest address accessed by the instruction that triggered the watchpoint.

UPPER_BOUND unsigned int

the highest watchpointed address accessed by that instruction.

ArchMsg.Warning.unpredictable_watchpoint_far_isscmp

DISPLAY The value for FAR/EDWAR for hitting this watchpoint may be anywhere in the range %{LOWER_BOUND} to %{UPPER_BOUND}. Fields:

LOWER_BOUND unsigned int

the lowest address accessed by the instruction that triggered the watchpoint.

UPPER_BOUND unsigned int

the highest watchpointed address accessed by that instruction.

ArchMsg.Warning.warning_AdvSIMDExpandImmUnexpectedZero

TAGS UNPREDICTABLE DISPLAY AdvSIMDExpandImm may treat this immediate value as UNPREDICTABLE.

ArchMsg.Warning.warning_ConditionalSMC

TAGS UNPREDICTABLE DISPLAY SMC instruction has UNPREDICTABLE effects when conditional and when combined with traps.

ArchMsg.Warning.warning_TLBID_out_of_range

TLBID out of range in TLBI according to the configuration of NIS, NVIS, NOS and NVOS. Fields:

NIS unsigned int

NIS.

NOS unsigned int

NOS.

NVIS unsigned int

NVIS.

NVOS unsigned int

NVOS.

TLBID unsigned int

The TLBID value used in the TLBI instruction.

ArchMsg.Warning.warning_TTL_or_TG_mismatches_current_translation_granule

TTL or TG mismatches current translation granule in TLBI VA/RVA/IPA/RIPA. Fields:

CURRENT_TG unsigned int

Current TG as indicated by TCR_ELx or VTCR_EL2.

TLBI_TG unsigned int

TG (or TTL[3:2]) used in TLBI instruction.

ArchMsg.Warning.warning_access_crosses_page_boundary

TAGS UNPREDICTABLE DISPLAY Access crosses page boundary.

ArchMsg.Warning.warning_access_crosses_page_boundary_has_fault

TAGS UNPREDICTABLE DISPLAY Access that crosses page boundary has fault(s).

ArchMsg.Warning.warning_access_crosses_page_boundary_has_fault_on_both_pages

TAGS **UNPREDICTABLE** DISPLAY Access that crosses page boundary has a fault on each page.

Fields:

ADDR1 unsigned int

FAR address on first page.

ADDR2 unsigned int

FAR address on second page.

FSTATUS1 unsigned int

Fault status on first page.

FSTATUS2 unsigned int

Fault status on second page.

ArchMsg.Warning.warning_access_crosses_page_boundary_spanning_different_memory

TAGS **UNPREDICTABLE** DISPLAY Access at address %{ADDR} crosses page from MEM_TYPE %{MEMTYPE_PAGE1} to %{MEMTYPE_PAGE2}. Fields:

ADDR unsigned int

address of access crossing page.

MEMTYPE_PAGE1 signed int

First Page's memory type (0-Normal, 1-Non-Normal).

MEMTYPE_PAGE2 signed int

Second Page's memory type (0-Normal, 1-Non-Normal).

ArchMsg.Warning.warning_access_wraps_around_memory

TAGS **UNPREDICTABLE** DISPLAY Access wraps around memory in %{WIDTH:d}bit mode.

Fields:

WIDTH unsigned int

Address width.

ArchMsg.Warning.warning_bcr_linking_status

TAGS **UNPREDICTABLE** DISPLAY DBGBCR.{BT2,BT} combination gives unpredictable behaviour.

ArchMsg.Warning.warning_bcr_mask_reserved

TAGS **UNPREDICTABLE** DISPLAY DBGBCR_ELn.MASK is set to a reserved value.

ArchMsg.Warning.warning_bvr_masked_bit_not_zero

TAGS **UNPREDICTABLE** DISPLAY DBGBVRn_EL1.VA contains a bit masked by DBGBCRn_EL1.MASK that is not zero.

ArchMsg.Warning.warning_ccsidr_unimplemented_level

TAGS **UNPREDICTABLE** DISPLAY CCSIDR read while CSSELR %{CSSELR:x} points to an unimplemented cache level. Fields:

CSSELR unsigned int

current effective value of CSSELR.

ArchMsg.Warning.warning_change_to_ns_when_tge_set

TAGS **UNPREDICTABLE** DISPLAY Attempting to change NS to 1 when HCR.TGE = 1.

ArchMsg.Warning.warning_contiguous_bit_check_abort

TAGS **UNPREDICTABLE** DISPLAY An abort occurred whilst attempting to check TLB entry is contiguous at %{ENTRY_ADDR}. Fields:

ENTRY_ADDR unsigned int

address of conflicting TLB entry.

ArchMsg.Warning.warning_contiguous_bit_error

TAGS **UNPREDICTABLE** DISPLAY TLB entry at %{ENTRY_ADDR} was not contiguous with entry at %{CONTIG_ADDR} Contents are %{ENTRY_DATA} but expected %{CONTIG_DATA}. Fields:

CONTIG_ADDR unsigned int

address of the TLB first read, with which this entry is expected to be contiguous.

CONTIG_DATA unsigned int

expected contents based on the entry at CONTIG_ADDR.

ENTRY_ADDR unsigned int

address of conflicting TLB entry.

ENTRY_DATA unsigned int

contents of conflicting TLB entry.

ArchMsg.Warning.warning_contiguous_bit_error_D128

TAGS **UNPREDICTABLE** DISPLAY TLB entry at %{ENTRY_ADDR} was not contiguous with entry at %{CONTIG_ADDR} Contents are %{ENTRY_DATA_HI} %{ENTRY_DATA_LO} but expected %{CONTIG_DATA}. Fields:

CONTIG_ADDR unsigned int

address of the TLB first read, with which this entry is expected to be contiguous.

CONTIG_DATA unsigned int

expected contents based on the entry at CONTIG_ADDR.

ENTRY_ADDR unsigned int

address of conflicting TLB entry.

ENTRY_DATA_HI unsigned int

Higher bit contents of conflicting TLB entry.

ENTRY_DATA_LO unsigned int

Lower bit contents of conflicting TLB entry.

ArchMsg.Warning.warning_cp10_cp11_mismatch

TAGS **UNPREDICTABLE** DISPLAY Unpredictable configuration of CP10 and CP11 controls at EL %{EL:d}. Fields:

EL unsigned int

EL owning the control.

ArchMsg.Warning.warning_cp10_cp11_reserved_value

TAGS **UNPREDICTABLE** DISPLAY Unpredictable configuration of CPACR CP10 ({CP10}) or CP11 ({CP11}). Fields:

CP10 unsigned int

CP10 access.

CP11 unsigned int

CP11 access.

ArchMsg.Warning.warning_csselr_level_out_of_range

TAGS **UNPREDICTABLE** DISPLAY CSSELR.Level out of range, written value {LEVEL} with only {IMPLEMENTED:d} levels implemented. Fields:

IMPLEMENTED unsigned int

number of cache levels implemented.

LEVEL unsigned int

written value of CSSELR.Level.

TYPE enum

The type of cache selected by CSSELR.InD and TnD.

ArchMsg.Warning.warning_debug_flow_control_bits_not_obeyed

TAGS **UNPREDICTABLE** DISPLAY Flow control bits for {REGISTER} were not obeyed, causing {ERROR}. Fields:

ERROR enum

type of error resulting.

REGISTER enum

register under consideration.

ArchMsg.Warning.warning_debug_register_access_during_reset

TAGS **UNPREDICTABLE** DISPLAY Unpredictable {IS_WRITE:(read|write)} access to debug register offset {OFFSET} during reset. Fields:

IS_WRITE unsigned int

Write Not Read.

OFFSET unsigned int

Register Offset.

ArchMsg.Warning.warning_decode_cps_inconsistent_fields

TAGS **UNPREDICTABLE** DISPLAY The mode setting fields in the CPS instruction are inconsistent.

ArchMsg.Warning.warning_decode_invalid_state

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable in current exception-level/security state.

ArchMsg.Warning.warning_default_cacheable_mmu_on

TAGS **UNPREDICTABLE** DISPLAY Default cacheable is enabled (HCR.DC = 1) while MMU is enabled.

ArchMsg.Warning.warning_default_cacheable_vmmu_off

TAGS **UNPREDICTABLE** DISPLAY Default cacheable is enabled (HCR.DC = 1) while HCR.VM = 0.

ArchMsg.Warning.warning_deprecated_wvr_bit_2_set

TAGS **UNPREDICTABLE** DISPLAY DBGWVR%{N:d}_EL1=%{WVR:h} has bit 2 set, deprecated in ARMv8. Fields:

N unsigned int

index.

WVR unsigned int

DBGWVR.

ArchMsg.Warning.warning_eret_while_software_step_active_pending

Missing ISB between setting MDSCR_EL1.SS and ERET TAGS **UNPREDICTABLE**.

ArchMsg.Warning.warning_exclusive_to_non_normal

TAGS **UNPREDICTABLE** DISPLAY Exclusive Access to Non-Normal Memory %{ADDR}. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_exclusive_to_non_writeback

TAGS **UNPREDICTABLE** DISPLAY Exclusive Access to Non-Writeback-Cacheable Memory %{ADDR}. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_execute_from_device_memory

TAGS **UNPREDICTABLE** DISPLAY An attempt was made to execute from Device memory.

ArchMsg.Warning.warning_illegal_cpsr_mode

TAGS **UNPREDICTABLE** DISPLAY Writing an illegal or unimplement mode (%{NEW_MODE:h}) to CPSR was unpredictable. Fields:

NEW_MODE unsigned int

New value of CPSR.M.

ArchMsg.Warning.warning_illegal_srs_mode

TAGS **UNPREDICTABLE** DISPLAY Illegal or **UNPREDICTABLE** mode (%{MODE:h}) used for SRS instruction. Fields:

MODE unsigned int

mode for Banked SP.

ArchMsg.Warning.warning_implementation_defined_read_debug_register_in_SCS

Read Access to Debug registers from the processor is IMPLEMENTATION DEFINED TAGS IMP_DEF. Fields:

PPB_OFFSET unsigned int
debug register offset.

ArchMsg.Warning.warning_implementation_defined_sequential_security_transitions_supported

The behavior of sequential instruction fetches that cross from non-secure to secure memory and contain an SG instruction is CONSTRAINED UNPREDICTABLE TAGS UNPREDICTABLE.

ArchMsg.Warning.warning_implementation_defined_stack_limit_check_supported

It is IMPLEMENTATION DEFINED whether stack pointer limit checking is performed for this instruction TAGS IMP_DEF. Fields:

ADDRESS unsigned int
Address of instruction.

OPCODE unsigned int
instruction opcode.

ArchMsg.Warning.warning_implementation_defined_taggednonsharedmemorytype

TAGS IMP_DEF DISPLAY Memory attribute has Tagged, non shared memory type; it is IMPLEMENTATION DEFINED whether the memory location is treated as Tagged or non-Tagged.

ArchMsg.Warning.warning_implementation_defined_write_debug_register_in_SCS

Write Access to Debug registers from the processor is IMPLEMENTATION DEFINED TAGS IMP_DEF. Fields:

DATA unsigned int
data attempted to be written.

PPB_OFFSET unsigned int
debug register offset.

ArchMsg.Warning.warning_invalid_tcr_granule

TAGS UNPREDICTABLE DISPLAY TCR.TG%{TG_ID:(0|1)} bits have been set to a granule size %{{REQUEST:d}}K not implemented - using %{{SUBSTITUTE:d}}K. Fields:

REQUEST signed int
page size requested (or 0 for reserved).

SUBSTITUTE signed int
best guess available page size.

TG_ID bool
bits TG0 or TG1.

ArchMsg.Warning.warning_load_multiple_user_registers_from_user_mode

TAGS UNPREDICTABLE DISPLAY An LDM(user registers) instruction executed from user mode.

ArchMsg.Warning.warning_mte_in_mem_attr_not_supported

TAGS UNPREDICTABLE DISPLAY Memory attribute %{ATTR} has Tagged memory type but Armv8.5-MTE is not implemented. Fields:

ATTR unsigned int

Memory attribute.

ArchMsg.Warning.warning_reserved_breakpoint_state_match

TAGS UNPREDICTABLE DISPLAY %{IS_BREAKPOINT:(Watchpoint|Breakpoint)} programmed with a reserved combination of HMC, SSC and %{IS_BREAKPOINT:(PAC|PMC)}. Fields:

IS_BREAKPOINT unsigned int

Is a breakpoint, not a watchpoint.

ArchMsg.Warning.warning_shareability

TAGS UNPREDICTABLE DISPLAY Unpredictable: combination of the 1st and 2nd stages of translation is Normal Inner Non-Cacheable, Outer Non-Cacheable.

ArchMsg.Warning.warning_software_step_set_while_enabled

TAGS UNPREDICTABLE DISPLAY MDSCR_EL1.SS set to 1 while software step debug exceptions are enabled.

ArchMsg.Warning.warning_thumb_instruction_wraps_around_memory

TAGS UNPREDICTABLE DISPLAY Thumb instruction wraps around memory.

ArchMsg.Warning.warning_ttbr_sbz_bits_are_not_zero

TAGS UNPREDICTABLE DISPLAY %{TTBR}[%{MSB:d}:0] should be zero, but are not. Fields:

MSB unsigned int

x-1.

TTBR enum

which TTBR register.

ArchMsg.Warning.warning_unaligned_address_dbgdtrrx_write

TAGS UNPREDICTABLE DISPLAY Unaligned address %{ADDR} in X0/R0 when DBGDTRRX written in memory access mode. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unaligned_address_dbgdtrtx_read

TAGS UNPREDICTABLE DISPLAY Unaligned address %{ADDR} in X0/R0 when DBGDTRTX read in memory access mode. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unaligned_to_device

TAGS UNPREDICTABLE DISPLAY Unaligned Access to Device Memory %{ADDR}. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unaligned_to_strongly_ordered

TAGS **UNPREDICTABLE** DISPLAY Unaligned Access to Strongly Ordered Memory %{ADDR}.

Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unknown_sau_rnr

TAGS **UNPREDICTABLE** DISPLAY SAU_RNR was set to an unsupported value.

ArchMsg.Warning.warning_unpredictable_AIRCR_PRIS_and_BFHFNMINs

TAGS **UNPREDICTABLE** DISPLAY The effect of setting both AIRCR.BFHFNMINs and AIRCR.PRIS to 1 is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_AIRCR_VECTCLRACTIVE_when_not_in_debug

TAGS **UNPREDICTABLE** DISPLAY The effect of writing a 1 to AIRCR.VECTCLRACTIVE if the processor is not halted in Debug state is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_AIRCR_VECTRESET_and_SYSRESETREQ

TAGS **UNPREDICTABLE** DISPLAY When the processor is halted in Debug state, if a write to the register writes a 1 to both VECTRESET and SYSRESETREQ, the behavior is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_AIRCR_VECTRESET_when_not_in_debug

TAGS **UNPREDICTABLE** DISPLAY The effect of writing a 1 to AIRCR.VECTRESET if the processor is not halted in Debug state is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_AIRCR_incorrect_VKEY

TAGS **UNPREDICTABLE** DISPLAY The value 0x05FA must be written to AIRCR.VKEY, otherwise the register write is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_EXC_RETURN_Reserved_Bit

EXC_RETURN[23:7] are reserved with the special condition that all bits should be written as one. Values other than all 1s are **UNPREDICTABLE**. TAGS **UNPREDICTABLE**. Fields:

EXC_RETURN unsigned int

Special PC value.

SBO_MASK unsigned int

Which bits Should Be One to not UNPRED.

ArchMsg.Warning.warning_unpredictable_PMEVTYPEPERn_ELO_VS_reserved_value

TAGS **UNPREDICTABLE** DISPLAY Unpredictable reserved value of PMEVTYPER<n>_ELO.VS.

ArchMsg.Warning.warning_unpredictable_change_to_DEMCR_MON_STEP_at_insufficient_priority

TAGS **UNPREDICTABLE** DISPLAY The effect of changing DEMCR.MON_STEP at an execution priority that is lower than the priority of the DebugMonitor exception is **CONSTRAINED UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_change_to_priority_of_active_exception

TAGS UNPREDICTABLE DISPLAY Changing the priority of an active exception is UNPREDICTABLE.

ArchMsg.Warning.warning_unpredictable_clear_lspact

TAGS UNPREDICTABLE DISPLAY UNPREDICTABLE state on context switch. Fields:

DRVEXC_TAKEN unsigned int

Derived exception taken, not pended.

REACHED_LAST_STORE unsigned int

Only the last faulted.

ArchMsg.Warning.warning_unpredictable_dbg_exit_unaligned_dlr

TAGS UNPREDICTABLE DISPLAY Debug exit to AArch32 with DLR[0] set to 1. PC[0] can be set to 0 or to DLR[0].

ArchMsg.Warning.warning_unpredictable_exception_catch

TAGS UNPREDICTABLE DISPLAY Generation of an exception catch event is unpredictable.

ArchMsg.Warning.warning_unpredictable_exception_return_inconsistent_state

TAGS UNPREDICTABLE DISPLAY state on exception return is unpredictable.

ArchMsg.Warning.warning_unpredictable_exception_return_instruction

TAGS UNPREDICTABLE DISPLAY exception return instruction is unpredictable.

ArchMsg.Warning.warning_unpredictable_in_debug_state

TAGS UNPREDICTABLE DISPLAY Instruction is UNPREDICTABLE when executed in debug state.

ArchMsg.Warning.warning_unpredictable_mair_encoding

TAGS UNPREDICTABLE DISPLAY Memory attribute %{ATTR} has invalid MAIR encoding; this may lead to unpredictable results. Fields:

ATTR unsigned int

Memory attribute.

ArchMsg.Warning.warning_unpredictable_pmu_counter_access

TAGS UNPREDICTABLE DISPLAY Access to PMU counters from non-secure EL0 or EL1 is UNPREDICTABLE with MDCR_EL2.HPMN set to 0.

ArchMsg.Warning.warning_unpredictable_prioritization_breakpoint_match_vector_catch

TAGS UNPREDICTABLE DISPLAY Two events with the same priority occurred at the same instruction: Address Matching Vector Catch debug event, and Breakpoint debug event. It is constrained unpredictable which is taken.

ArchMsg.Warning.warning_unpredictable_stack_selection

TAGS UNPREDICTABLE DISPLAY UNPREDICTABLE state on context switch.

ArchMsg.Warning.warning_unpredictable_tsize_out_of_range

TAGS UNPREDICTABLE DISPLAY %{TCR}.T%{TTBR:d}SZ (%{TSIZE:d} bits) is out of range. Must be between 25 and %{TSIZE_MAX:d}. Fields:

TCR enum

Which TCR.

TSize signed int

Value of TSize.

TSize_MAX signed int

The maximum allowed value.

TTBR unsigned int

Which TTBR region.

ArchMsg.Warning.warning_unpredictable_unaligned_pc_as_base_register

TAGS **UNPREDICTABLE** DISPLAY The PC value must be word-aligned, otherwise the behavior of the instruction is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_unaligned_pop_stack

TAGS **UNPREDICTABLE** DISPLAY **CONSTRAINED UNPREDICTABLE** unaligned stack frame `%{FRAME_ADDR:x}` in exception-return. Fields:

FRAME_ADDR unsigned int

unaligned stack frame address.

ArchMsg.Warning.warning_unpredictable_unaligned_to_non_writeback

TAGS **UNPREDICTABLE** DISPLAY Unaligned Access to Non-Writeback-Cacheable Memory.

ArchMsg.Warning.warning_unpredictable_vmsa_memattrib

TAGS **UNPREDICTABLE** DISPLAY Unpredictable vmsa memory attribute with texcb value programmed to `%{TEXCB}` when tex remap is `%{TEX_REMAP}`. Fields:

TEXCB unsigned int

texcb value.

TEX_REMAP bool

use tex remap.

ArchMsg.Warning.warning_unpredictable_vtcr_t0sz_sl0

TAGS **UNPREDICTABLE** DISPLAY The combination of VTCR.TOSZ=`%{TOSZ:d}` and VTCR.SL0=`%{SL0:d}` is unpredictable. Fields:

SL0 signed int

VTCR.SL0, starting level for stage 2 translation table walks.

TOSZ signed int

VTCR.TOSZ, required input address range.

ArchMsg.Warning.warning_unpredictable_write_DHCSR

TAGS **UNPREDICTABLE** DISPLAY **UNPREDICTABLE** set of bit-changes in DHCSR. Fields:

NEW unsigned int

data attempted to be written.

OLD unsigned int

previous value.

ArchMsg.Warning.warning_unpredictable_write_SHCSR

TAGS UNPREDICTABLE DISPLAY UNPREDICTABLE setting a pending bit to 1 for an exception with priority \geq execution priority.

ArchMsg.Warning.warning_unsupported_access_to_memory_mapped_register

TAGS UNPREDICTABLE DISPLAY Access to memory mapped register is unsupported.

ArchMsg.Warning.warning_user_jmcr_access

TAGS UNPREDICTABLE DISPLAY User %{WRITE:(write|read)} access to JMCR is unpredictable. Fields:

WRITE unsigned int
access is write.

ArchMsg.Warning.warning_wcr_mask_and_bas

TAGS UNPREDICTABLE DISPLAY DBGWCR_EL1.MASK is non-zero and BAS is not 0xff.

ArchMsg.Warning.warning_wcr_mask_reserved

TAGS UNPREDICTABLE DISPLAY DBGWCR_ELn.MASK is set to a reserved value.

ArchMsg.Warning.warning_wcr_non_configuous_bas

TAGS UNPREDICTABLE DISPLAY DBGWCR_EL1.BAS has non-contiguous set of ones.

ArchMsg.Warning.warning_wvr_masked_bit_not_zero

TAGS UNPREDICTABLE DISPLAY DBGWVRn_EL1.VA contains a bit masked by DBGWCRn_EL1.MASK that is not zero.

ArchMsg.Why.why_branch_target_exception

A branch target exception has occurred. DISPLAY branch target exception: target pc=%{PC}, target opcode=%{OPCODE}. PRIMARY KEY PC. Fields:

BTYPE unsigned int
pstate.BTYPE at time branch taken.

OPCODE unsigned int
opcode at branch target.

PC unsigned int
Virtual Address of branch target.

ArchMsg.Why.why_illegal_state

An illegal mode change has occurred. DISPLAY illegal mode change: %{MESSAGE}. Fields:

MESSAGE string
The message.

BRANCH_MISPREDICT

Simulating branch mispredict. Fields:

PC unsigned int
Origin address (or 0 if unavailable).

BRA_DIR

Direct branches, to immediate address. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISSET enum

The instruction set of the branch instruction.

IS_COND bool

Indicates if this is a conditional branch.

IS_HINTED bool

Whether the conditional instruction is hinted.

IS_ISB bool

Whether the instruction is ISB.

IS_LINK bool

Indicates if this is a branch with link.

PC unsigned int

The address of the branch instruction.

TARGET_ISET enum

The instruction set after the branch.

TARGET_PC unsigned int

The address the instruction branches to.

BRA_INDIR

Indirect branches, perhaps to a register. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISSET enum

The instruction set of the branch instruction.

IS_COND bool

Indicates if this is a conditional branch.

IS_HINTED bool

Whether the conditional instruction is hinted.

IS_ISB bool

Whether the instruction is ISB.

IS_LINK bool

Indicates if this is a branch with link.

PC unsigned int

The address of the branch instruction.

TARGET_ISET enum

The instruction set after the branch.

TARGET_PC unsigned int

The address the instruction branches to.

BRA_INDIR_CCFAIL

Indirect branches fail, perhaps to a register.

BRA_RET_CCFAIL

Branch return fail, perhaps to a register.

CACHE_MAINTENANCE_OP

Cache Maintenance Operation. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

DATA unsigned int

Specified MVA or Set/way.

FUNCTION enum

Clean / Invalidate.

SCOPE enum

Affected region.

SIDE enum

Inst / Data.

CCFAIL

Conditional instruction condition check fail. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CCFAIL_UNC

Unconditional instruction condition check fail. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CCPASS

Conditional instruction condition check pass. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CCPASS_UNC

Unconditional instruction condition check pass. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CODE_CACHE_MAINT

Code cache maintenance. Fields:

END_INCL unsigned int

Inclusive end address for VA or PA requests.

NS bool

Non-secure world for PA.

START unsigned int

Start address for VA or PA requests.

TYPE enum

Request type.

COMPILE_BLOCK_END

Last instruction of basic block translated. Fields:

VADDR unsigned int

Address of next instruction after this basic block.

COMPILE_BLOCK_START

First instruction of basic block translated. Fields:

VADDR unsigned int

Address of first instruction in basic block.

COMPILE_INST

ARM instruction compiled. Fields:

DISASS string

Disassembly of the instruction.

ISSET enum

The instruction set of this instruction.

ITSTATE unsigned int

The ITSTATE current for the instruction.

OPCODE unsigned int

The opcode of the instruction.

PC unsigned int

The address of the instruction.

SIZE unsigned int

The size of the instruction in bytes.

CONTEXTIDR

Context ID Register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

NS bool

Secure or nonsecure banked register is accessed.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The new value written.

CONTEXT_SYNC

Called for every context synchronization event. Fields:

ARCH_CSE bool

Architectural Context Synchronization Event.

PC unsigned int

The address of the synchronization instruction.

SYSREG_SYNC bool

System register synchronization event.

CORE_ENDIAN

Core BE8 Big-Endian state changed. Fields:

BE8 bool

Core BE8 Big-Endian state.

CORE_INFO

Static processor attributes. Only triggered by a call to DumpState(). Fields:

ARCH_PROFILE enum

The architecture profile of the core.

CLUSTER_ID unsigned int

The cluster ID of this processor.

CORE_NUM unsigned int

The number of this core in an MP processor.

DETAILED_CORE_FEATURES enum

Detailed core features as a bitfield. Enum values are bitmasks to extract the info from the field.

FPU_VERSION enum

The VFP version implemented by the core.

MEM_ARCH enum

The memory architecture of the core.

MIDR_EL1 unsigned int

MIDR_EL1 value.

NUM_CORES unsigned int

The number of cores in this MP processor.

QUANTUM_SIZE unsigned int

The default quantum size of the core (in ticks).

SECURITY_FEATURES bool

Does the core have security features?.

CORE_LOADS

Processor load accesses. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ACQREL enum

Is this an acquire/release.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DATA unsigned int

The data read or written.

ELEMENT_SIZE unsigned int

Width of each element.

ENDIANNESS enum

Byte ordering of this access.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

LOCK enum

Normal, exclusive or locked access.

MECID unsigned int

Memory Encryption Context ID, if applicable.

MECID2 unsigned int

Memory Encryption Context ID of the second page, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PAGE_SIZE unsigned int

Page size as $\log_2(\text{size in bytes})$.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK, 2=Exclusive Failed, 3=Unsupported.

SIZE unsigned int

Width of the access in bytes. Only required if DATA is not traced.

TRANS bool

Is this a translated access.

VADDR unsigned int

The virtual address of the access.

CORE_REGS

Changes of the core registers R0 to R14. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number, 0 to 14.

MODE enum

Bank of the register accessed.

OLD_VALUE unsigned int

The old value overwritten.

PHYS_ID enum

The physical register accessed.

VALUE unsigned int

The new value written to the register.

CORE_REGS64

Changes of the core registers X0..X30, SP_ELn. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ID enum

The register number, 0-30 for X0-X30, >=32 for SP_ELn.

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

CORE_REGS64_READ

General purpose AArch64 register has been read X0..X30, SP_ELn. Fields:

ID enum

The register number, 0-30 for X0-X30, >=32 for SP_ELn.

VALUE unsigned int

The value read from the register.

CORE_STALLED_TRANSITION

Detect when a core is prevented from progressing PChannel states. Fields:

CURRENT_STATE enum

Current power mode of core.

REASON string

The reason a transition was stalled, and what must be done to unblock it.

TARGET_STATE enum

Desired power mode of core after transition.

CORE_STORES

Processor store accesses. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ACQREL enum

Is this an acquire/release.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DATA unsigned int

The data read or written.

ELEMENT_SIZE unsigned int

Width of each element.

ENDIANNESS enum

Byte ordering of this access.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

LOCK enum

Normal, exclusive or locked access.

MECID unsigned int

Memory Encryption Context ID, if applicable.

MECID2 unsigned int

Memory Encryption Context ID of the second page, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PAGE_SIZE unsigned int

Page size as $\log_2(\text{size in bytes})$.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK, 2=Exclusive Failed, 3=Unsupported.

SIZE unsigned int

Width of the access in bytes. Only required if DATA is not traced.

TRANS bool

Is this a translated access.

VADDR unsigned int

The virtual address of the access.

CP14_READ

System Coprocessor register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP14 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc1 unsigned int

opcode 1.

opc2 unsigned int

opcode 2.

CP14_WRITE

System Coprocessor register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP14 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc1 unsigned int

opcode 1.

opc2 unsigned int

opcode 2.

CP15_READ

System Coprocessor register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP15 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc1 unsigned int

opcode 1.

opc2 unsigned int

opcode 2.

CP15_READ64

System Coprocessor register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP15 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc unsigned int

opcode 1.

CP15_WRITE

System Control Coprocessor register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP15 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc1 unsigned int

opcode 1.

opc2 unsigned int
opcode 2.

CP15_WRITE64

System Coprocessor register write. Fields:

CORE_NUM unsigned int
Core number in a multi processor.

CRm unsigned int
CRm.

NS bool
Secure or nonsecure banked register is accessed.

REG_NAME string
Name of the CP15 register accessed.

SECURITY_STATE enum
The current security state.

UNDEF bool
The register accessed is undefined.

UPDATED_VALUE unsigned int
Updated value of the register now it has been written.

VALUE unsigned int
The new value written.

opc unsigned int
opcode 1.

CPSR

CPSR change. (In AArch64 PSTATE represented in SPSR format). Fields:

CORE_NUM unsigned int
Core number in a multi processor.

OLD_VALUE unsigned int
The old CPSR value.

UNKNOWN unsigned int
Bits within the register that have unknown value.

VALUE unsigned int
The new CPSR value.

CRYPTO_SPEC

Every crypto instruction speculatively executed.

CorePowerStateChange

Core power-up state has changed. DISPLAY Core powered %{PU:{up|down}}. Fields:

PU bool

Whether the core powered up.

DATA_CACHE_SET_ALLOC_TAGS

Set allocation tags from DC GVA instruction. Fields:

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

NSDESC unsigned int

The security state of the access.

PADDR unsigned int

The physical address of the access.

PAS enum

The physical address space of the page.

RESPONSE enum

0=Aborted, 1=OK.

SIZE unsigned int

The size of memory in bytes.

TAG unsigned int

The value of the tag being set.

TAGGING_TYPE enum

Type of Memory tagging (Physical or Virtual).

TAG_VADDR unsigned int

The virtual address of the Virtual Tag if Virtual tagging is used; otherwise 0.

VADDR unsigned int

Virtual address used for setting allocation tags.

DATA_CACHE_ZERO

Zero and invalidate cache line from DC ZVA instruction. Fields:

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

NSDESC unsigned int

The security state of the access.

PADDR unsigned int

The physical address of the access.

PADDR_TAG unsigned int

If tag checked, the physical address tag for the access.

PAS enum

The physical address space of the page.

RESPONSE enum

0=Aborted, 1=OK.

SIZE unsigned int

The size of zero'd memory in bytes.

TAG_CHECKED bool

Is this access tag checked.

VADDR unsigned int

Virtual address of the zero'd cache line.

DEBUG_EVENT

Hardware debug support event. Fields:

EVENT enum

Description of event.

VALUE unsigned int

data value.

DEBUG_TARGET_ISET

Hardware debug target ISA. Fields:

TARGET_ISET enum

The target instruction set of the debug state.

DMI_ALLOCATE

DMI allocated. Fields:

ACCESS_DIRECTION enum

READ / WRITE.

PADDR unsigned int

The physical address of the DMI.

PTR unsigned int

DMI range pointer.

SIDE enum

DSIDE / ISIDE.

SIZE unsigned int

The size of DMI.

VADDR unsigned int

The virtual address of the DMI..

DMI_HIT

DMI hit, access used DMI (only fire for small DMI regions i.e. less than a full page). Fields:

ACCESS_DIRECTION enum

READ / WRITE.

PADDR unsigned int

The physical address of the DMI.

SIDE enum

DSIDE / ISIDE.

SIZE unsigned int

The size of the DMI.

VADDR unsigned int

The virtual address of the DMI..

DMI_REVOKE

DMI revoked. Fields:

ACCESS_DIRECTION enum

READ / WRITE.

PADDR unsigned int

The physical address of the DMI.

SIDE enum

DSIDE / ISIDE.

SIZE unsigned int

The size of DMI.

VADDR unsigned int

The virtual address of the DMI..

END_COMPILE

Compilation end. Fields:

END_OF_PAGE_COUNT unsigned int

Number of basic blocks exited due to page boundary since START_COMPILE.

FETCHFAIL_COUNT unsigned int

Number of basic blocks exited due to ifetch failure START_COMPILE.

INST_COUNT unsigned int

Number of instructions compiled since START_COMPILE.

NONSEQ_COUNT unsigned int

Number of basic blocks exited due to non-sequential instructions since START_COMPILE.

PAGE_STRADDLE_COUNT unsigned int

Number of basic blocks exited due to unaligned instructions crossing page since START_COMPILE.

ESB

Error Synchronization Barrier hit. Fields:

ASYNC bool

ESB is async.

IMPLICIT bool

ESB is implicit.

EXCEPTION

Exception taken. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ESR unsigned int

The value assigned to ESR when exception is taken.

ESR_EC unsigned int

The Value assigned to ESR.EC when exception is taken.

IS_PHYSICAL bool

Physical or Virtual exception.

IS_TRAP bool

Exception coming from a trap.

LR unsigned int

The value assigned to the link register.

PC unsigned int

The location where the exception occurred.

PREFERRED_RETURN unsigned int

The preferred return address for the exception.

TAKEN_LOCALLY bool

Whether exception is taken locally(same exception level).

TARGET_ISET enum

The instruction set of the exception handler code.

TARGET_PC unsigned int

The address the exception branches to.

VECTOR enum

The exception vector.

VECTOR_OFFSET unsigned int

The offset of an exception vector from the base address.

EXCEPTION_END

Every exception completed.

EXCEPTION_INFO

Trace human-readable reason of the current exception for data/instruction aborts, floating point exceptions and unknown exceptions. Fields:

DESCRIPTION string

Reason for current exception.

ESR_EC unsigned int

The value of ESR.EC when exception is taken.

EXCEPTION_RAISE

Every exception raised.

EXCEPTION_RETURN

Branches on leaving exception. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISSET enum

The instruction set of the branch instruction.

IS_COND bool

Indicates if this is a conditional branch.

IS_HINTED bool

Whether the conditional instruction is hinted.

IS_ISB bool

Whether the instruction is ISB.

IS_LINK bool

Indicates if this is a branch with link.

PC unsigned int

The address of the branch instruction.

TARGET_ISET enum

The instruction set after the branch.

TARGET_PC unsigned int

The address the instruction branches to.

EXCEPTION_RETURN_PREBRANCH

Exception return event that must occur before PC and processor state have updated.

EXCEPTION_START

Every exception started.

FIQ_TAKEN

FIQ taken exception.

FP_EXCEPTIONS_SIGNALLED

Trace which FP exceptions were signalled as a part of the FP operations performed by an instruction. Fields:

FP_EXCEPTIONS_SIGNALLED_MASK unsigned int

Indicates which FP exceptions were signalled by an instruction. This uses the same encoding as the cumulative floating-point exception bitfields in FPSR.

FP_STATE

Floating point state. Fields:

FPSR unsigned int

FPSR register value.

MASK unsigned int

The mask indicating updated cumulative bits.

FREQ_CHANGED

Core frequency changed. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The instruction count of this core.

NEW_FREQ signed int

The new frequency of this core (expressed in Hz).

OLD_FREQ signed int

The old frequency of this core (expressed in Hz).

HLT

HLT instruction occurred. Fields:

IMM16 unsigned int

The 16-bit immediate value encoded in HLT instruction.

INST

Every instruction executed. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DEBUG_STATE bool

The instruction is executed in debug state.

DISASS string

Disassembly of instruction.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISET enum

The current instruction set.

ITSTATE unsigned int

The current ITSTATE.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

MODE enum

The mode the core is in.

NS unsigned int

The core's non-secure bit.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

OPCODE unsigned int

The opcode of the instruction.

PADDR unsigned int

The physical address of the instruction.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the instruction.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the page.

PC unsigned int

The address of the instruction.

SECURITY_STATE enum

The core's security state.

SIZE unsigned int

The size of the instruction in bytes.

INST_END

Every instruction completed.

INST_START

Every instruction started. Fields:

DEBUG_STATE **bool**

The instruction is executed in debug state.

ISET **enum**

The current instruction set.

MODE **enum**

The mode the core is in.

NS **enum**

The current Secure State.

PC **unsigned int**

The address of the conditional instruction.

SECURITY_STATE **enum**

The current Security State.

INST_STRADDLE

Instruction straddles boundary. Fields:

DEBUG_STATE **bool**

The instruction is executed in debug state.

ISET **enum**

The current instruction set.

MODE **enum**

The mode the core is in.

NS **enum**

The current Secure State.

PC **unsigned int**

The address of the conditional instruction.

SECURITY_STATE **enum**

The current Security State.

IRQ_TAKEN

IRQ taken exception.

ISB

ISB instruction occurred. Fields:

PC **unsigned int**

The address of the ISB instruction.

LOAD_MULTIPLE_REGS_LIST

List of destination registers (not source) of a multiple registers load instruction. Fields:

REG string

Register to add into the list.

LOCAL_MONITOR

Local monitor activity. Fields:

PADDR unsigned int

Local Monitor Address.

State enum

State of the monitor (Open/Exclusive).

MAPPING_CREATE

A memory mapping has been created. This is an implementation detail of the model and so may change without notice. Fields:

ASID unsigned int

The ASID used during translation, or 0 if no ASID was used.

IS_VALID bool

Whether the mapping is valid or not.

PADDR unsigned int

Base physical address of region, or 0 if not known.

SIDE enum

I-side or D-side.

SIZE unsigned int

Size of the page, or 0 if not known.

VADDR unsigned int

Base virtual address of region.

VMID unsigned int

The VMID used during translation, or 0 if no VMID was used.

MAPPING_DESTROY

A memory mapping has been destroyed. This is an implementation detail of the model and so may change without notice. Fields:

ASID unsigned int

The ASID used during translation.

CAUSE enum

The cause of invalidation.

PADDR unsigned int

Base physical address of page.

SIDE enum

I-side or D-side.

SIZE **unsigned int**

Size of the page.

VADDR **unsigned int**

Base virtual address of page.

VMID **unsigned int**

The VMID used during translation.

MEMMAP_DEBUG_READ

Memory mapped reads to the debug registers. Fields:

ADDR **unsigned int**

address.

CORE_NUM **unsigned int**

Core number in a multi processor.

EXT **bool**

Whether access is from an external device (such as the DAP).

NS **enum**

Secure state of the access.

REG_NAME **string**

Name of the debug register accessed.

UNDEF **bool**

The register accessed is undefined.

VALUE **unsigned int**

The value read.

MEMMAP_DEBUG_WRITE

Memory mapped writes to the debug registers. Fields:

ADDR **unsigned int**

address.

CORE_NUM **unsigned int**

Core number in a multi processor.

EXT **bool**

Whether access is from an external device (such as the DAP).

NS **enum**

Secure state of the access.

REG_NAME **string**

Name of the debug register accessed.

UNDEF **bool**

The register accessed is undefined.

VALUE unsigned int

The new value written.

MMU_D128_TTB_READ

This event is triggered by reads caused by a D128 translation table walk. Fields:

ABORTED bool

Was the walk successful.

DATA_HIGH unsigned int

The data upper bits read.

DATA_LOW unsigned int

The data lower bits read.

IPA unsigned int

For stage 1, the IPA of the read.

LEVEL unsigned int

Translation table level.

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

OUTADDR unsigned int

The output physical address.

PADDR unsigned int

The physical address of the read.

PASmem enum

Physical Address Space of the accessed memory.

PASreq enum

Physical Address Space of the requested lookup.

REGIME_EL enum

Entry matches in this translation regime.

REQUESTER enum

The requester of this table walk.

SIDE enum

Inst / Data.

STAGE unsigned int

Translation stage.

MMU_D128_TTB_WRITE

This event is triggered by writes caused by a D128 translation table walk. Fields:

ABORTED bool

Was the walk successful.

DATA_HIGH unsigned int

The data upper bits written.

DATA_LOW unsigned int

The data lower bits written.

IPA unsigned int

For stage 1, the IPA of the write.

IS_HACDBS_ACCESS bool

If the access was due to HACDBS_CLEAN.

LEVEL unsigned int

Translation table level.

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

OUTADDR unsigned int

The output physical address.

PADDR unsigned int

The physical address of the write.

PASmem enum

Physical Address Space of the accessed memory.

PASreq enum

Physical Address Space of the requested lookup.

REGIME_EL enum

Entry matches in this translation regime.

SIDE enum

Inst / Data.

STAGE unsigned int

Translation stage.

MMU_TRANS

Address translation information. Fields:

ASID unsigned int

Address space identifier.

CORE_NUM unsigned int

Core number in a multi processor.

Hyp bool

Entry matches in Hyp state only.

INNERCACHE_RA bool

Is the inner cache allocate on read.

INNERCACHE_TRANSIENT bool

Is the inner write-through transient.

INNERCACHE_TYPE enum

Inner Caching scheme (NC/MB/WA).

INNERCACHE_WA bool

Is the inner cache allocate on write.

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MEMTYPE enum

Memory type.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC enum

Is secure side supposed to access secure or nonsecure memory.

OUTERCACHE_RA bool

Is the outer cache allocate on read.

OUTERCACHE_TRANSIENT bool

Is the outer write-through transient.

OUTERCACHE_TYPE enum

Outer Caching scheme (NC/MB/WA).

OUTERCACHE_WA bool

Is the outer cache allocate on write.

PADDR unsigned int

Physical address of the access.

PAGESIZE unsigned int

Page size as log2(size).

PAS enum

PAS supposed to be accessed is secure or nonsecure memory or Root/Realm if RME is implemented.

SH enum

Shareability.

SIDE enum

Inst / Data.

STAGE1_PERM unsigned int

Stage 1 permissions, [5:3] Privileged access (XWR) [2:0] User access (XWR).

STAGE2_PERM unsigned int

Stage 2 permission mask [5:3] Privileged access (XWR) [2:0] User access (XWR).

VADDR unsigned int

Virtual address of the access.

VMID unsigned int

Virtual machine identifier.

XS bool

XS Attribute.

nG enum

Flag indicating whether ASID will be matched.

MMU_TTB_READ

This event is triggered by reads caused by a translation table walk. Fields:

ABORTED bool

Was the walk successful.

DATA unsigned int

The data read.

DESC_KIND enum

The encoding scheme of the descriptors used for this walk.

IPA unsigned int

For stage 1, the IPA of the read.

LEVEL unsigned int

Translation table level.

LPAE bool

Is this for an LPAE translation (*DEPRECATED*, use DESC_KIND).

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSmem bool

Non secure memory access.

NSreq bool

Non secure request.

OUTADDR unsigned int

The output physical address.

PADDR unsigned int

The physical address of the read.

PASmem enum

Physical Address Space of the accessed memory.

PASreq enum

Physical Address Space of the requested lookup.

REGIME_EL enum

Entry matches in this translation regime.

REQUESTER enum

The requester of this table walk.

SIDE enum

Inst / Data.

STAGE unsigned int

Translation stage.

MMU_TTB_WRITE

This event is triggered by writes caused by a translation table walk. Fields:

ABORTED bool

Was the walk successful.

DATA unsigned int

The data written.

DESC_KIND enum

The encoding scheme of the descriptors used for this walk.

IPA unsigned int

For stage 1, the IPA of the write.

IS_HACDBS_ACCESS bool

If the access was due to HACDBS_CLEAN.

LEVEL unsigned int

Translation table level.

LPAE bool

Is this for an LPAE translation (*DEPRECATED*, use DESC_KIND).

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSmem bool

Non secure memory access.

NSreq bool

Non secure request.

OUTADDR unsigned int

The output physical address.

PADDR unsigned int

The physical address of the write.

PASmem enum

Physical Address Space of the accessed memory.

PASreq enum

Physical Address Space of the requested lookup.

REGIME_EL enum

Entry matches in this translation regime.

SIDE enum

Inst / Data.

STAGE unsigned int

Translation stage.

MODE_CHANGE

Mode change. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

MODE enum

The new mode.

OLD_MODE enum

The old mode.

MPU_TRANS

Address translation information. Fields:

ACCESS_STATUS enum

Access status(0xAA/0xBB/0xDD/0xEE=overlap/bg hit/default/bg fault).

CORE_NUM unsigned int

Core number in a multi processor.

EXEC_PERM enum

Execution Permission.

INNERCACHE_RA bool

Is the inner cache allocate on read.

INNERCACHE_TYPE enum

Inner Caching scheme (NC/MB/WA).

INNERCACHE_WA bool

Is the inner cache allocate on write.

MEMTYPE enum

Memory type.

NSDESC enum

Is non secure access.

OUTERCACHE_RA bool

Is the outer cache allocate on read.

OUTERCACHE_TYPE enum

Outer Caching scheme (NC/MB/WA).

OUTERCACHE_WA bool

Is the outer cache allocate on write.

PADDR unsigned int

Address of the access.

PAGESIZE unsigned int

Page size as log2(size).

PAS enum

PAS accessed is secure or nonsecure memory.

READ_PERM enum

Read Permission.

REG_NUM unsigned int

MPU region-number 0-31.

SH enum

Shareability.

SIDE enum

Inst / Data.

STAGE unsigned int

The translation stage.

WRITE_PERM enum

Write Permission.

NOT_TAG_CHECKED_INST

MTE Tag not checked for current instruction.

PERIODIC

Called for every quantum. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The instruction count of this CPU.

PC unsigned int

The address of the next instruction to be executed on this CPU.

PMU_COUNTER_OVERFLOW

PMU counter overflow. Fields:

COUNTER_GROUP enum

Counter group.

COUNTER_RANGE enum

Counter range.

EVENT_ID unsigned int

Event Identifier, field is valid only for Event counter group.

INDEX unsigned int

Counter index (as selected by PMSELR), field is valid only for Event/Cycle counter groups.

INTERRUPT bool

Is interrupt enabled on overflow for this counter.

POTINDEX_UPDATE

Value of POT index for current pc changed. Fields:

NEW_POTINDEX unsigned int

New POTIndex.

OLD_POTINDEX unsigned int

Old POTIndex.

PREFETCH_MEMORY64

Prefetch from PRFM or PRFUM instructions. Fields:

PRFOP enum

Prefetch hint.

VADDR unsigned int

Virtual address of the location that should be prefetched.

PRELOAD64_REQUEST

Preload requests from PRFM or PRFUM instructions. Fields:

PRFOP enum

Prefetch hint.

VADDR unsigned int

Virtual address of the location that should be prefetched.

PRELOAD_DATA

Data preload from PLD instruction. Fields:

VADDR unsigned int

Virtual address of the data that should be preloaded.

PRELOAD_INST

Instruction preload from PLI instruction. Fields:

VADDR unsigned int

Virtual address of the instruction that should be preloaded.

PRE_CORE_LOAD

Trace just before a core load. Fields:

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

PRE_CORE_STORE

Trace just before a core store. Fields:

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

PRE_TTB_READ

This event is triggered before reads caused by a translation table walk. Fields:

ADDR unsigned int

The physical address of the read.

PSTATE

PSTATE fields update. Fields:

ALLINT unsigned int

All interrupt mask bit.

BTYPE unsigned int

Branch target identification bit.

CORE_NUM unsigned int

Core number in a multi processor.

DAIF unsigned int

Exception mask bits.

DIT unsigned int

Data Independent Timing bit.

EL unsigned int

Current Exception level field.

EXLOCK unsigned int

Exception return state lock bit.

IL unsigned int

Illegal Execution state bit.

NZCV unsigned int

Condition flags.

PACM unsigned int

PACM bit.

PAN unsigned int

Privileged Access Never state bit.

PM unsigned int

PMU exception mask bit.

PPEND unsigned int

PMU exception pending bit.

SM unsigned int

StreamingMode status bit.

SP unsigned int

Stack pointer register selection bit.

SS unsigned int

Software Step bit.

SSBS unsigned int

Speculative Store Bypass Safe bit.

TCO unsigned int

Tag Check Override bit.

TOUCHED unsigned int

Mask of the fields that have been touched since this trace has last been emitted. Note that touched does not necessarily mean the value changed. The index of the field is the bit index in the mask plus the index of this field plus one.

UAO unsigned int

User Access Override bit.

UINJ unsigned int

Undefined Instruction exceptions bit.

ZA unsigned int

ZA storage and ZT0 register enable bit.

nRW unsigned int

Current Execution state bit.

PSTATE_PACTIVE_OPMODE_TRANSITION

Trace pactive and Opmode mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Operating mode after the transition.

OLD_STATE enum

Operating mode before the transition.

PSTATE_PACTIVE_OPMODE_TRANSITION_DENIED

Trace the denied pactive and Opmode mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Operating mode after the transition.

OLD_STATE enum

Operating mode before the transition.

REASON string

Reason for the transition being denied.

PSTATE_PACTIVE_POWER_TRANSITION

Trace pactive and power mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Power mode after the transition.

OLD_STATE enum

Power mode before the transition.

PSTATE_PACTIVE_POWER_TRANSITION_DENIED

Trace the denied pactive and power mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Power mode after the transition.

OLD_STATE enum

Power mode before the transition.

REASON string

Reason for the transition being denied.

PSTATE_PACTIVE_THREAD_OPMODE_TRANSITION

Trace pactive and Opmode mode transitions for threads on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Operating mode after the transition.

OLD_STATE enum

Operating mode before the transition.

QUANTUM_START

Called at the start of every quantum.

RANGE_PRELOAD_REQUEST

Preload requests from RPRFM. Fields:

COUNT unsigned int

Number of blocks of data to access.

LENGTH unsigned int

Number of contiguous bytes to be accessed in each block.

OP unsigned int

Range prefetch operation .

REUSE unsigned int

Maximum number of bytes to be accessed before next RPRFM .

STRIDE signed int

Stride to advance after accessing LENGTH bytes.

VADDR unsigned int

Base virtual address of range prefetch request.

RUN_STATE

Run state transition. Fields:

INST_COUNT unsigned int

Ticks count at point of transition.

NEW enum

New run state.

OLD enum

Old run state.

SEMIHOSTING_CALL

Call of a semihost function occurred. Fields:

PC unsigned int

The program counter after the semihosting call.

SEMIHOSTING_PRECALL

About to call semihost. Fields:

EL2 bool

Use EL2 translation regime.

NS bool

Is Non-Secure.

REG_WIDTH unsigned int

The current register width in bytes.

SIGNAL

External signal state change. Fields:

SIGNAL enum

Signal that changed.

STATE bool

Signal asserted state.

SOFTWARE_STEP

Return the Debug Software Step state. Fields:

STATE *enum*

Software Step state.

SPSR

SPSR change. Fields:

CORE_NUM *unsigned int*

Core number in a multi processor.

MODE *enum*

Which of the banked SPSR registers is written.

OLD_VALUE *unsigned int*

The old SPSR value.

VALUE *unsigned int*

The new SPSR value.

START_COMPILE

Compilation started. Fields:

VADDR *unsigned int*

Instruction where compilation begins.

STG_MULTIPLE_MICRO_OPS_ADDRESS_RANGE

Address range for MTE stores with multiple uops. Fields:

ADDR_END *unsigned int*

Final address.

ADDR_START *unsigned int*

Starting address.

STORE_MULTIPLE_REGS_ADDRESS_RANGE

Address range for stores with multiple registers. Fields:

ADDR_END *unsigned int*

Final address.

ADDR_START *unsigned int*

Starting address.

SUBOPTIMAL_LDST_RETIRED

suboptimal load/store. Fields:

ACCESS_SIZE *unsigned int*

Log2 of the access width in bytes used for alignment checking (1: half-word, 2: word ...).

OPTIMAL_ALIGN_SELECTOR unsigned int

Selector to decide the optimal alignment size (1: access width, 2: 4-bytes, 3: 8-bytes, ..., 12: 4-kbytes).

VADDR unsigned int

The virtual address of the access.

SYNC

Called for every synchronization, that is every quantum break. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The instruction count of this CPU.

LOCAL_QUANTUM unsigned int

The local quantum of this CPU.

LOCAL_TIME unsigned int

The local time of this CPU.

SYNC_REASON enum

Reason for sync.

SYSCALL

System call instruction executed. Fields:

IMM unsigned int

Immediate value of the system call instruction.

TYPE enum

System call type.

VADDR unsigned int

Instruction that caused the system call.

SYSREG_READ128

System Coprocessor 128 bit register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

REGNUM enum

Internal register number.

REGNUM_PHYS enum

Internal register number physically accessed (e.g. in presense of VHE redirection).

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_READ64

System Coprocessor register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REGNUM enum

Internal register number.

REGNUM_OPERAND enum

Register number corresponding to the instruction operand.

REGNUM_PHYS enum

Internal register number physically accessed (e.g. in presense of VHE redirection).

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_UPDATE128

Triggers when the system updates a register. Fields:

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG enum

Register number.

UNKNOWN unsigned int

Bits of the register which became unknown.

VALUE unsigned int

Value written to the register.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_UPDATE32

Triggers when the system updates a register. Fields:

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG enum

Register number.

UNKNOWN unsigned int

Bits of the register which became unknown.

VALUE unsigned int

Value written to the register.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_UPDATE64

Triggers when the system updates a register. Fields:

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG enum

Register number.

UNKNOWN unsigned int

Bits of the register which became unknown.

VALUE unsigned int

Value written to the register.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_WRITE128

System Coprocessor 128 bit register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

REGNUM enum

Internal register number.

REGNUM_PHYS enum

Internal register number physically accessed (e.g. in presense of VHE redirection).

UNDEF bool

The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_WRITE64

System Coprocessor register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REGNUM enum

Internal register number.

REGNUM_OPERAND enum

Register number corresponding to the instruction operand.

REGNUM_PHYS enum

Internal register number physically accessed (e.g. in presense of VHE redirection).

UNDEF bool

The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

UNALIGNED_LDST_RETIRED

Processor unaligned load/store. Fields:

ACCESS_SIZE unsigned int

Log2 of the access width in bytes used for alignment checking (1=>half-word, 2->word ...).

NUMBER_OF_BEATS unsigned int

Number of accesses (beats) in this burst (Total size in bytes is ACCESS_SIZE * NUMBER_OF_BEATS).

VADDR unsigned int

The virtual address of the access.

VFP_D_REGS

VFP/NEON D 64 bit register write. Fields:

ALIASING enum

Type of register aliasing used.

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a write to the corresponding bit in the register. For example, 0x1 would indicate a write to VALUE[0].

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

VFP_D_REGS_READ

VFP/NEON D 64 bit register read. Fields:

ID unsigned int

The register number.

VALUE unsigned int

The current value of the register.

VFP_Q_REGS

VFP/NEON Q 128 bit register write. Fields:

ALIASING enum

Type of register aliasing used.

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a write to the corresponding byte in the register. For example, 0x1 would indicate a write to VALUE[7:0].

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

VFP_Q_REGS_READ

VFP/NEON Q 128 bit register read. Fields:

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a read to the corresponding byte in the register. For example, 0x1 would indicate a read to VALUE[7:0].

VALUE unsigned int

The current value of the register.

VFP_SYS_REGS

Writes to the VFP/NEON units system registers. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ID enum

Which VFP system register is written.

MASK unsigned int

The mask indicating updated cumulative bits.

OLD_VALUE unsigned int

The register's old value overwritten.

VALUE unsigned int

The new value written to the VFP system register.

VFP_S_REGS

VFP/NEON S 32 bit register write. Fields:

ALIASING enum

Type of register aliasing used.

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a write to the corresponding bit in the register. For example, 0x1 would indicate a write to VALUE[0].

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

VFP_S_REGS_READ

VFP/NEON S 32 bit register read. Fields:

ID unsigned int

The register number.

VALUE unsigned int

The current value of the register.

VIRTUAL_TAG_ALLOCATED

Virtual tag information when allocated. Fields:

TAG_VADDR unsigned int

Virtual tag virtual address.

VADDR unsigned int

Data Virtual address.

VIRTUAL_TAG unsigned int

Virtual tag value.

WAYPOINT

Signals end of basic block execution. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ISSET enum

Origin instruction set.

IS_COND bool

Indicates if this is a conditional waypoint.

PC unsigned int

Origin address (or 0 if unavailable).

TAKEN bool

Indicates if this waypoint was taken.

TARGET unsigned int

Destination address.

TARGET_ISET enum

Destination instruction set.

WFE_END

WFE ended. Fields:

INST_COUNT unsigned int

Ticks count when leaving WFE.

WFE_EVENT_REGISTER

WFE event register status: set/clear, reason. Fields:

INST_COUNT unsigned int

Ticks count.

REASON enum

Reason for set/clear. Only REASON==0 (Cleared by WFE) clears the bit, all other reasons set it.

WFE_IGNORED

WFE ignored. Fields:

EVENT bool

This WFE was ignored because the event register was set.

INST_COUNT unsigned int

Ticks count when ignoring WFE.

TRAPPED bool

This WFE was trapped.

WFE_START

WFE entered. Fields:

INST_COUNT unsigned int

Ticks count when entering WFE.

WFI_END

WFI ended. Fields:

INST_COUNT unsigned int

Ticks count when leaving WFI.

WFI_IGNORED

WFI ignored. Fields:

DISABLED bool

This WFI was ignored because WFI is disabled.

INST_COUNT unsigned int

Ticks count when ignoring WFI.

REASON enum

specifies reason why WFI trace was ignored.

TRAPPED bool

This WFI was trapped.

WFI_START

WFI entered. Fields:

INST_COUNT unsigned int

Ticks count when entering WFI.

WFI_WAKEUP

WFI wakeup. Fields:

INST_COUNT unsigned int

Ticks count when WFI wakeup occurred.

REASON enum

Reason for wakeup.

tcm0

Trace TCM accesses.

tcm0_RAZ_WI

Trace TCM accesses. Fields:

ADDR unsigned int

Address of the **RAZ/WI** access.

tcm1

Trace TCM accesses.

tcm1_RAZ_WI

Trace TCM accesses. Fields:

ADDR unsigned int

Address of the **RAZ/WI** access.

tcm2

Trace TCM accesses.

tcm2_RAZ_WI

Trace TCM accesses. Fields:

ADDR unsigned int

Address of the **RAZ/WI** access.

2.63 ARM_Neoverse-E1

This section describes the trace sources.

AA64_ASE_SVE_REGS

Modification of AArch64 Advanced SIMD or SVE Vector Registers (via SIMD or Scalar operations), SVE Predicate Registers and ZT0. Fields:

ID **enum**

The register identifier.

MASK **unsigned int**

Bitmask of the register to signal the modified bits in the VALUE field.

SM **bool**

Whether the PE is in Streaming Mode.

VALUE **unsigned int**

Value of AArch64 vector register.

ASYNC_MEMORY_FAULT

Context ID Register write. Fields:

FAULT **unsigned int**

Fault status present in ESR.ISS encoding for Serror in ARCH64 and present in DFSR encoding for external abort format in ARCH32.

PADDR **unsigned int**

Physical Address (or 0 if unavailable).

VADDR **signed int**

Virtual Address (or 0 if unavailable).

ATOMIC_END_ACCESS

Bus trace access for atomics. Fields:

ACCESS_FAIL **bool**

Memory access failed.

ADDR **unsigned int**

The virtual address of the access.

ATTR **unsigned int**

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

COMPARE_VALUE **unsigned int**

Compare value for CAS.

LOAD_VALUE unsigned int

Loaded values.

NSDESC unsigned int

The security state of the access.

OPERAND_VALUE unsigned int

Operation's operand.

OPERATION enum

Operation type.

PADDR unsigned int

The physical address of the access.

PRIV bool

Is this a privileged access?.

ATOMIC_START_ACCESS

Bus trace access for atomics. Fields:

ADDR unsigned int

The virtual address of the access.

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

COMPARE_VALUE unsigned int

Compare value for CAS.

NSDESC unsigned int

The security state of the access.

OPERAND_VALUE unsigned int

Operation's operand.

OPERATION enum

Operation type.

PADDR unsigned int

The physical address of the access.

PRIV bool

Is this a privileged access?.

ArchMsg.Error.exit_code

DISPLAY exit code: %{EXITCODE} component kind: %{KIND} reason: %{REASON}. Fields:

EXITCODE signed int

Exit code to be returned.

KIND string

Component kind that invoked the exit code trace.

REASON string

Reason why the exit code trace was invoked (optional).

ArchMsg.Warning.IgnoredWarmResetRequest

Event fires when software attempts to request warm reset through write to deprecated DBGPRCR.CWRR.

ArchMsg.Warning.RAS_ErrselrUnpredictableState

DISPLAY ERRSELR is in unpredictable state due to value greater than number_of_error_records written to it.

ArchMsg.Warning.RAS_UnpredictableStateWrite

DISPLAY %{REG_NAME} write ignored due to unpredictable ERRSELR state. Fields:

REG_NAME string

Name of register being written.

ArchMsg.Warning.ReportedRevisionRegisterRead

DISPLAY Reported revision and patch do not match that of the cpu. Fields:

ORIGINAL_VALUE unsigned int

The value the register would have based on the modelled revision.

REGISTER string

The name of the register that has ben read.

REPORTED_VALUE unsigned int

The value the register has been overridden to.

ArchMsg.Warning.Unpredictably Indexed PM Event Register

TAGS UNPREDICTABLE DISPLAY Event %{IsDirect:(Selector value|Register)} %{SEL} is >= %{N}. Fields:

IsDirect unsigned int

Direct Access rather than indirect.

N unsigned int

Number of Accessible Registers.

SEL unsigned int

Selector.

ArchMsg.Warning.cache_contents_unknown

A cache has been activated without being invalidated and may contain **UNKNOWN** content. DISPLAY %{SIDE:(D|I)}-Cache was enabled in %{REGIME:(SECURE|NON_SECURE|HYP)} regime but was not invalidated since power-on: cache lines could contain **UNKNOWN** content %{SIDE:(Note: D-side caches must invalidate SECURE lines even if the cache is only used by NON_SECURE code in order to ensure that unexpected dirty lines tagged SECURE are not naturally evicted)}. Fields:

REGIME unsigned int

Current translation regime.

SIDE unsigned int

0 d-side cache or unified cache, 1 i-side.

ArchMsg.Warning.dcimvac_matches_watchpoint

TAGS **UNPREDICTABLE** DISPLAY Watchpoints matching an AArch32 DCIMVA are implementation defined.

ArchMsg.Warning.decode_fieldmismatch

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with different fields %{FIELD1} and %{FIELD2}. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_fracbitsnegative

TAGS **UNPREDICTABLE** DISPLAY VCVT Instruction is unpredictable with a negative number of fraction bits.

ArchMsg.Warning.decode_initblock

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable in an IT block.

ArchMsg.Warning.decode_initblocknotlast

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable in an IT block when not the last.

ArchMsg.Warning.decode_invalid_condition

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable when the condition is not AL.

ArchMsg.Warning.decode_invalidfieldcombination

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with the values in %{FIELD1} and %{FIELD2}. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_invalidvalue

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with the value in %{FIELD}. Fields:

FIELD enum

Field name.

ArchMsg.Warning.decode_mem_hint_unallocated

TAGS **UNPREDICTABLE** DISPLAY There is no memory hint allocated to this bit pattern.

ArchMsg.Warning.decode_registermatch

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with %{FIELD1} and %{FIELD2} as the same register. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_registermismatch

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with %{FIELD1} and %{FIELD2} as different registers. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_registeroutofrange

TAGS **UNPREDICTABLE** DISPLAY Floating-point instruction is unpredictable indexing beyond the end of the register bank.

ArchMsg.Warning.decode_sbzsbo

TAGS **UNPREDICTABLE** DISPLAY Reserved bits in the instruction are not canonical.

ArchMsg.Warning.decode_transactiontoobig

TAGS **UNPREDICTABLE** DISPLAY Attempt to perform a memory transaction of more than 128 bytes.

ArchMsg.Warning.decode_unpred_other

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable.

ArchMsg.Warning.decode_unpreduseofpc

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable when %{FIELD} is PC. Fields:

FIELD enum

Field name.

ArchMsg.Warning.decode_unpreduseofr13

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable when %{FIELD} is R13. Fields:

FIELD enum

Field name.

ArchMsg.Warning.decode_unpreduseofr13inreglist

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with R13 in register-list.

ArchMsg.Warning.decode_useofd16orhigher

TAGS **UNPREDICTABLE** DISPLAY Attempt to use register D16 or higher where it is not allowed.

ArchMsg.Warning.decode_writebackandbaseinlist

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with write back when the base register is in the transfer list.

ArchMsg.Warning.decode_zeroregistersinlist

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable when no registers are to be transferred.

ArchMsg.Warning.recursive_branch

An instruction is performing a branch that targets the same instruction. If this was intended then a WFI might be more effective DISPLAY recursive branch detected at PC=%{PC}. PRIMARY KEY PC. Fields:

PC unsigned int

Address of instruction causing the branch.

ArchMsg.Warning.recursive_exception

An instruction has generated an exception that targets the same instruction. DISPLAY recursive %{TYPE} exception detected at PC=%{PC} (CPSR=%{CPSR}). PRIMARY KEY PC. Fields:

CPSR unsigned int

Processor state.

PC unsigned int

Address of instruction causing the exception.

TYPE enum

Type of exception.

ArchMsg.Warning.reentrant_vector_catch

TAGS **UNPREDICTABLE** DISPLAY Vector catch debug event on Prefetch or Data abort vector %{ADDR}. Fields:

ADDR unsigned int

vector address.

ArchMsg.Warning.reserved_it_state

TAGS **UNPREDICTABLE** DISPLAY Execution with reserved IT state %{ITSTATE}. Fields:

ITSTATE unsigned int

reserved IT state.

ArchMsg.Warning.tlb_contents_unknown

Translation has been enabled without invalidating the TLB. DISPLAY Translation was enabled in the %{PAS:(Secure|Non-secure|Root|Realm)} EL%{EL:d} translation regime (vmid=%{VMID:x}) without invalidating %{INVALIDITY:(|d-side |i-side |)}entries since power-on: %{INVALIDITY:(|D|I|)}TLB could contain **UNKNOWN** content. Fields:

EL unsigned int

Exception level of translation regime.

INVALIDITY unsigned int

Which TLB may be invalid (if TLBs are separate).

NON_SECURE bool

Whether the translation regime is non-secure.

PAS unsigned int

Physical address space of translation regime.

VMID unsigned int

Current VMID, if applicable.

ArchMsg.Warning.unknown_DLR_DSPSR

Exiting debug state with unknown DLR or DSPSR.

ArchMsg.Warning.unknown_ELR_SPSR

Returning from debug exception with unknown ELR or SPSR.

ArchMsg.Warning.unpred_FAR_on_LRCPC3_non_atomic_mem_access

TAGS **UNPREDICTABLE** Unpredictable FAR/FSTATUS on LRCPC3 pair instructions resulting in two separate memory accesses
DISPLAY Unpredictable FAR on LRCPC3 non atomic memory access at %{ADDR1} with FSTATUS %{FSTATUS1} or %{ADDR2} with FSTATUS %{FSTATUS2}. Fields:

ADDR1 unsigned int

Possible FAR address 1.

ADDR2 unsigned int

Possible FAR address 2.

FSTATUS1 unsigned int

Possible fault status value 1.

FSTATUS2 unsigned int

Possible fault status value 2.

ArchMsg.Warning.unpred_async_tag_check_fail_on_sve_speculative_load_do_update_ffr

Unpredictable set to FALSE of the FFR element associated to SVE speculative load with an async tag check fail TAGS **UNPREDICTABLE**.

ArchMsg.Warning.unpredictable_a32_breakpoint

TAGS **UNPREDICTABLE** DISPLAY Unpredictable breakpoint on a A32 instruction at %{ADDR1} Offset %{OFFSET} BAS %{BAS} for a %{IS_ADDRESS_MISMATCH_BREAKPOINT:(Mismatch|Match)} Breakpoint number %{N:d} Breakpoint address %{ADDR2}. Fields:

ADDR1 unsigned int

Address.

ADDR2 unsigned int

Breakpoint address.

BAS unsigned int

BAS field.

IS_ADDRESS_MISMATCH_BREAKPOINT bool

Is a Mismatch breakpoint, not a Match breakpoint.

N unsigned int

Breakpoint number.

OFFSET unsigned int

Always has a value of zero.

ArchMsg.Warning.unpredictable_banked_register_access

TAGS **UNPREDICTABLE** DISPLAY Access to banked register (R=%{R:d} SYSm=%{SYSm:2x}) is unpredictable. Fields:

R unsigned int

R bit.

SYSm unsigned int

SYSm.

ArchMsg.Warning.unpredictable_ccfail_undef

TAGS **UNPREDICTABLE** DISPLAY An instruction %{OPCODE} at VA=%{VADDR} failed its condition codes check but would otherwise have caused a non-data-dependent trap or undefined exception. Behaviour is implementation-defined choice of exception or nop. Fields:

OPCODE unsigned int

opcode of the instruction.

VADDR unsigned int

Virtual address of the instruction being executed.

ArchMsg.Warning.unpredictable_ici_bits

TAGS **UNPREDICTABLE** DISPLAY ICI bits do not apply or are architecturally unpredictable for instruction.

ArchMsg.Warning.unpredictable_t32_breakpoint

TAGS **UNPREDICTABLE** DISPLAY Unpredictable breakpoint on a T32 instruction at %{ADDR1} Offset %{OFFSET} BAS %{BAS} for a %{IS_ADDRESS_MISMATCH_BREAKPOINT:(Mismatch|Match)} Breakpoint number %{N:d} Breakpoint address %{ADDR2}. Fields:

ADDR1 unsigned int

Address.

ADDR2 unsigned int

Breakpoint address.

BAS unsigned int

BAS field.

IS_ADDRESS_MISMATCH_BREAKPOINT bool

Is a Mismatch breakpoint, not a Match breakpoint.

N unsigned int

Breakpoint number.

OFFSET unsigned int

Indicates which halfword of a 32-bit Thumb instruction the breakpoint is set on. 0 for the first halfword or if the instruction is 16 bits wide, or 1 for the second halfword.

ArchMsg.Warning.unpredictable_t32_vector_catch

TAGS **UNPREDICTABLE** DISPLAY Unpredictable Vector catch exception on a T32 instruction at %{ADDR} Offset %{OFFSET}. Fields:

ADDR unsigned int

Address.

OFFSET unsigned int

Indicates which halfword of a 32-bit Thumb instruction the vector catch exception is set on. 0 for the first halfword or if the instruction is 16 bits wide, or 1 for the second halfword.

ArchMsg.Warning.unpredictable_watchpoint_far

TAGS **UNPREDICTABLE** DISPLAY The value for FAR/EDWAR for hitting this watchpoint may be anywhere in the range %{LOWER_BOUND} to %{UPPER_BOUND}. Fields:

LOWER_BOUND unsigned int

the lowest address accessed by the instruction that triggered the watchpoint.

UPPER_BOUND unsigned int

the highest watchpointed address accessed by that instruction.

ArchMsg.Warning.unpredictable_watchpoint_far_isscmp

DISPLAY The value for FAR/EDWAR for hitting this watchpoint may be anywhere in the range %{LOWER_BOUND} to %{UPPER_BOUND}. Fields:

LOWER_BOUND unsigned int

the lowest address accessed by the instruction that triggered the watchpoint.

UPPER_BOUND unsigned int

the highest watchpointed address accessed by that instruction.

ArchMsg.Warning.warning_AdvSIMDEExpandImmUnexpectedZero

TAGS **UNPREDICTABLE** DISPLAY AdvSIMDEExpandImm may treat this immediate value as **UNPREDICTABLE**.

ArchMsg.Warning.warning_ConditionalSMC

TAGS **UNPREDICTABLE** DISPLAY SMC instruction has **UNPREDICTABLE** effects when conditional and when combined with traps.

ArchMsg.Warning.warning_TLBID_out_of_range

TLBID out of range in TLBI according to the configuration of NIS, NVIS, NOS and NVOS. Fields:

NIS unsigned int

NIS.

NOS unsigned int

NOS.

NVIS unsigned int

NVIS.

NVOS unsigned int

NVOS.

TLBID unsigned int

The TLBID value used in the TLBI instruction.

ArchMsg.Warning.warning_TTL_or_TG_mismatches_current_translation_granule

TTL or TG mismatches current translation granule in TLBI VA/RVA/IPA/RIPA. Fields:

CURRENT_TG unsigned int

Current TG as indicated by TCR_ELx or VTCR_EL2.

TLBI_TG unsigned int

TG (or TTL[3:2]) used in TLBI instruction.

ArchMsg.Warning.warning_access_crosses_page_boundary

TAGS **UNPREDICTABLE** DISPLAY Access crosses page boundary.

ArchMsg.Warning.warning_access_crosses_page_boundary_has_fault

TAGS **UNPREDICTABLE** DISPLAY Access that crosses page boundary has fault(s).

ArchMsg.Warning.warning_access_crosses_page_boundary_has_fault_on_both_pages

TAGS **UNPREDICTABLE** DISPLAY Access that crosses page boundary has a fault on each page.
Fields:

ADDR1 unsigned int

FAR address on first page.

ADDR2 unsigned int

FAR address on second page.

FSTATUS1 unsigned int

Fault status on first page.

FSTATUS2 unsigned int

Fault status on second page.

ArchMsg.Warning.warning_access_crosses_page_boundary_spanning_different_memory

TAGS **UNPREDICTABLE** DISPLAY Access at address %{ADDR} crosses page from MEM_TYPE %{MEMTYPE_PAGE1} to %{MEMTYPE_PAGE2}. Fields:

ADDR unsigned int

address of access crossing page.

MEMTYPE_PAGE1 signed int

First Page's memory type (0-Normal, 1-Non-Normal).

MEMTYPE_PAGE2 signed int

Second Page's memory type (0-Normal, 1-Non-Normal).

ArchMsg.Warning.warning_access_wraps_around_memory

TAGS **UNPREDICTABLE** DISPLAY Access wraps around memory in %{WIDTH:d}bit mode.

Fields:

WIDTH unsigned int

Address width.

ArchMsg.Warning.warning_bcr_linking_status

TAGS **UNPREDICTABLE** DISPLAY DBGBCR.{BT2,BT} combination gives unpredictable behaviour.

ArchMsg.Warning.warning_bcr_mask_reserved

TAGS **UNPREDICTABLE** DISPLAY DBGBCR_ELn.MASK is set to a reserved value.

ArchMsg.Warning.warning_bvr_masked_bit_not_zero

TAGS **UNPREDICTABLE** DISPLAY DBGBVRn_EL1.VA contains a bit masked by DBGBCRn_EL1.MASK that is not zero.

ArchMsg.Warning.warning_ccsidr_unimplemented_level

TAGS **UNPREDICTABLE** DISPLAY CCSIDR read while CSSELR %{CSSELR:x} points to an unimplemented cache level. Fields:

CSSELR unsigned int

current effective value of CSSELR.

ArchMsg.Warning.warning_change_to_ns_when_tge_set

TAGS **UNPREDICTABLE** DISPLAY Attempting to change NS to 1 when HCR.TGE = 1.

ArchMsg.Warning.warning_contiguous_bit_check_abort

TAGS **UNPREDICTABLE** DISPLAY An abort occurred whilst attempting to check TLB entry is contiguous at %{ENTRY_ADDR}. Fields:

ENTRY_ADDR unsigned int

address of conflicting TLB entry.

ArchMsg.Warning.warning_contiguous_bit_error

TAGS **UNPREDICTABLE** DISPLAY TLB entry at %{ENTRY_ADDR} was not contiguous with entry at %{CONTIG_ADDR} Contents are %{ENTRY_DATA} but expected %{CONTIG_DATA}. Fields:

CONTIG_ADDR unsigned int

address of the TLB first read, with which this entry is expected to be contiguous.

CONTIG_DATA unsigned int

expected contents based on the entry at CONTIG_ADDR.

ENTRY_ADDR unsigned int

address of conflicting TLB entry.

ENTRY_DATA unsigned int

contents of conflicting TLB entry.

ArchMsg.Warning.warning_contiguous_bit_error_D128

TAGS **UNPREDICTABLE** DISPLAY TLB entry at %{ENTRY_ADDR} was not contiguous with entry at %{CONTIG_ADDR} Contents are %{ENTRY_DATA_HI} %{ENTRY_DATA_LO} but expected %{CONTIG_DATA}. Fields:

CONTIG_ADDR unsigned int

address of the TLB first read, with which this entry is expected to be contiguous.

CONTIG_DATA unsigned int

expected contents based on the entry at CONTIG_ADDR.

ENTRY_ADDR unsigned int

address of conflicting TLB entry.

ENTRY_DATA_HI unsigned int

Higher bit contents of conflicting TLB entry.

ENTRY_DATA_LO unsigned int

Lower bit contents of conflicting TLB entry.

ArchMsg.Warning.warning_cp10_cp11_mismatch

TAGS **UNPREDICTABLE** DISPLAY Unpredictable configuration of CP10 and CP11 controls at EL %{EL:d}. Fields:

EL unsigned int

EL owning the control.

ArchMsg.Warning.warning_cp10_cp11_reserved_value

TAGS **UNPREDICTABLE** DISPLAY Unpredictable configuration of CPACR CP10 (%{CP10}) or CP11 (%{CP11}). Fields:

CP10 unsigned int

CP10 access.

CP11 unsigned int

CP11 access.

ArchMsg.Warning.warning_csselr_level_out_of_range

TAGS **UNPREDICTABLE** DISPLAY CSSELR.Level out of range, written value %{LEVEL} with only %{IMPLEMENTED:d} levels implemented. Fields:

IMPLEMENTED unsigned int

number of cache levels implemented.

LEVEL unsigned int

written value of CSSELR.Level.

TYPE enum

The type of cache selected by CSSELR.InD and TnD.

ArchMsg.Warning.warning_debug_flow_control_bits_not_obeyed

TAGS **UNPREDICTABLE** DISPLAY Flow control bits for %{REGISTER} were not obeyed, causing %{ERROR}. Fields:

ERROR enum

type of error resulting.

REGISTER enum

register under consideration.

ArchMsg.Warning.warning_debug_register_access_during_reset

TAGS **UNPREDICTABLE** DISPLAY Unpredictable %{IS_WRITE:(read|write)} access to debug register offset %{OFFSET} during reset. Fields:

IS_WRITE unsigned int

Write Not Read.

OFFSET unsigned int

Register Offset.

ArchMsg.Warning.warning_decode_cps_inconsistent_fields

TAGS **UNPREDICTABLE** DISPLAY The mode setting fields in the CPS instruction are inconsistent.

ArchMsg.Warning.warning_decode_invalid_state

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable in current exception-level/security state.

ArchMsg.Warning.warning_default_cacheable_mmu_on

TAGS **UNPREDICTABLE** DISPLAY Default cacheable is enabled (HCR.DC = 1) while MMU is enabled.

ArchMsg.Warning.warning_default_cacheable_vmmu_off

TAGS **UNPREDICTABLE** DISPLAY Default cacheable is enabled (HCR.DC = 1) while HCR.VM = 0.

ArchMsg.Warning.warning_deprecated_wvr_bit_2_set

TAGS **UNPREDICTABLE** DISPLAY DBGWVR%{N:d}_EL1=%{WVR:h} has bit 2 set, deprecated in ARMv8. Fields:

N unsigned int

index.

WVR unsigned int

DBGWVR.

ArchMsg.Warning.warning_eret_while_software_step_active_pending

Missing ISB between setting MDSCR_EL1.SS and ERET TAGS **UNPREDICTABLE**.

ArchMsg.Warning.warning_exclusive_to_non_normal

TAGS **UNPREDICTABLE** DISPLAY Exclusive Access to Non-Normal Memory %{ADDR}. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_exclusive_to_non_writeback

TAGS **UNPREDICTABLE** DISPLAY Exclusive Access to Non-Writeback-Cacheable Memory
%{ADDR}. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_execute_from_device_memory

TAGS **UNPREDICTABLE** DISPLAY An attempt was made to execute from Device memory.

ArchMsg.Warning.warning_illegal_cpsr_mode

TAGS **UNPREDICTABLE** DISPLAY Writing an illegal or unimplement mode (%{NEW_MODE:h})
to CPSR was unpredictable. Fields:

NEW_MODE unsigned int

New value of CPSR.M.

ArchMsg.Warning.warning_illegal_srs_mode

TAGS **UNPREDICTABLE** DISPLAY Illegal or **UNPREDICTABLE** mode (%{MODE:h}) used for SRS
instruction. Fields:

MODE unsigned int

mode for Banked SP.

ArchMsg.Warning.warning_implementation_defined_read_debug_register_in_SCS

Read Access to Debug registers from the processor is IMPLEMENTATION DEFINED TAGS
IMP_DEF. Fields:

PPB_OFFSET unsigned int

debug register offset.

ArchMsg.Warning.warning_implementation_defined_sequential_security_transitions_supported

The behavior of sequential instruction fetches that cross from non-secure to secure memory
and contain an SG instruction is CONSTRAINED **UNPREDICTABLE** TAGS **UNPREDICTABLE**.

ArchMsg.Warning.warning_implementation_defined_stack_limit_check_supported

It is **IMPLEMENTATION DEFINED** whether stack pointer limit checking is performed for this
instruction TAGS IMP_DEF. Fields:

ADDRESS unsigned int

Address of instruction.

OPCODE unsigned int

instruction opcode.

ArchMsg.Warning.warning_implementation_defined_taggednonsharedmemorytype

TAGS IMP_DEF DISPLAY Memory attribute has Tagged, non shared memory type; it is
IMPLEMENTATION DEFINED whether the memory location is treated as Tagged or non-Tagged.

ArchMsg.Warning.warning_implementation_defined_write_debug_register_in_SCS

Write Access to Debug registers from the processor is IMPLEMENTATION DEFINED TAGS IMP_DEF. Fields:

DATA unsigned int

data attempted to be written.

PPB_OFFSET unsigned int

debug register offset.

ArchMsg.Warning.warning_invalid_tcr_granule

TAGS UNPREDICTABLE DISPLAY TCR.TG{%TG_ID:(0|1)} bits have been set to a granule size {%REQUEST:d}K not implemented - using {%SUBSTITUTE:d}K. Fields:

REQUEST signed int

page size requested (or 0 for reserved).

SUBSTITUTE signed int

best guess available page size.

TG_ID bool

bits TG0 or TG1.

ArchMsg.Warning.warning_load_multiple_user_registers_from_user_mode

TAGS UNPREDICTABLE DISPLAY An LDM(user registers) instruction executed from user mode.

ArchMsg.Warning.warning_mte_in_mem_attr_not_supported

TAGS UNPREDICTABLE DISPLAY Memory attribute {%ATTR} has Tagged memory type but Armv8.5-MTE is not implemented. Fields:

ATTR unsigned int

Memory attribute.

ArchMsg.Warning.warning_reserved_breakpoint_state_match

TAGS UNPREDICTABLE DISPLAY {%IS_BREAKPOINT:(Watchpoint|Breakpoint)} programmed with a reserved combination of HMC, SSC and {%IS_BREAKPOINT:(PAC|PMC)}. Fields:

IS_BREAKPOINT unsigned int

Is a breakpoint, not a watchpoint.

ArchMsg.Warning.warning_shareability

TAGS UNPREDICTABLE DISPLAY Unpredictable: combination of the 1st and 2nd stages of translation is Normal Inner Non-Cacheable, Outer Non-Cacheable.

ArchMsg.Warning.warning_software_step_set_while_enabled

TAGS UNPREDICTABLE DISPLAY MDSCR_EL1.SS set to 1 while software step debug exceptions are enabled.

ArchMsg.Warning.warning_thumb_instruction_wraps_around_memory

TAGS UNPREDICTABLE DISPLAY Thumb instruction wraps around memory.

ArchMsg.Warning.warning_ttbr_sbz_bits_are_not_zero

TAGS UNPREDICTABLE DISPLAY %{TTBR}[%{MSB:d}:0] should be zero, but are not. Fields:

MSB unsigned int

x-1.

TTBR enum

which TTBR register.

ArchMsg.Warning.warning_unaligned_address_dbgdttrrx_write

TAGS UNPREDICTABLE DISPLAY Unaligned address %{ADDR} in XO/RO when DBGDTTRRX written in memory access mode. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unaligned_address_dbgdttrtx_read

TAGS UNPREDICTABLE DISPLAY Unaligned address %{ADDR} in XO/RO when DBGDTTRTX read in memory access mode. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unaligned_to_device

TAGS UNPREDICTABLE DISPLAY Unaligned Access to Device Memory %{ADDR}. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unaligned_to_strongly_ordered

TAGS UNPREDICTABLE DISPLAY Unaligned Access to Strongly Ordered Memory %{ADDR}. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unknown_sau_rnr

TAGS UNPREDICTABLE DISPLAY SAU_RNR was set to an unsupported value.

ArchMsg.Warning.warning_unpredictable_AIRCR_PRIS_and_BFHFNMINs

TAGS UNPREDICTABLE DISPLAY The effect of setting both AIRCR.BFHFNMINs and AIRCR.PRIS to 1 is UNPREDICTABLE.

ArchMsg.Warning.warning_unpredictable_AIRCR_VECTCLRACTIVE_when_not_in_debug

TAGS UNPREDICTABLE DISPLAY The effect of writing a 1 to AIRCR.VECTCLRACTIVE if the processor is not halted in Debug state is UNPREDICTABLE.

ArchMsg.Warning.warning_unpredictable_AIRCR_VECTRESET_and_SYSRESETREQ

TAGS UNPREDICTABLE DISPLAY When the processor is halted in Debug state, if a write to the register writes a 1 to both VECTRESET and SYSRESETREQ, the behavior is UNPREDICTABLE.

ArchMsg.Warning.warning_unpredictable_AIRCR_VECTRESET_when_not_in_debug

TAGS **UNPREDICTABLE** DISPLAY The effect of writing a 1 to AIRCR.VECTRESET if the processor is not halted in Debug state is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_AIRCR_incorrect_VKEY

TAGS **UNPREDICTABLE** DISPLAY The value 0x05FA must be written to AIRCR.VKEY, otherwise the register write is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_EXC_RETURN_Reserved_Bit

EXC_RETURN[23:7] are reserved with the special condition that all bits should be written as one. Values other than all 1s are **UNPREDICTABLE**. TAGS **UNPREDICTABLE**. Fields:

EXC_RETURN unsigned int

Special PC value.

SBO_MASK unsigned int

Which bits Should Be One to not UNPRED.

ArchMsg.Warning.warning_unpredictable_PMEVTYPEPn_ELO_VS_reserved_value

TAGS **UNPREDICTABLE** DISPLAY Unpredictable reserved value of PMEVTYPEN<n>_ELO.VS.

ArchMsg.Warning.warning_unpredictable_change_to_DEMCR_MON_STEP_at_insufficient_priority

TAGS **UNPREDICTABLE** DISPLAY The effect of changing DEMCR.MON_STEP at an execution priority that is lower than the priority of the DebugMonitor exception is **CONSTRAINED UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_change_to_priority_of_active_exception

TAGS **UNPREDICTABLE** DISPLAY Changing the priority of an active exception is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_clear_lspact

TAGS **UNPREDICTABLE** DISPLAY **UNPREDICTABLE** state on context switch. Fields:

DRVEXC_TAKEN unsigned int

Derived exception taken, not pended.

REACHED_LAST_STORE unsigned int

Only the last faulted.

ArchMsg.Warning.warning_unpredictable_dbg_exit_unaligned_dlr

TAGS **UNPREDICTABLE** DISPLAY Debug exit to AArch32 with DLR[0] set to 1. PC[0] can be set to 0 or to DLR[0].

ArchMsg.Warning.warning_unpredictable_exception_catch

TAGS **UNPREDICTABLE** DISPLAY Generation of an exception catch event is unpredictable.

ArchMsg.Warning.warning_unpredictable_exception_return_inconsistent_state

TAGS **UNPREDICTABLE** DISPLAY state on exception return is unpredictable.

ArchMsg.Warning.warning_unpredictable_exception_return_instruction

TAGS **UNPREDICTABLE** DISPLAY exception return instruction is unpredictable.

ArchMsg.Warning.warning_unpredictable_in_debug_state

TAGS **UNPREDICTABLE** DISPLAY Instruction is **UNPREDICTABLE** when executed in debug state.

ArchMsg.Warning.warning_unpredictable_mair_encoding

TAGS **UNPREDICTABLE** DISPLAY Memory attribute %{ATTR} has invalid MAIR encoding; this may lead to unpredictable results. Fields:

ATTR unsigned int

Memory attribute.

ArchMsg.Warning.warning_unpredictable_pmu_counter_access

TAGS **UNPREDICTABLE** DISPLAY Access to PMU counters from non-secure EL0 or EL1 is **UNPREDICTABLE** with MDCR_EL2.HPMN set to 0.

ArchMsg.Warning.warning_unpredictable_prioritization_breakpoint_match_vector_catch

TAGS **UNPREDICTABLE** DISPLAY Two events with the same priority occurred at the same instruction: Address Matching Vector Catch debug event, and Breakpoint debug event. It is constrained unpredictable which is taken.

ArchMsg.Warning.warning_unpredictable_stack_selection

TAGS **UNPREDICTABLE** DISPLAY **UNPREDICTABLE** state on context switch.

ArchMsg.Warning.warning_unpredictable_tsize_out_of_range

TAGS **UNPREDICTABLE** DISPLAY %{TCR}.T%{TTBR:d}SZ (%{TSIZE:d} bits) is out of range. Must be between 25 and %{TSIZE_MAX:d}. Fields:

TCR enum

Which TCR.

TSIZE signed int

Value of TSize.

TSIZE_MAX signed int

The maximum allowed value.

TTBR unsigned int

Which TTBR region.

ArchMsg.Warning.warning_unpredictable_unaligned_pc_as_base_register

TAGS **UNPREDICTABLE** DISPLAY The PC value must be word-aligned, otherwise the behavior of the instruction is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_unaligned_pop_stack

TAGS **UNPREDICTABLE** DISPLAY **CONSTRAINED UNPREDICTABLE** unaligned stack frame %{FRAME_ADDR:x} in exception-return. Fields:

FRAME_ADDR unsigned int

unaligned stack frame address.

ArchMsg.Warning.warning_unpredictable_unaligned_to_non_writeback

TAGS **UNPREDICTABLE** DISPLAY Unaligned Access to Non-Writeback-Cacheable Memory.

ArchMsg.Warning.warning_unpredictable_vmsa_memattrib

TAGS **UNPREDICTABLE** DISPLAY Unpredictable vmsa memory attribute with texcb value programmed to %{TEXCB} when tex remap is %{TEX_REMAP}. Fields:

TEXCB unsigned int

texcb value.

TEX_REMAP bool

use tex remap.

ArchMsg.Warning.warning_unpredictable_vtcr_t0sz_sl0

TAGS **UNPREDICTABLE** DISPLAY The combination of VTCR.TOSZ=%{TOSZ:d} and VTCR.SLO=%{SLO:d} is unpredictable. Fields:

SLO signed int

VTCR.SLO, starting level for stage 2 translation table walks.

TOSZ signed int

VTCR.TOSZ, required input address range.

ArchMsg.Warning.warning_unpredictable_write_DHCSR

TAGS **UNPREDICTABLE** DISPLAY **UNPREDICTABLE** set of bit-changes in DHCSR. Fields:

NEW unsigned int

data attempted to be written.

OLD unsigned int

previous value.

ArchMsg.Warning.warning_unpredictable_write_SHCSR

TAGS **UNPREDICTABLE** DISPLAY **UNPREDICTABLE** setting a pending bit to 1 for an exception with priority >= execution priority.

ArchMsg.Warning.warning_unsupported_access_to_memory_mapped_register

TAGS **UNPREDICTABLE** DISPLAY Access to memory mapped register is unsupported.

ArchMsg.Warning.warning_user_jmcr_access

TAGS **UNPREDICTABLE** DISPLAY User %{WRITE:(write|read)} access to JMCR is unpredictable. Fields:

WRITE unsigned int

access is write.

ArchMsg.Warning.warning_wcr_mask_and_bas

TAGS **UNPREDICTABLE** DISPLAY DBGWCR_EL1.MASK is non-zero and BAS is not 0xff.

ArchMsg.Warning.warning_wcr_mask_reserved

TAGS **UNPREDICTABLE** DISPLAY DBGWCR_ELn.MASK is set to a reserved value.

ArchMsg.Warning.warning_wcr_non_configuous_bas

TAGS **UNPREDICTABLE** DISPLAY DBGWCR_EL1.BAS has non-contiguous set of ones.

ArchMsg.Warning.warning_wvr_masked_bit_not_zero

TAGS **UNPREDICTABLE** DISPLAY DBGWVRn_EL1.VA contains a bit masked by DBGWCRn_EL1.MASK that is not zero.

ArchMsg.Why.why_branch_target_exception

A branch target exception has occurred. DISPLAY branch target exception: target pc={PC}, target opcode={OPCODE}. PRIMARY KEY PC. Fields:

BTTYPE unsigned int

pstate.BTYPE at time branch taken.

OPCODE unsigned int

opcode at branch target.

PC unsigned int

Virtual Address of branch target.

ArchMsg.Why.why_illegal_state

An illegal mode change has occurred. DISPLAY illegal mode change: %{MESSAGE}. Fields:

MESSAGE string

The message.

BRANCH_MISPREDICT

Simulating branch mispredict. Fields:

PC unsigned int

Origin address (or 0 if unavailable).

BRA_DIR

Direct branches, to immediate address. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISET enum

The instruction set of the branch instruction.

IS_COND bool

Indicates if this is a conditional branch.

IS_HINTED bool

Whether the conditional instruction is hinted.

IS_ISB bool

Whether the instruction is ISB.

IS_LINK bool

Indicates if this is a branch with link.

PC unsigned int

The address of the branch instruction.

TARGET_ISET enum

The instruction set after the branch.

TARGET_PC unsigned int

The address the instruction branches to.

BRA_INDIR

Indirect branches, perhaps to a register. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISET enum

The instruction set of the branch instruction.

IS_COND bool

Indicates if this is a conditional branch.

IS_HINTED bool

Whether the conditional instruction is hinted.

IS_ISB bool

Whether the instruction is ISB.

IS_LINK bool

Indicates if this is a branch with link.

PC unsigned int

The address of the branch instruction.

TARGET_ISET enum

The instruction set after the branch.

TARGET_PC unsigned int

The address the instruction branches to.

BRA_INDIR_CCFAIL

Indirect branches fail, perhaps to a register.

BRA_RET_CCFAIL

Branch return fail, perhaps to a register.

CACHE_MAINTENANCE_OP

Cache Maintenance Operation. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

DATA unsigned int

Specified MVA or Set/way.

FUNCTION enum

Clean / Invalidate.

SCOPE enum

Affected region.

SIDE enum

Inst / Data.

CCFAIL

Conditional instruction condition check fail. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CCFAIL_UNC

Unconditional instruction condition check fail. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CCPASS

Conditional instruction condition check pass. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CCPASS_UNC

Unconditional instruction condition check pass. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CODE_CACHE_MAINT

Code cache maintenance. Fields:

END_INCL unsigned int

Inclusive end address for VA or PA requests.

NS bool

Non-secure world for PA.

START unsigned int

Start address for VA or PA requests.

TYPE enum

Request type.

COMPILE_BLOCK_END

Last instruction of basic block translated. Fields:

VADDR unsigned int

Address of next instruction after this basic block.

COMPILE_BLOCK_START

First instruction of basic block translated. Fields:

VADDR unsigned int

Address of first instruction in basic block.

COMPILE_INST

ARM instruction compiled. Fields:

DISASS string

Disassembly of the instruction.

ISSET enum

The instruction set of this instruction.

ITSTATE unsigned int

The ITSTATE current for the instruction.

OPCODE unsigned int

The opcode of the instruction.

PC unsigned int

The address of the instruction.

SIZE unsigned int

The size of the instruction in bytes.

CONTEXTIDR

Context ID Register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

NS bool

Secure or nonsecure banked register is accessed.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The new value written.

CONTEXT_SYNC

Called for every context synchronization event. Fields:

ARCH_CSE bool

Architectural Context Synchronization Event.

PC unsigned int

The address of the synchronization instruction.

SYSREG_SYNC bool

System register synchronization event.

CORE_ENDIAN

Core BE8 Big-Endian state changed. Fields:

BE8 bool

Core BE8 Big-Endian state.

CORE_INFO

Static processor attributes. Only triggered by a call to DumpState(). Fields:

ARCH_PROFILE enum

The architecture profile of the core.

CLUSTER_ID unsigned int

The cluster ID of this processor.

CORE_NUM unsigned int

The number of this core in an MP processor.

DETAILED_CORE_FEATURES enum

Detailed core features as a bitfield. Enum values are bitmasks to extract the info from the field.

FPU_VERSION enum

The VFP version implemented by the core.

MEM_ARCH enum

The memory architecture of the core.

MIDR_EL1 unsigned int

MIDR_EL1 value.

NUM_CORES unsigned int

The number of cores in this MP processor.

QUANTUM_SIZE unsigned int

The default quantum size of the core (in ticks).

SECURITY_FEATURES bool

Does the core have security features?.

CORE_LOADS

Processor load accesses. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ACQREL enum

Is this an acquire/release.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DATA unsigned int

The data read or written.

ELEMENT_SIZE unsigned int

Width of each element.

ENDIANNESS enum

Byte ordering of this access.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

LOCK enum

Normal, exclusive or locked access.

MECID unsigned int

Memory Encryption Context ID, if applicable.

MECID2 unsigned int

Memory Encryption Context ID of the second page, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PAGE_SIZE unsigned int

Page size as $\log_2(\text{size in bytes})$.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK, 2=Exclusive Failed, 3=Unsupported.

SIZE unsigned int

Width of the access in bytes. Only required if DATA is not traced.

TRANS bool

Is this a translated access.

VADDR unsigned int

The virtual address of the access.

CORE_REGS

Changes of the core registers R0 to R14. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number, 0 to 14.

MODE enum

Bank of the register accessed.

OLD_VALUE unsigned int

The old value overwritten.

PHYS_ID enum

The physical register accessed.

VALUE unsigned int

The new value written to the register.

CORE_REGS64

Changes of the core registers X0..X30, SP_ELn. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ID enum

The register number, 0-30 for X0-X30, >=32 for SP_ELn.

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

CORE_REGS64_READ

General purpose AArch64 register has been read X0..X30, SP_ELn. Fields:

ID enum

The register number, 0-30 for X0-X30, >=32 for SP_ELn.

VALUE unsigned int

The value read from the register.

CORE_STALLED_TRANSITION

Detect when a core is prevented from progressing PChannel states. Fields:

CURRENT_STATE enum

Current power mode of core.

REASON string

The reason a transition was stalled, and what must be done to unblock it.

TARGET_STATE enum

Desired power mode of core after transition.

CORE_STORES

Processor store accesses. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ACQREL enum

Is this an acquire/release.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DATA unsigned int

The data read or written.

ELEMENT_SIZE unsigned int

Width of each element.

ENDIANNESS enum

Byte ordering of this access.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

LOCK enum

Normal, exclusive or locked access.

MECID unsigned int

Memory Encryption Context ID, if applicable.

MECID2 unsigned int

Memory Encryption Context ID of the second page, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PAGE_SIZE unsigned int

Page size as $\log_2(\text{size in bytes})$.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK, 2=Exclusive Failed, 3=Unsupported.

SIZE unsigned int

Width of the access in bytes. Only required if DATA is not traced.

TRANS bool

Is this a translated access.

VADDR unsigned int

The virtual address of the access.

CP14_READ

System Coprocessor register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP14 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc1 unsigned int

opcode 1.

opc2 unsigned int
opcode 2.

CP14_WRITE

System Coprocessor register write. Fields:

CORE_NUM unsigned int
Core number in a multi processor.

CRm unsigned int
CRm.

CRn unsigned int
CRn.

NS bool
Secure or nonsecure banked register is accessed.

REG_NAME string
Name of the CP14 register accessed.

SECURITY_STATE enum
The current security state.

UNDEF bool
The register accessed is undefined.

UPDATED_VALUE unsigned int
Updated value of the register now it has been written.

VALUE unsigned int
The new value written.

opc1 unsigned int
opcode 1.

opc2 unsigned int
opcode 2.

CP15_READ

System Coprocessor register read. Fields:

CORE_NUM unsigned int
Core number in a multi processor.

CRm unsigned int
CRm.

CRn unsigned int
CRn.

NS bool
Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP15 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc1 unsigned int

opcode 1.

opc2 unsigned int

opcode 2.

CP15_READ64

System Coprocessor register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP15 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc unsigned int

opcode 1.

CP15_WRITE

System Control Coprocessor register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP15 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc1 unsigned int

opcode 1.

opc2 unsigned int

opcode 2.

CP15_WRITE64

System Coprocessor register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP15 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc unsigned int

opcode 1.

CPSR

CPSR change. (In AArch64 PSTATE represented in SPSR format). Fields:

CORE_NUM unsigned int

Core number in a multi processor.

OLD_VALUE unsigned int

The old CPSR value.

UNKNOWN unsigned int

Bits within the register that have unknown value.

VALUE unsigned int

The new CPSR value.

CRYPTO_SPEC

Every crypto instruction speculatively executed.

CorePowerStateChange

Core power-up state has changed. DISPLAY Core powered %{PU:(up|down)}. Fields:

PU bool

Whether the core powered up.

DATA_CACHE_SET_ALLOC_TAGS

Set allocation tags from DC GVA instruction. Fields:

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

NSDESC unsigned int

The security state of the access.

PADDR unsigned int

The physical address of the access.

PAS enum

The physical address space of the page.

RESPONSE enum

0=Aborted, 1=OK.

SIZE unsigned int

The size of memory in bytes.

TAG unsigned int

The value of the tag being set.

TAGGING_TYPE enum

Type of Memory tagging (Physical or Virtual).

TAG_VADDR unsigned int

The virtual address of the Virtual Tag if Virtual tagging is used; otherwise 0.

VADDR unsigned int

Virtual address used for setting allocation tags.

DATA_CACHE_ZERO

Zero and invalidate cache line from DC ZVA instruction. Fields:

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

NSDESC unsigned int

The security state of the access.

PADDR unsigned int

The physical address of the access.

PADDR_TAG unsigned int

If tag checked, the physical address tag for the access.

PAS enum

The physical address space of the page.

RESPONSE enum

0=Aborted, 1=OK.

SIZE unsigned int

The size of zero'd memory in bytes.

TAG_CHECKED bool

Is this access tag checked.

VADDR unsigned int

Virtual address of the zero'd cache line.

DEBUG_EVENT

Hardware debug support event. Fields:

EVENT enum

Description of event.

VALUE unsigned int

data value.

DEBUG_TARGET_ISET

Hardware debug target ISA. Fields:

TARGET_ISET enum

The target instruction set of the debug state.

DMI_ALLOCATE

DMI allocated. Fields:

ACCESS_DIRECTION enum

READ / WRITE.

PADDR unsigned int

The physical address of the DMI.

PTR unsigned int

DMI range pointer.

SIDE enum

DSIDE / ISIDE.

SIZE unsigned int

The size of DMI.

VADDR unsigned int

The virtual address of the DMI..

DMI_HIT

DMI hit, access used DMI (only fire for small DMI regions i.e. less than a full page). Fields:

ACCESS_DIRECTION enum

READ / WRITE.

PADDR unsigned int

The physical address of the DMI.

SIDE enum

DSIDE / ISIDE.

SIZE unsigned int

The size of the DMI.

VADDR unsigned int

The virtual address of the DMI..

DMI_REVOKE

DMI revoked. Fields:

ACCESS_DIRECTION enum

READ / WRITE.

PADDR unsigned int

The physical address of the DMI.

SIDE enum

DSIDE / ISIDE.

SIZE unsigned int

The size of DMI.

VADDR unsigned int

The virtual address of the DMI..

END_COMPILE

Compilation end. Fields:

END_OF_PAGE_COUNT unsigned int

Number of basic blocks exited due to page boundary since START_COMPILE.

FETCHFAIL_COUNT unsigned int

Number of basic blocks exited due to ifetch failure START_COMPILE.

INST_COUNT unsigned int

Number of instructions compiled since START_COMPILE.

NONSEQ_COUNT unsigned int

Number of basic blocks exited due to non-sequential instructions since START_COMPILE.

PAGE_STRADDLE_COUNT unsigned int

Number of basic blocks exited due to unaligned instructions crossing page since START_COMPILE.

ESB

Error Synchronization Barrier hit. Fields:

ASYNC bool

ESB is async.

IMPLICIT bool

ESB is implicit.

EXCEPTION

Exception taken. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ESR unsigned int

The value assigned to ESR when exception is taken.

ESR_EC unsigned int

The Value assigned to ESR.EC when exception is taken.

IS_PHYSICAL bool

Physical or Virtual exception.

IS_TRAP bool

Exception coming from a trap.

LR unsigned int

The value assigned to the link register.

PC unsigned int

The location where the exception occurred.

PREFERRED_RETURN unsigned int

The preferred return address for the exception.

TAKEN_LOCALLY bool

Whether exception is taken locally(same exception level).

TARGET_ISET enum

The instruction set of the exception handler code.

TARGET_PC unsigned int

The address the exception branches to.

VECTOR enum

The exception vector.

VECTOR_OFFSET unsigned int

The offset of an exception vector from the base address.

EXCEPTION_END

Every exception completed.

EXCEPTION_INFO

Trace human-readable reason of the current exception for data/instruction aborts, floating point exceptions and unknown exceptions. Fields:

DESCRIPTION string

Reason for current exception.

ESR_EC unsigned int

The value of ESR.EC when exception is taken.

EXCEPTION_RAISE

Every exception raised.

EXCEPTION_RETURN

Branches on leaving exception. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISSET enum

The instruction set of the branch instruction.

IS_COND bool

Indicates if this is a conditional branch.

IS_HINTED bool

Whether the conditional instruction is hinted.

IS_ISB bool

Whether the instruction is ISB.

IS_LINK bool

Indicates if this is a branch with link.

PC unsigned int

The address of the branch instruction.

TARGET_ISET enum

The instruction set after the branch.

TARGET_PC unsigned int

The address the instruction branches to.

EXCEPTION_RETURN_PREBRANCH

Exception return event that must occur before PC and processor state have updated.

EXCEPTION_START

Every exception started.

FIQ_TAKEN

FIQ taken exception.

FP_EXCEPTIONS_SIGNALLED

Trace which FP exceptions were signalled as a part of the FP operations performed by an instruction. Fields:

FP_EXCEPTIONS_SIGNALLED_MASK unsigned int

Indicates which FP exceptions were signalled by an instruction. This uses the same encoding as the cumulative floating-point exception bitfields in FPSR.

FP_STATE

Floating point state. Fields:

FPSR unsigned int

FPSR register value.

MASK unsigned int

The mask indicating updated cumulative bits.

FREQ_CHANGED

Core frequency changed. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The instruction count of this core.

NEW_FREQ signed int

The new frequency of this core (expressed in Hz).

OLD_FREQ signed int

The old frequency of this core (expressed in Hz).

HLT

HLT instruction occurred. Fields:

IMM16 unsigned int

The 16-bit immediate value encoded in HLT instruction.

INST

Every instruction executed. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DEBUG_STATE bool

The instruction is executed in debug state.

DISASS string

Disassembly of instruction.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISSET enum

The current instruction set.

ITSTATE unsigned int

The current ITSTATE.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

MODE enum

The mode the core is in.

NS unsigned int

The core's non-secure bit.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

OPCODE unsigned int

The opcode of the instruction.

PADDR unsigned int

The physical address of the instruction.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the instruction.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the page.

PC unsigned int

The address of the instruction.

SECURITY_STATE enum

The core's security state.

SIZE unsigned int

The size of the instruction in bytes.

INST_END

Every instruction completed.

INST_START

Every instruction started. Fields:

DEBUG_STATE bool

The instruction is executed in debug state.

ISSET enum

The current instruction set.

MODE enum

The mode the core is in.

NS enum

The current Secure State.

PC unsigned int

The address of the conditional instruction.

SECURITY_STATE enum

The current Security State.

INST_STRADDLE

Instruction straddles boundary. Fields:

DEBUG_STATE bool

The instruction is executed in debug state.

ISSET enum

The current instruction set.

MODE enum

The mode the core is in.

NS enum

The current Secure State.

PC unsigned int

The address of the conditional instruction.

SECURITY_STATE enum

The current Security State.

IRQ_TAKEN

IRQ taken exception.

ISB

ISB instruction occurred. Fields:

PC unsigned int

The address of the ISB instruction.

LOAD_MULTIPLE_REGS_LIST

List of destination registers (not source) of a multiple registers load instruction. Fields:

REG string

Register to add into the list.

LOCAL_MONITOR

Local monitor activity. Fields:

PADDR unsigned int

Local Monitor Address.

State enum

State of the monitor (Open/Exclusive).

MAPPING_CREATE

A memory mapping has been created. This is an implementation detail of the model and so may change without notice. Fields:

ASID unsigned int

The ASID used during translation, or 0 if no ASID was used.

IS_VALID bool

Whether the mapping is valid or not.

PADDR unsigned int

Base physical address of region, or 0 if not known.

SIDE enum

I-side or D-side.

SIZE unsigned int

Size of the page, or 0 if not known.

VADDR unsigned int

Base virtual address of region.

VMID unsigned int

The VMID used during translation, or 0 if no VMID was used.

MAPPING_DESTROY

A memory mapping has been destroyed. This is an implementation detail of the model and so may change without notice. Fields:

ASID unsigned int

The ASID used during translation.

CAUSE enum

The cause of invalidation.

PADDR unsigned int

Base physical address of page.

SIDE enum

I-side or D-side.

SIZE unsigned int

Size of the page.

VADDR unsigned int

Base virtual address of page.

VMID unsigned int

The VMID used during translation.

MEMMAP_DEBUG_READ

Memory mapped reads to the debug registers. Fields:

ADDR unsigned int

address.

CORE_NUM unsigned int

Core number in a multi processor.

EXT bool

Whether access is from an external device (such as the DAP).

NS enum

Secure state of the access.

REG_NAME string

Name of the debug register accessed.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

MEMMAP_DEBUG_WRITE

Memory mapped writes to the debug registers. Fields:

ADDR unsigned int

address.

CORE_NUM unsigned int

Core number in a multi processor.

EXT bool

Whether access is from an external device (such as the DAP).

NS enum

Secure state of the access.

REG_NAME string

Name of the debug register accessed.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The new value written.

MMU_D128_TTB_READ

This event is triggered by reads caused by a D128 translation table walk. Fields:

ABORTED bool

Was the walk successful.

DATA_HIGH unsigned int

The data upper bits read.

DATA_LOW unsigned int

The data lower bits read.

IPA unsigned int

For stage 1, the IPA of the read.

LEVEL unsigned int

Translation table level.

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

OUTADDR unsigned int

The output physical address.

PADDR unsigned int

The physical address of the read.

PASmem enum

Physical Address Space of the accessed memory.

PASreq enum

Physical Address Space of the requested lookup.

REGIME_EL enum

Entry matches in this translation regime.

REQUESTER enum

The requester of this table walk.

SIDE enum

Inst / Data.

STAGE unsigned int

Translation stage.

MMU_D128_TTB_WRITE

This event is triggered by writes caused by a D128 translation table walk. Fields:

ABORTED bool

Was the walk successful.

DATA_HIGH unsigned int

The data upper bits written.

DATA_LOW unsigned int

The data lower bits written.

IPA unsigned int

For stage 1, the IPA of the write.

IS_HACDBS_ACCESS bool

If the access was due to HACDBS_CLEAN.

LEVEL unsigned int

Translation table level.

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

OUTADDR unsigned int

The output physical address.

PADDR unsigned int

The physical address of the write.

PASmem enum

Physical Address Space of the accessed memory.

PASreq enum

Physical Address Space of the requested lookup.

REGIME_EL enum

Entry matches in this translation regime.

SIDE enum

Inst / Data.

STAGE unsigned int

Translation stage.

MMU_TRANS

Address translation information. Fields:

ASID unsigned int

Address space identifier.

CORE_NUM unsigned int

Core number in a multi processor.

Hyp bool

Entry matches in Hyp state only.

INNERCACHE_RA bool

Is the inner cache allocate on read.

INNERCACHE_TRANSIENT bool

Is the inner write-through transient.

INNERCACHE_TYPE enum

Inner Caching scheme (NC/MB/WA).

INNERCACHE_WA bool

Is the inner cache allocate on write.

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MEMTYPE enum

Memory type.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC enum

Is secure side supposed to access secure or nonsecure memory.

OUTERCACHE_RA bool

Is the outer cache allocate on read.

OUTERCACHE_TRANSIENT bool

Is the outer write-through transient.

OUTERCACHE_TYPE enum

Outer Caching scheme (NC/MB/WA).

OUTERCACHE_WA bool

Is the outer cache allocate on write.

PADDR unsigned int

Physical address of the access.

PAGESIZE unsigned int

Page size as log2(size).

PAS enum

PAS supposed to be accessed is secure or nonsecure memory or Root/Realm if RME is implemented.

SH enum

Shareability.

SIDE enum

Inst / Data.

STAGE1_PERM unsigned int

Stage 1 permissions, [5:3] Privileged access (XWR) [2:0] User access (XWR).

STAGE2_PERM unsigned int

Stage 2 permission mask [5:3] Privileged access (XWR) [2:0] User access (XWR).

VADDR unsigned int

Virtual address of the access.

VMID unsigned int

Virtual machine identifier.

XS bool

XS Attribute.

nG enum

Flag indicating whether ASID will be matched.

MMU_TTB_READ

This event is triggered by reads caused by a translation table walk. Fields:

ABORTED bool

Was the walk successful.

DATA unsigned int

The data read.

DESC_KIND enum

The encoding scheme of the descriptors used for this walk.

IPA unsigned int

For stage 1, the IPA of the read.

LEVEL unsigned int

Translation table level.

LPAAE bool

Is this for an LPAE translation (*DEPRECATED*, use DESC_KIND).

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSmem bool

Non secure memory access.

NSreq bool

Non secure request.

OUTADDR unsigned int

The output physical address.

PADDR unsigned int

The physical address of the read.

PASmem enum

Physical Address Space of the accessed memory.

PASreq enum

Physical Address Space of the requested lookup.

REGIME_EL enum

Entry matches in this translation regime.

REQUESTER enum

The requester of this table walk.

SIDE enum

Inst / Data.

STAGE unsigned int

Translation stage.

MMU_TTB_WRITE

This event is triggered by writes caused by a translation table walk. Fields:

ABORTED bool

Was the walk successful.

DATA unsigned int

The data written.

DESC_KIND enum

The encoding scheme of the descriptors used for this walk.

IPA unsigned int

For stage 1, the IPA of the write.

IS_HACDBS_ACCESS bool

If the access was due to HACDBS_CLEAN.

LEVEL unsigned int

Translation table level.

LPAE bool

Is this for an LPAE translation (*DEPRECATED*, use DESC_KIND).

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSmem bool

Non secure memory access.

NSreq bool

Non secure request.

OUTADDR unsigned int

The output physical address.

PADDR unsigned int

The physical address of the write.

PASmem enum

Physical Address Space of the accessed memory.

PASreq enum

Physical Address Space of the requested lookup.

REGIME_EL enum

Entry matches in this translation regime.

SIDE enum

Inst / Data.

STAGE unsigned int

Translation stage.

MODE_CHANGE

Mode change. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

MODE enum

The new mode.

NON_SECURE enum

The core's new non-secure bit.

OLD_MODE enum

The old mode.

SECURITY_STATE enum

The new security state.

NOT_TAG_CHECKED_INST

MTE Tag not checked for current instruction.

PERIODIC

Called for every quantum. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The instruction count of this CPU.

PC unsigned int

The address of the next instruction to be executed on this CPU.

PMU_COUNTER_OVERFLOW

PMU counter overflow. Fields:

COUNTER_GROUP enum

Counter group.

COUNTER_RANGE enum

Counter range.

EVENT_ID unsigned int

Event Identifier, field is valid only for Event counter group.

INDEX unsigned int

Counter index (as selected by PMSELR), field is valid only for Event/Cycle counter groups.

INTERRUPT bool

Is interrupt enabled on overflow for this counter.

POTINDEX_UPDATE

Value of POT index for current pc changed. Fields:

NEW_POTINDEX unsigned int

New POTIndex.

OLD_POTINDEX unsigned int

Old POTIndex.

PREFETCH_MEMORY64

Prefetch from PRFM or PRFUM instructions. Fields:

PRFOP enum

Prefetch hint.

VADDR unsigned int

Virtual address of the location that should be prefetched.

PRELOAD64_REQUEST

Preload requests from PRFM or PRFUM instructions. Fields:

PRFOP enum

Prefetch hint.

VADDR unsigned int

Virtual address of the location that should be prefetched.

PRELOAD_DATA

Data preload from PLD instruction. Fields:

VADDR unsigned int

Virtual address of the data that should be preloaded.

PRELOAD_INST

Instruction preload from PLI instruction. Fields:

VADDR unsigned int

Virtual address of the instruction that should be preloaded.

PRE_CORE_LOAD

Trace just before a core load. Fields:

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

PRE_CORE_STORE

Trace just before a core store. Fields:

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

PRE_TTB_READ

This event is triggered before reads caused by a translation table walk. Fields:

ADDR unsigned int

The physical address of the read.

PSTATE

PSTATE fields update. Fields:

ALLINT unsigned int

All interrupt mask bit.

BTYPE unsigned int

Branch target identification bit.

CORE_NUM unsigned int

Core number in a multi processor.

DAIF unsigned int

Exception mask bits.

DIT unsigned int

Data Independent Timing bit.

EL unsigned int

Current Exception level field.

EXLOCK unsigned int

Exception return state lock bit.

IL unsigned int

Illegal Execution state bit.

NZCV unsigned int

Condition flags.

PACM unsigned int

PACM bit.

PAN unsigned int

Privileged Access Never state bit.

PM unsigned int

PMU exception mask bit.

PPEND unsigned int

PMU exception pending bit.

SM unsigned int

StreamingMode status bit.

SP unsigned int

Stack pointer register selection bit.

SS unsigned int

Software Step bit.

SSBS unsigned int

Speculative Store Bypass Safe bit.

TCO unsigned int

Tag Check Override bit.

TOUCHED unsigned int

Mask of the fields that have been touched since this trace has last been emitted. Note that touched does not necessarily mean the value changed. The index of the field is the bit index in the mask plus the index of this field plus one.

UAO unsigned int

User Access Override bit.

UINJ unsigned int

Undefined Instruction exceptions bit.

ZA unsigned int

ZA storage and ZT0 register enable bit.

nRW unsigned int

Current Execution state bit.

PSTATE_PACTIVE_OPMODE_TRANSITION

Trace pactive and Opmode mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Operating mode after the transition.

OLD_STATE enum

Operating mode before the transition.

PSTATE_PACTIVE_OPMODE_TRANSITION_DENIED

Trace the denied pactive and Opmode mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Operating mode after the transition.

OLD_STATE enum

Operating mode before the transition.

REASON string

Reason for the transition being denied.

PSTATE_PACTIVE_POWER_TRANSITION

Trace pactive and power mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Power mode after the transition.

OLD_STATE enum

Power mode before the transition.

PSTATE_PACTIVE_POWER_TRANSITION_DENIED

Trace the denied pactive and power mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Power mode after the transition.

OLD_STATE enum

Power mode before the transition.

REASON string

Reason for the transition being denied.

PSTATE_PACTIVE_THREAD_OPMODE_TRANSITION

Trace pactive and Opmode mode transitions for threads on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Operating mode after the transition.

OLD_STATE enum

Operating mode before the transition.

QUANTUM_START

Called at the start of every quantum.

RANGE_PRELOAD_REQUEST

Preload requests from RPRFM. Fields:

COUNT unsigned int

Number of blocks of data to access.

LENGTH unsigned int

Number of contiguous bytes to be accessed in each block.

OP unsigned int

Range prefetch operation .

REUSE unsigned int

Maximum number of bytes to be accessed before next RPRFM .

STRIDE signed int

Stride to advance after accessing LENGTH bytes.

VADDR unsigned int

Base virtual address of range prefetch request.

RUN_STATE

Run state transition. Fields:

INST_COUNT unsigned int

Ticks count at point of transition.

NEW enum

New run state.

OLD enum

Old run state.

SEMIHOSTING_CALL

Call of a semihost function occurred. Fields:

PC unsigned int

The program counter after the semihosting call.

SEMIHOSTING_PRECALL

About to call semihost. Fields:

EL2 **bool**

Use EL2 translation regime.

NS **bool**

Is Non-Secure.

REG_WIDTH **unsigned int**

The current register width in bytes.

SIGNAL

External signal state change. Fields:

SIGNAL **enum**

Signal that changed.

STATE **bool**

Signal asserted state.

SOFTWARE_STEP

Return the Debug Software Step state. Fields:

STATE **enum**

Software Step state.

SPSR

SPSR change. Fields:

CORE_NUM **unsigned int**

Core number in a multi processor.

MODE **enum**

Which of the banked SPSR registers is written.

OLD_VALUE **unsigned int**

The old SPSR value.

VALUE **unsigned int**

The new SPSR value.

START_COMPILE

Compilation started. Fields:

VADDR **unsigned int**

Instruction where compilation begins.

STG_MULTIPLE_MICRO_OPS_ADDRESS_RANGE

Address range for MTE stores with multiple uops. Fields:

ADDR_END **unsigned int**

Final address.

ADDR_START unsigned int

Starting address.

STORE_MULTIPLE_REGS_ADDRESS_RANGE

Address range for stores with multiple registers. Fields:

ADDR_END unsigned int

Final address.

ADDR_START unsigned int

Starting address.

SUBOPTIMAL_LDST_RETIRED

suboptimal load/store. Fields:

ACCESS_SIZE unsigned int

Log2 of the access width in bytes used for alignment checking (1: half-word, 2: word ...).

OPTIMAL_ALIGN_SELECTOR unsigned int

Selector to decide the optimal alignment size (1: access width, 2: 4-bytes, 3: 8-bytes, ..., 12: 4-kbytes).

VADDR unsigned int

The virtual address of the access.

SYNC

Called for every synchronization, that is every quantum break. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The instruction count of this CPU.

LOCAL_QUANTUM unsigned int

The local quantum of this CPU.

LOCAL_TIME unsigned int

The local time of this CPU.

SYNC_REASON enum

Reason for sync.

SYSCALL

System call instruction executed. Fields:

IMM unsigned int

Immediate value of the system call instruction.

TYPE enum

System call type.

VADDR unsigned int

Instruction that caused the system call.

SYSREG_READ128

System Coprocessor 128 bit register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

REGNUM enum

Internal register number.

REGNUM_PHYS enum

Internal register number physically accessed (e.g. in presense of VHE redirection).

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_READ64

System Coprocessor register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REGNUM enum

Internal register number.

REGNUM_OPERAND enum

Register number corresponding to the instruction operand.

REGNUM_PHYS enum

Internal register number physically accessed (e.g. in presense of VHE redirection).

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_UPDATE128

Triggers when the system updates a register. Fields:

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG enum

Register number.

UNKNOWN unsigned int

Bits of the register which became unknown.

VALUE unsigned int

Value written to the register.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_UPDATE32

Triggers when the system updates a register. Fields:

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG enum

Register number.

UNKNOWN unsigned int

Bits of the register which became unknown.

VALUE unsigned int

Value written to the register.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_UPDATE64

Triggers when the system updates a register. Fields:

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG enum

Register number.

UNKNOWN unsigned int

Bits of the register which became unknown.

VALUE unsigned int

Value written to the register.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int
opcode 2.

SYSREG_WRITE128

System Coprocessor 128 bit register write. Fields:

CORE_NUM unsigned int
Core number in a multi processor.

CRm unsigned int
CRm.

CRn unsigned int
CRn.

REGNUM enum
Internal register number.

REGNUM_PHYS enum
Internal register number physically accessed (e.g. in presense of VHE redirection).

UNDEF bool
The register accessed is undefined.

UPDATED_VALUE unsigned int
Updated value of the register now it has been written.

VALUE unsigned int
The new value written.

opc unsigned int
opcode 1.

opc0 unsigned int
opcode 0.

opc2 unsigned int
opcode 2.

SYSREG_WRITE64

System Coprocessor register write. Fields:

CORE_NUM unsigned int
Core number in a multi processor.

CRm unsigned int
CRm.

CRn unsigned int
CRn.

NS bool
Secure or nonsecure banked register is accessed.

REGNUM enum

Internal register number.

REGNUM_OPERAND enum

Register number corresponding to the instruction operand.

REGNUM_PHYS enum

Internal register number physically accessed (e.g. in presense of VHE redirection).

UNDEF bool

The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

UNALIGNED_LDST_RETIRED

Processor unaligned load/store. Fields:

ACCESS_SIZE unsigned int

Log2 of the access width in bytes used for alignment checking (1=>half-word, 2->word ...).

NUMBER_OF_BEATS unsigned int

Number of accesses (beats) in this burst (Total size in bytes is ACCESS_SIZE * NUMBER_OF_BEATS).

VADDR unsigned int

The virtual address of the access.

VFP_D_REGS

VFP/NEON D 64 bit register write. Fields:

ALIASING enum

Type of register aliasing used.

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a write to the corresponding bit in the register. For example, 0x1 would indicate a write to VALUE[0].

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

VFP_D_REGS_READ

VFP/NEON D 64 bit register read. Fields:

ID unsigned int

The register number.

VALUE unsigned int

The current value of the register.

VFP_Q_REGS

VFP/NEON Q 128 bit register write. Fields:

ALIASING enum

Type of register aliasing used.

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a write to the corresponding byte in the register. For example, 0x1 would indicate a write to VALUE[7:0].

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

VFP_Q_REGS_READ

VFP/NEON Q 128 bit register read. Fields:

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a read to the corresponding byte in the register. For example, 0x1 would indicate a read to VALUE[7:0].

VALUE unsigned int

The current value of the register.

VFP_SYS_REGS

Writes to the VFP/NEON units system registers. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ID enum

Which VFP system register is written.

MASK unsigned int

The mask indicating updated cumulative bits.

OLD_VALUE unsigned int

The register's old value overwritten.

VALUE unsigned int

The new value written to the VFP system register.

VFP_S_REGS

VFP/NEON S 32 bit register write. Fields:

ALIASING enum

Type of register aliasing used.

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a write to the corresponding bit in the register. For example, 0x1 would indicate a write to VALUE[0].

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

VFP_S_REGS_READ

VFP/NEON S 32 bit register read. Fields:

ID unsigned int

The register number.

VALUE unsigned int

The current value of the register.

VIRTUAL_TAG_ALLOCATED

Virtual tag information when allocated. Fields:

TAG_VADDR unsigned int

Virtual tag virtual address.

VADDR unsigned int

Data Virtual address.

VIRTUAL_TAG unsigned int

Virtual tag value.

WAYPOINT

Signals end of basic block execution. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ISET enum

Origin instruction set.

IS_COND bool

Indicates if this is a conditional waypoint.

PC unsigned int

Origin address (or 0 if unavailable).

TAKEN bool

Indicates if this waypoint was taken.

TARGET unsigned int

Destination address.

TARGET_ISET enum

Destination instruction set.

WFE_END

WFE ended. Fields:

INST_COUNT unsigned int

Ticks count when leaving WFE.

WFE_EVENT_REGISTER

WFE event register status: set/clear, reason. Fields:

INST_COUNT unsigned int

Ticks count.

REASON enum

Reason for set/clear. Only REASON==0 (Cleared by WFE) clears the bit, all other reasons set it.

WFE_IGNORED

WFE ignored. Fields:

EVENT bool

This WFE was ignored because the event register was set.

INST_COUNT unsigned int

Ticks count when ignoring WFE.

TRAPPED bool

This WFE was trapped.

WFE_START

WFE entered. Fields:

INST_COUNT unsigned int

Ticks count when entering WFE.

WFI_END

WFI ended. Fields:

INST_COUNT unsigned int

Ticks count when leaving WFI.

WFI_IGNORED

WFI ignored. Fields:

DISABLED bool

This WFI was ignored because WFI is disabled.

INST_COUNT unsigned int

Ticks count when ignoring WFI.

REASON enum

specifies reason why WFI trace was ignored.

TRAPPED bool

This WFI was trapped.

WFI_START

WFI entered. Fields:

INST_COUNT unsigned int

Ticks count when entering WFI.

WFI_WAKEUP

WFI wakeup. Fields:

INST_COUNT unsigned int

Ticks count when WFI wakeup occurred.

REASON enum

Reason for wakeup.

2.64 ARM_Neoverse-N1

This section describes the trace sources.

AA64_ASE_SVE_REGS

Modification of AArch64 Advanced SIMD or SVE Vector Registers (via SIMD or Scalar operations), SVE Predicate Registers and ZT0. Fields:

ID **enum**

The register identifier.

MASK **unsigned int**

Bitmask of the register to signal the modified bits in the VALUE field.

SM **bool**

Whether the PE is in Streaming Mode.

VALUE **unsigned int**

Value of AArch64 vector register.

ASYNC_MEMORY_FAULT

Context ID Register write. Fields:

FAULT **unsigned int**

Fault status present in ESR.ISS encoding for Serror in ARCH64 and present in DFSR encoding for external abort format in ARCH32.

PADDR **unsigned int**

Physical Address (or 0 if unavailable).

VADDR **signed int**

Virtual Address (or 0 if unavailable).

ATOMIC_END_ACCESS

Bus trace access for atomics. Fields:

ACCESS_FAIL **bool**

Memory access failed.

ADDR **unsigned int**

The virtual address of the access.

ATTR **unsigned int**

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

COMPARE_VALUE **unsigned int**

Compare value for CAS.

LOAD_VALUE unsigned int

Loaded values.

NSDESC unsigned int

The security state of the access.

OPERAND_VALUE unsigned int

Operation's operand.

OPERATION enum

Operation type.

PADDR unsigned int

The physical address of the access.

PRIV bool

Is this a privileged access?.

ATOMIC_START_ACCESS

Bus trace access for atomics. Fields:

ADDR unsigned int

The virtual address of the access.

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

COMPARE_VALUE unsigned int

Compare value for CAS.

NSDESC unsigned int

The security state of the access.

OPERAND_VALUE unsigned int

Operation's operand.

OPERATION enum

Operation type.

PADDR unsigned int

The physical address of the access.

PRIV bool

Is this a privileged access?.

ArchMsg.Error.exit_code

DISPLAY exit code: %{EXITCODE} component kind: %{KIND} reason: %{REASON}. Fields:

EXITCODE signed int

Exit code to be returned.

KIND string

Component kind that invoked the exit code trace.

REASON string

Reason why the exit code trace was invoked (optional).

ArchMsg.Warning.IgnoredWarmResetRequest

Event fires when software attempts to request warm reset through write to deprecated DBGPRCR.CWRR.

ArchMsg.Warning.RAS_ErrselrUnpredictableState

DISPLAY ERRSELR is in unpredictable state due to value greater than number_of_error_records written to it.

ArchMsg.Warning.RAS_UnpredictableStateWrite

DISPLAY %{REG_NAME} write ignored due to unpredictable ERRSELR state. Fields:

REG_NAME string

Name of register being written.

ArchMsg.Warning.ReportedRevisionRegisterRead

DISPLAY Reported revision and patch do not match that of the cpu. Fields:

ORIGINAL_VALUE unsigned int

The value the register would have based on the modelled revision.

REGISTER string

The name of the register that has ben read.

REPORTED_VALUE unsigned int

The value the register has been overridden to.

ArchMsg.Warning.Unpredictably Indexed PM Event Register

TAGS UNPREDICTABLE DISPLAY Event %{IsDirect:(Selector value|Register)} %{SEL} is >= %{N}. Fields:

IsDirect unsigned int

Direct Access rather than indirect.

N unsigned int

Number of Accessible Registers.

SEL unsigned int

Selector.

ArchMsg.Warning.cache_contents_unknown

A cache has been activated without being invalidated and may contain **UNKNOWN** content. DISPLAY %{SIDE:(D|I)}-Cache was enabled in %{REGIME:(SECURE|NON_SECURE|HYP)} regime but was not invalidated since power-on: cache lines could contain **UNKNOWN** content %{SIDE:(Note: D-side caches must invalidate SECURE lines even if the cache is only used by NON_SECURE code in order to ensure that unexpected dirty lines tagged SECURE are not naturally evicted)}. Fields:

REGIME unsigned int

Current translation regime.

SIDE unsigned int

0 d-side cache or unified cache, 1 i-side.

ArchMsg.Warning.dcimvac_matches_watchpoint

TAGS **UNPREDICTABLE** DISPLAY Watchpoints matching an AArch32 DCIMVA are implementation defined.

ArchMsg.Warning.decode_fieldmismatch

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with different fields %{FIELD1} and %{FIELD2}. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_fracbitsnegative

TAGS **UNPREDICTABLE** DISPLAY VCVT Instruction is unpredictable with a negative number of fraction bits.

ArchMsg.Warning.decode_initblock

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable in an IT block.

ArchMsg.Warning.decode_initblocknotlast

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable in an IT block when not the last.

ArchMsg.Warning.decode_invalid_condition

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable when the condition is not AL.

ArchMsg.Warning.decode_invalidfieldcombination

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with the values in %{FIELD1} and %{FIELD2}. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_invalidvalue

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with the value in %{FIELD}. Fields:

FIELD enum

Field name.

ArchMsg.Warning.decode_mem_hint_unallocated

TAGS **UNPREDICTABLE** DISPLAY There is no memory hint allocated to this bit pattern.

ArchMsg.Warning.decode_registermatch

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with %{FIELD1} and %{FIELD2} as the same register. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_registermismatch

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with %{FIELD1} and %{FIELD2} as different registers. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_registeroutofrange

TAGS **UNPREDICTABLE** DISPLAY Floating-point instruction is unpredictable indexing beyond the end of the register bank.

ArchMsg.Warning.decode_sbzsbo

TAGS **UNPREDICTABLE** DISPLAY Reserved bits in the instruction are not canonical.

ArchMsg.Warning.decode_transactiontoobig

TAGS **UNPREDICTABLE** DISPLAY Attempt to perform a memory transaction of more than 128 bytes.

ArchMsg.Warning.decode_unpred_other

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable.

ArchMsg.Warning.decode_unpreduseofpc

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable when %{FIELD} is PC. Fields:

FIELD enum

Field name.

ArchMsg.Warning.decode_unpreduseofr13

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable when %{FIELD} is R13. Fields:

FIELD enum

Field name.

ArchMsg.Warning.decode_unpreduseofr13inreglist

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with R13 in register-list.

ArchMsg.Warning.decode_useofd16orhigher

TAGS **UNPREDICTABLE** DISPLAY Attempt to use register D16 or higher where it is not allowed.

ArchMsg.Warning.decode_writebackandbaseinlist

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with write back when the base register is in the transfer list.

ArchMsg.Warning.decode_zeroregistersinlist

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable when no registers are to be transferred.

ArchMsg.Warning.recursive_branch

An instruction is performing a branch that targets the same instruction. If this was intended then a WFI might be more effective DISPLAY recursive branch detected at PC=%{PC}. PRIMARY KEY PC. Fields:

PC unsigned int

Address of instruction causing the branch.

ArchMsg.Warning.recursive_exception

An instruction has generated an exception that targets the same instruction. DISPLAY recursive %{TYPE} exception detected at PC=%{PC} (CPSR=%{CPSR}). PRIMARY KEY PC. Fields:

CPSR unsigned int

Processor state.

PC unsigned int

Address of instruction causing the exception.

TYPE enum

Type of exception.

ArchMsg.Warning.reentrant_vector_catch

TAGS **UNPREDICTABLE** DISPLAY Vector catch debug event on Prefetch or Data abort vector %{ADDR}. Fields:

ADDR unsigned int

vector address.

ArchMsg.Warning.reserved_it_state

TAGS **UNPREDICTABLE** DISPLAY Execution with reserved IT state %{ITSTATE}. Fields:

ITSTATE unsigned int

reserved IT state.

ArchMsg.Warning.tlb_contents_unknown

Translation has been enabled without invalidating the TLB. DISPLAY Translation was enabled in the %{PAS:(Secure|Non-secure|Root|Realm)} EL%{EL:d} translation regime (vmid=%{VMID:x}) without invalidating %{INVALIDITY:(|d-side |i-side |)}entries since power-on: %{INVALIDITY:(|D|I|)}TLB could contain **UNKNOWN** content. Fields:

EL unsigned int

Exception level of translation regime.

INVALIDITY unsigned int

Which TLB may be invalid (if TLBs are separate).

NON_SECURE bool

Whether the translation regime is non-secure.

PAS unsigned int

Physical address space of translation regime.

VMID unsigned int

Current VMID, if applicable.

ArchMsg.Warning.unknown_DLR_DSPSR

Exiting debug state with unknown DLR or DSPSR.

ArchMsg.Warning.unknown_ELR_SPSR

Returning from debug exception with unknown ELR or SPSR.

ArchMsg.Warning.unpred_FAR_on_LRCPC3_non_atomic_mem_access

TAGS **UNPREDICTABLE** Unpredictable FAR/FSTATUS on LRCPC3 pair instructions resulting in two separate memory accesses
DISPLAY Unpredictable FAR on LRCPC3 non atomic memory access at %{ADDR1} with FSTATUS %{FSTATUS1} or %{ADDR2} with FSTATUS %{FSTATUS2}. Fields:

ADDR1 unsigned int

Possible FAR address 1.

ADDR2 unsigned int

Possible FAR address 2.

FSTATUS1 unsigned int

Possible fault status value 1.

FSTATUS2 unsigned int

Possible fault status value 2.

ArchMsg.Warning.unpred_async_tag_check_fail_on_sve_speculative_load_do_update_ffr

Unpredictable set to FALSE of the FFR element associated to SVE speculative load with an async tag check fail TAGS **UNPREDICTABLE**.

ArchMsg.Warning.unpredictable_a32_breakpoint

TAGS **UNPREDICTABLE** DISPLAY Unpredictable breakpoint on a A32 instruction at %{ADDR1} Offset %{OFFSET} BAS %{BAS} for a %{IS_ADDRESS_MISMATCH_BREAKPOINT:(Mismatch|Match)} Breakpoint number %{N:d} Breakpoint address %{ADDR2}. Fields:

ADDR1 unsigned int

Address.

ADDR2 unsigned int

Breakpoint address.

BAS unsigned int

BAS field.

IS_ADDRESS_MISMATCH_BREAKPOINT bool

Is a Mismatch breakpoint, not a Match breakpoint.

N unsigned int

Breakpoint number.

OFFSET unsigned int

Always has a value of zero.

ArchMsg.Warning.unpredictable_banked_register_access

TAGS **UNPREDICTABLE** DISPLAY Access to banked register (R=%{R:d} SYSm=%{SYSM:2x}) is unpredictable. Fields:

R unsigned int

R bit.

SYSM unsigned int

SYSm.

ArchMsg.Warning.unpredictable_ccfail_undef

TAGS **UNPREDICTABLE** DISPLAY An instruction %{OPCODE} at VA=%{VADDR} failed its condition codes check but would otherwise have caused a non-data-dependent trap or undefined exception. Behaviour is implementation-defined choice of exception or nop. Fields:

OPCODE unsigned int

opcode of the instruction.

VADDR unsigned int

Virtual address of the instruction being executed.

ArchMsg.Warning.unpredictable_ici_bits

TAGS **UNPREDICTABLE** DISPLAY ICI bits do not apply or are architecturally unpredictable for instruction.

ArchMsg.Warning.unpredictable_t32_breakpoint

TAGS **UNPREDICTABLE** DISPLAY Unpredictable breakpoint on a T32 instruction at %{ADDR1} Offset %{OFFSET} BAS %{BAS} for a %{IS_ADDRESS_MISMATCH_BREAKPOINT:(Mismatch|Match)} Breakpoint number %{N:d} Breakpoint address %{ADDR2}. Fields:

ADDR1 unsigned int

Address.

ADDR2 unsigned int

Breakpoint address.

BAS unsigned int

BAS field.

IS_ADDRESS_MISMATCH_BREAKPOINT bool

Is a Mismatch breakpoint, not a Match breakpoint.

N unsigned int

Breakpoint number.

OFFSET unsigned int

Indicates which halfword of a 32-bit Thumb instruction the breakpoint is set on. 0 for the first halfword or if the instruction is 16 bits wide, or 1 for the second halfword.

ArchMsg.Warning.unpredictable_t32_vector_catch

TAGS **UNPREDICTABLE** DISPLAY Unpredictable Vector catch exception on a T32 instruction at **%{ADDR}** Offset **%{OFFSET}**. Fields:

ADDR unsigned int

Address.

OFFSET unsigned int

Indicates which halfword of a 32-bit Thumb instruction the vector catch exception is set on. 0 for the first halfword or if the instruction is 16 bits wide, or 1 for the second halfword.

ArchMsg.Warning.unpredictable_watchpoint_far

TAGS **UNPREDICTABLE** DISPLAY The value for FAR/EDWAR for hitting this watchpoint may be anywhere in the range **%{LOWER_BOUND}** to **%{UPPER_BOUND}**. Fields:

LOWER_BOUND unsigned int

the lowest address accessed by the instruction that triggered the watchpoint.

UPPER_BOUND unsigned int

the highest watchpointed address accessed by that instruction.

ArchMsg.Warning.unpredictable_watchpoint_far_isscmp

DISPLAY The value for FAR/EDWAR for hitting this watchpoint may be anywhere in the range **%{LOWER_BOUND}** to **%{UPPER_BOUND}**. Fields:

LOWER_BOUND unsigned int

the lowest address accessed by the instruction that triggered the watchpoint.

UPPER_BOUND unsigned int

the highest watchpointed address accessed by that instruction.

ArchMsg.Warning.warning_AdvSIMDExpandImmUnexpectedZero

TAGS **UNPREDICTABLE** DISPLAY AdvSIMDExpandImm may treat this immediate value as **UNPREDICTABLE**.

ArchMsg.Warning.warning_ConditionalSMC

TAGS **UNPREDICTABLE** DISPLAY SMC instruction has **UNPREDICTABLE** effects when conditional and when combined with traps.

ArchMsg.Warning.warning_TLBID_out_of_range

TLBID out of range in TLBI according to the configuration of NIS, NVIS, NOS and NVOS. Fields:

NIS unsigned int

NIS.

NOS unsigned int

NOS.

NVIS unsigned int

NVIS.

NVOS unsigned int

NVOS.

TLBID unsigned int

The TLBID value used in the TLBI instruction.

ArchMsg.Warning.warning_TTL_or_TG_mismatches_current_translation_granule

TTL or TG mismatches current translation granule in TLBI VA/RVA/IPA/RIPA. Fields:

CURRENT_TG unsigned int

Current TG as indicated by TCR_ELx or VTCR_EL2.

TLBI_TG unsigned int

TG (or TTL[3:2]) used in TLBI instruction.

ArchMsg.Warning.warning_access_crosses_page_boundary

TAGS **UNPREDICTABLE** DISPLAY Access crosses page boundary.

ArchMsg.Warning.warning_access_crosses_page_boundary_has_fault

TAGS **UNPREDICTABLE** DISPLAY Access that crosses page boundary has fault(s).

ArchMsg.Warning.warning_access_crosses_page_boundary_has_fault_on_both_pages

TAGS **UNPREDICTABLE** DISPLAY Access that crosses page boundary has a fault on each page.
Fields:

ADDR1 unsigned int

FAR address on first page.

ADDR2 unsigned int

FAR address on second page.

FSTATUS1 unsigned int

Fault status on first page.

FSTATUS2 unsigned int

Fault status on second page.

ArchMsg.Warning.warning_access_crosses_page_boundary_spanning_different_memory

TAGS **UNPREDICTABLE** DISPLAY Access at address %{ADDR} crosses page from MEM_TYPE %{MEMTYPE_PAGE1} to %{MEMTYPE_PAGE2}. Fields:

ADDR unsigned int

address of access crossing page.

MEMTYPE_PAGE1 signed int

First Page's memory type (0-Normal, 1-Non-Normal).

MEMTYPE_PAGE2 signed int

Second Page's memory type (0-Normal, 1-Non-Normal).

ArchMsg.Warning.warning_access_wraps_around_memory

TAGS **UNPREDICTABLE** DISPLAY Access wraps around memory in %{WIDTH:d}bit mode.

Fields:

WIDTH unsigned int

Address width.

ArchMsg.Warning.warning_bcr_linking_status

TAGS **UNPREDICTABLE** DISPLAY DBGBCR.{BT2,BT} combination gives unpredictable behaviour.

ArchMsg.Warning.warning_bcr_mask_reserved

TAGS **UNPREDICTABLE** DISPLAY DBGBCR_ELn.MASK is set to a reserved value.

ArchMsg.Warning.warning_bvr_masked_bit_not_zero

TAGS **UNPREDICTABLE** DISPLAY DBGBVRn_EL1.VA contains a bit masked by DBGBCRn_EL1.MASK that is not zero.

ArchMsg.Warning.warning_ccsidr_unimplemented_level

TAGS **UNPREDICTABLE** DISPLAY CCSIDR read while CSSELR %{CSSELR:x} points to an unimplemented cache level. Fields:

CSSELR unsigned int

current effective value of CSSELR.

ArchMsg.Warning.warning_change_to_ns_when_tge_set

TAGS **UNPREDICTABLE** DISPLAY Attempting to change NS to 1 when HCR.TGE = 1.

ArchMsg.Warning.warning_contiguous_bit_check_abort

TAGS **UNPREDICTABLE** DISPLAY An abort occurred whilst attempting to check TLB entry is contiguous at %{ENTRY_ADDR}. Fields:

ENTRY_ADDR unsigned int

address of conflicting TLB entry.

ArchMsg.Warning.warning_contiguous_bit_error

TAGS **UNPREDICTABLE** DISPLAY TLB entry at %{ENTRY_ADDR} was not contiguous with entry at %{CONTIG_ADDR} Contents are %{ENTRY_DATA} but expected %{CONTIG_DATA}. Fields:

CONTIG_ADDR unsigned int

address of the TLB first read, with which this entry is expected to be contiguous.

CONTIG_DATA unsigned int

expected contents based on the entry at CONTIG_ADDR.

ENTRY_ADDR unsigned int

address of conflicting TLB entry.

ENTRY_DATA unsigned int

contents of conflicting TLB entry.

ArchMsg.Warning.warning_contiguous_bit_error_D128

TAGS **UNPREDICTABLE** DISPLAY TLB entry at %{ENTRY_ADDR} was not contiguous with entry at %{CONTIG_ADDR} Contents are %{ENTRY_DATA_HI} %{ENTRY_DATA_LO} but expected %{CONTIG_DATA}. Fields:

CONTIG_ADDR unsigned int

address of the TLB first read, with which this entry is expected to be contiguous.

CONTIG_DATA unsigned int

expected contents based on the entry at CONTIG_ADDR.

ENTRY_ADDR unsigned int

address of conflicting TLB entry.

ENTRY_DATA_HI unsigned int

Higher bit contents of conflicting TLB entry.

ENTRY_DATA_LO unsigned int

Lower bit contents of conflicting TLB entry.

ArchMsg.Warning.warning_cp10_cp11_mismatch

TAGS **UNPREDICTABLE** DISPLAY Unpredictable configuration of CP10 and CP11 controls at EL %{EL:d}. Fields:

EL unsigned int

EL owning the control.

ArchMsg.Warning.warning_cp10_cp11_reserved_value

TAGS **UNPREDICTABLE** DISPLAY Unpredictable configuration of CPACR CP10 (%{CP10}) or CP11 (%{CP11}). Fields:

CP10 unsigned int

CP10 access.

CP11 unsigned int

CP11 access.

ArchMsg.Warning.warning_csselr_level_out_of_range

TAGS **UNPREDICTABLE** DISPLAY CSSELR.Level out of range, written value %{LEVEL} with only %{IMPLEMENTED:d} levels implemented. Fields:

IMPLEMENTED unsigned int

number of cache levels implemented.

LEVEL unsigned int

written value of CSSELR.Level.

TYPE enum

The type of cache selected by CSSELR.InD and TnD.

ArchMsg.Warning.warning_debug_flow_control_bits_not_obeyed

TAGS **UNPREDICTABLE** DISPLAY Flow control bits for %{REGISTER} were not obeyed, causing %{ERROR}. Fields:

ERROR enum

type of error resulting.

REGISTER enum

register under consideration.

ArchMsg.Warning.warning_debug_register_access_during_reset

TAGS **UNPREDICTABLE** DISPLAY Unpredictable %{IS_WRITE:(read|write)} access to debug register offset %{OFFSET} during reset. Fields:

IS_WRITE unsigned int

Write Not Read.

OFFSET unsigned int

Register Offset.

ArchMsg.Warning.warning_decode_cps_inconsistent_fields

TAGS **UNPREDICTABLE** DISPLAY The mode setting fields in the CPS instruction are inconsistent.

ArchMsg.Warning.warning_decode_invalid_state

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable in current exception-level/security state.

ArchMsg.Warning.warning_default_cacheable_mmu_on

TAGS **UNPREDICTABLE** DISPLAY Default cacheable is enabled (HCR.DC = 1) while MMU is enabled.

ArchMsg.Warning.warning_default_cacheable_vmmu_off

TAGS **UNPREDICTABLE** DISPLAY Default cacheable is enabled (HCR.DC = 1) while HCR.VM = 0.

ArchMsg.Warning.warning_deprecated_wvr_bit_2_set

TAGS **UNPREDICTABLE** DISPLAY DBGWVR%{N:d}_EL1=%{WVR:h} has bit 2 set, deprecated in ARMv8. Fields:

N unsigned int

index.

WVR unsigned int

DBGWVR.

ArchMsg.Warning.warning_eret_while_software_step_active_pending

Missing ISB between setting MDSCR_EL1.SS and ERET TAGS **UNPREDICTABLE**.

ArchMsg.Warning.warning_exclusive_to_non_normal

TAGS **UNPREDICTABLE** DISPLAY Exclusive Access to Non-Normal Memory %{ADDR}. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_exclusive_to_non_writeback

TAGS **UNPREDICTABLE** DISPLAY Exclusive Access to Non-Writeback-Cacheable Memory
%{ADDR}. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_execute_from_device_memory

TAGS **UNPREDICTABLE** DISPLAY An attempt was made to execute from Device memory.

ArchMsg.Warning.warning_illegal_cpsr_mode

TAGS **UNPREDICTABLE** DISPLAY Writing an illegal or unimplement mode (%{NEW_MODE:h})
to CPSR was unpredictable. Fields:

NEW_MODE unsigned int

New value of CPSR.M.

ArchMsg.Warning.warning_illegal_srs_mode

TAGS **UNPREDICTABLE** DISPLAY Illegal or **UNPREDICTABLE** mode (%{MODE:h}) used for SRS
instruction. Fields:

MODE unsigned int

mode for Banked SP.

ArchMsg.Warning.warning_implementation_defined_read_debug_register_in_SCS

Read Access to Debug registers from the processor is IMPLEMENTATION DEFINED TAGS
IMP_DEF. Fields:

PPB_OFFSET unsigned int

debug register offset.

ArchMsg.Warning.warning_implementation_defined_sequential_security_transitions_supported

The behavior of sequential instruction fetches that cross from non-secure to secure memory
and contain an SG instruction is CONSTRAINED **UNPREDICTABLE** TAGS **UNPREDICTABLE**.

ArchMsg.Warning.warning_implementation_defined_stack_limit_check_supported

It is **IMPLEMENTATION DEFINED** whether stack pointer limit checking is performed for this
instruction TAGS IMP_DEF. Fields:

ADDRESS unsigned int

Address of instruction.

OPCODE unsigned int

instruction opcode.

ArchMsg.Warning.warning_implementation_defined_taggednonsharedmemorytype

TAGS IMP_DEF DISPLAY Memory attribute has Tagged, non shared memory type; it is
IMPLEMENTATION DEFINED whether the memory location is treated as Tagged or non-Tagged.

ArchMsg.Warning.warning_implementation_defined_write_debug_register_in_SCS

Write Access to Debug registers from the processor is IMPLEMENTATION DEFINED TAGS IMP_DEF. Fields:

DATA unsigned int

data attempted to be written.

PPB_OFFSET unsigned int

debug register offset.

ArchMsg.Warning.warning_invalid_tcr_granule

TAGS UNPREDICTABLE DISPLAY TCR.TG%{TG_ID:(0|1)} bits have been set to a granule size %{{REQUEST:d}}K not implemented - using %{{SUBSTITUTE:d}}K. Fields:

REQUEST signed int

page size requested (or 0 for reserved).

SUBSTITUTE signed int

best guess available page size.

TG_ID bool

bits TG0 or TG1.

ArchMsg.Warning.warning_load_multiple_user_registers_from_user_mode

TAGS UNPREDICTABLE DISPLAY An LDM(user registers) instruction executed from user mode.

ArchMsg.Warning.warning_mte_in_mem_attr_not_supported

TAGS UNPREDICTABLE DISPLAY Memory attribute %{{ATTR}} has Tagged memory type but Armv8.5-MTE is not implemented. Fields:

ATTR unsigned int

Memory attribute.

ArchMsg.Warning.warning_reserved_breakpoint_state_match

TAGS UNPREDICTABLE DISPLAY %{{IS_BREAKPOINT:(Watchpoint|Breakpoint)}} programmed with a reserved combination of HMC, SSC and %{{IS_BREAKPOINT:(PAC|PMC)}}. Fields:

IS_BREAKPOINT unsigned int

Is a breakpoint, not a watchpoint.

ArchMsg.Warning.warning_shareability

TAGS UNPREDICTABLE DISPLAY Unpredictable: combination of the 1st and 2nd stages of translation is Normal Inner Non-Cacheable, Outer Non-Cacheable.

ArchMsg.Warning.warning_software_step_set_while_enabled

TAGS UNPREDICTABLE DISPLAY MDSCR_EL1.SS set to 1 while software step debug exceptions are enabled.

ArchMsg.Warning.warning_thumb_instruction_wraps_around_memory

TAGS UNPREDICTABLE DISPLAY Thumb instruction wraps around memory.

ArchMsg.Warning.warning_ttbr_sbz_bits_are_not_zero

TAGS UNPREDICTABLE DISPLAY `%{TTBR}[%{MSB:d}:0]` should be zero, but are not. Fields:

MSB unsigned int

x-1.

TTBR enum

which TTBR register.

ArchMsg.Warning.warning_unaligned_address_dbgdttrrx_write

TAGS UNPREDICTABLE DISPLAY Unaligned address `%{ADDR}` in X0/R0 when DBGDTTRRX written in memory access mode. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unaligned_address_dbgdttrtx_read

TAGS UNPREDICTABLE DISPLAY Unaligned address `%{ADDR}` in X0/R0 when DBGDTTRTX read in memory access mode. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unaligned_to_device

TAGS UNPREDICTABLE DISPLAY Unaligned Access to Device Memory `%{ADDR}`. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unaligned_to_strongly_ordered

TAGS UNPREDICTABLE DISPLAY Unaligned Access to Strongly Ordered Memory `%{ADDR}`. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unknown_sau_rnr

TAGS UNPREDICTABLE DISPLAY SAU_RNR was set to an unsupported value.

ArchMsg.Warning.warning_unpredictable_AIRCR_PRIS_and_BFHFNMINs

TAGS UNPREDICTABLE DISPLAY The effect of setting both AIRCR.BFHFNMINs and AIRCR.PRIS to 1 is UNPREDICTABLE.

ArchMsg.Warning.warning_unpredictable_AIRCR_VECTCLRACTIVE_when_not_in_debug

TAGS UNPREDICTABLE DISPLAY The effect of writing a 1 to AIRCR.VECTCLRACTIVE if the processor is not halted in Debug state is UNPREDICTABLE.

ArchMsg.Warning.warning_unpredictable_AIRCR_VECTRESET_and_SYSRESETREQ

TAGS UNPREDICTABLE DISPLAY When the processor is halted in Debug state, if a write to the register writes a 1 to both VECTRESET and SYSRESETREQ, the behavior is UNPREDICTABLE.

ArchMsg.Warning.warning_unpredictable_AIRCR_VECTRESET_when_not_in_debug

TAGS **UNPREDICTABLE** DISPLAY The effect of writing a 1 to AIRCR.VECTRESET if the processor is not halted in Debug state is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_AIRCR_incorrect_VKEY

TAGS **UNPREDICTABLE** DISPLAY The value 0x05FA must be written to AIRCR.VKEY, otherwise the register write is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_EXC_RETURN_Reserved_Bit

EXC_RETURN[23:7] are reserved with the special condition that all bits should be written as one. Values other than all 1s are **UNPREDICTABLE**. TAGS **UNPREDICTABLE**. Fields:

EXC_RETURN unsigned int

Special PC value.

SBO_MASK unsigned int

Which bits Should Be One to not UNPRED.

ArchMsg.Warning.warning_unpredictable_PMEVTYPEPn_ELO_VS_reserved_value

TAGS **UNPREDICTABLE** DISPLAY Unpredictable reserved value of PMEVTYPEn<n>_ELO.VS.

ArchMsg.Warning.warning_unpredictable_change_to_DEMCR_MON_STEP_at_insufficient_priority

TAGS **UNPREDICTABLE** DISPLAY The effect of changing DEMCR.MON_STEP at an execution priority that is lower than the priority of the DebugMonitor exception is **CONSTRAINED UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_change_to_priority_of_active_exception

TAGS **UNPREDICTABLE** DISPLAY Changing the priority of an active exception is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_clear_lspact

TAGS **UNPREDICTABLE** DISPLAY **UNPREDICTABLE** state on context switch. Fields:

DRVEXC_TAKEN unsigned int

Derived exception taken, not pended.

REACHED_LAST_STORE unsigned int

Only the last faulted.

ArchMsg.Warning.warning_unpredictable_dbg_exit_unaligned_dlr

TAGS **UNPREDICTABLE** DISPLAY Debug exit to AArch32 with DLR[0] set to 1. PC[0] can be set to 0 or to DLR[0].

ArchMsg.Warning.warning_unpredictable_exception_catch

TAGS **UNPREDICTABLE** DISPLAY Generation of an exception catch event is unpredictable.

ArchMsg.Warning.warning_unpredictable_exception_return_inconsistent_state

TAGS **UNPREDICTABLE** DISPLAY state on exception return is unpredictable.

ArchMsg.Warning.warning_unpredictable_exception_return_instruction

TAGS **UNPREDICTABLE** DISPLAY exception return instruction is unpredictable.

ArchMsg.Warning.warning_unpredictable_in_debug_state

TAGS **UNPREDICTABLE** DISPLAY Instruction is **UNPREDICTABLE** when executed in debug state.

ArchMsg.Warning.warning_unpredictable_mair_encoding

TAGS **UNPREDICTABLE** DISPLAY Memory attribute %{ATTR} has invalid MAIR encoding; this may lead to unpredictable results. Fields:

ATTR unsigned int

Memory attribute.

ArchMsg.Warning.warning_unpredictable_pmu_counter_access

TAGS **UNPREDICTABLE** DISPLAY Access to PMU counters from non-secure EL0 or EL1 is **UNPREDICTABLE** with MDCR_EL2.HPMN set to 0.

ArchMsg.Warning.warning_unpredictable_prioritization_breakpoint_match_vector_catch

TAGS **UNPREDICTABLE** DISPLAY Two events with the same priority occurred at the same instruction: Address Matching Vector Catch debug event, and Breakpoint debug event. It is constrained unpredictable which is taken.

ArchMsg.Warning.warning_unpredictable_stack_selection

TAGS **UNPREDICTABLE** DISPLAY **UNPREDICTABLE** state on context switch.

ArchMsg.Warning.warning_unpredictable_tsize_out_of_range

TAGS **UNPREDICTABLE** DISPLAY %{TCR}.T%{TTBR:d}SZ (%{TSIZE:d} bits) is out of range. Must be between 25 and %{TSIZE_MAX:d}. Fields:

TCR enum

Which TCR.

TSIZE signed int

Value of TSize.

TSIZE_MAX signed int

The maximum allowed value.

TTBR unsigned int

Which TTBR region.

ArchMsg.Warning.warning_unpredictable_unaligned_pc_as_base_register

TAGS **UNPREDICTABLE** DISPLAY The PC value must be word-aligned, otherwise the behavior of the instruction is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_unaligned_pop_stack

TAGS **UNPREDICTABLE** DISPLAY **CONSTRAINED UNPREDICTABLE** unaligned stack frame %{FRAME_ADDR:x} in exception-return. Fields:

FRAME_ADDR unsigned int

unaligned stack frame address.

ArchMsg.Warning.warning_unpredictable_unaligned_to_non_writeback

TAGS **UNPREDICTABLE** DISPLAY Unaligned Access to Non-Writeback-Cacheable Memory.

ArchMsg.Warning.warning_unpredictable_vmsa_memattrib

TAGS **UNPREDICTABLE** DISPLAY Unpredictable vmsa memory attribute with texcb value programmed to %{TEXCB} when tex remap is %{TEX_REMAP}. Fields:

TEXCB unsigned int

texcb value.

TEX_REMAP bool

use tex remap.

ArchMsg.Warning.warning_unpredictable_vtcr_t0sz_sl0

TAGS UNPREDICTABLE DISPLAY The combination of VTCR.TOSZ=%{TOSZ:d} and VTCR.SLO=%{SLO:d} is unpredictable. Fields:

SLO signed int

VTCR.SLO, starting level for stage 2 translation table walks.

TOSZ signed int

VTCR.TOSZ, required input address range.

ArchMsg.Warning.warning_unpredictable_write_DHCSR

TAGS UNPREDICTABLE DISPLAY UNPREDICTABLE set of bit-changes in DHCSR. Fields:

NEW unsigned int

data attempted to be written.

OLD unsigned int

previous value.

ArchMsg.Warning.warning_unpredictable_write_SHCSR

TAGS UNPREDICTABLE DISPLAY UNPREDICTABLE setting a pending bit to 1 for an exception with priority >= execution priority.

ArchMsg.Warning.warning_unsupported_access_to_memory_mapped_register

TAGS UNPREDICTABLE DISPLAY Access to memory mapped register is unsupported.

ArchMsg.Warning.warning_user_jmcr_access

TAGS UNPREDICTABLE DISPLAY User %{WRITE:(write|read)} access to JMCR is unpredictable. Fields:

WRITE unsigned int

access is write.

ArchMsg.Warning.warning_wcr_mask_and_bas

TAGS UNPREDICTABLE DISPLAY DBGWCR_EL1.MASK is non-zero and BAS is not 0xff.

ArchMsg.Warning.warning_wcr_mask_reserved

TAGS UNPREDICTABLE DISPLAY DBGWCR_ELn.MASK is set to a reserved value.

ArchMsg.Warning.warning_wcr_non_configuous_bas

TAGS UNPREDICTABLE DISPLAY DBGWCR_EL1.BAS has non-contiguous set of ones.

ArchMsg.Warning.warning_wvr_masked_bit_not_zero

TAGS UNPREDICTABLE DISPLAY DBGWVRn_EL1.VA contains a bit masked by DBGWCRn_EL1.MASK that is not zero.

ArchMsg.Why.why_branch_target_exception

A branch target exception has occurred. DISPLAY branch target exception: target pc={PC}, target opcode={OPCODE}. PRIMARY KEY PC. Fields:

BTYPE unsigned int

pstate.BTYPE at time branch taken.

OPCODE unsigned int

opcode at branch target.

PC unsigned int

Virtual Address of branch target.

ArchMsg.Why.why_illegal_state

An illegal mode change has occurred. DISPLAY illegal mode change: %{MESSAGE}. Fields:

MESSAGE string

The message.

ArchMsg.warning_invalid_interval_counter_reload_value

Invalid interval counter reload value in PMSIRR_EL1.INTERVAL: software should set this to a value greater than or equal to the minimum indicated by PMSIDR_EL1.Interval.

ArchMsg.warning_invalid_spe_buffer_write_pointer

Invalid value programmed in PMBPTR_EL1.PTR or PMBLIMITR_EL1.LIMIT: Current write pointer must be at least one sample record below the write limit pointer and PMBPTR_EL1.PTR[63:56] must be equal to PMBLIMITR_EL1.LIMIT[63:56].

BRANCH_MISPREDICT

Simulating branch mispredict. Fields:

PC unsigned int

Origin address (or 0 if unavailable).

BRA_DIR

Direct branches, to immediate address. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISET enum

The instruction set of the branch instruction.

IS_COND bool

Indicates if this is a conditional branch.

IS_HINTED bool

Whether the conditional instruction is hinted.

IS_ISB bool

Whether the instruction is ISB.

IS_LINK bool

Indicates if this is a branch with link.

PC unsigned int

The address of the branch instruction.

TARGET_ISET enum

The instruction set after the branch.

TARGET_PC unsigned int

The address the instruction branches to.

BRA_INDIR

Indirect branches, perhaps to a register. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISET enum

The instruction set of the branch instruction.

IS_COND bool

Indicates if this is a conditional branch.

IS_HINTED bool

Whether the conditional instruction is hinted.

IS_ISB bool

Whether the instruction is ISB.

IS_LINK bool

Indicates if this is a branch with link.

PC unsigned int

The address of the branch instruction.

TARGET_ISET enum

The instruction set after the branch.

TARGET_PC unsigned int

The address the instruction branches to.

BRA_INDIR_CCFAIL

Indirect branches fail, perhaps to a register.

BRA_RET_CCFAIL

Branch return fail, perhaps to a register.

CACHE_MAINTENANCE_OP

Cache Maintenance Operation. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

DATA unsigned int

Specified MVA or Set/way.

FUNCTION enum

Clean / Invalidate.

SCOPE enum

Affected region.

SIDE enum

Inst / Data.

CCFAIL

Conditional instruction condition check fail. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CCFAIL_UNC

Unconditional instruction condition check fail. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CCPASS

Conditional instruction condition check pass. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CCPASS_UNC

Unconditional instruction condition check pass. Fields:

COND **enum**

The condition of the conditional instruction.

CORE_NUM **unsigned int**

Core number in a multi processor.

PC **unsigned int**

The address of the conditional instruction.

CODE_CACHE_MAINT

Code cache maintenance. Fields:

END_INCL **unsigned int**

Inclusive end address for VA or PA requests.

NS **bool**

Non-secure world for PA.

START **unsigned int**

Start address for VA or PA requests.

TYPE **enum**

Request type.

COMPILE_BLOCK_END

Last instruction of basic block translated. Fields:

VADDR **unsigned int**

Address of next instruction after this basic block.

COMPILE_BLOCK_START

First instruction of basic block translated. Fields:

VADDR **unsigned int**

Address of first instruction in basic block.

COMPILE_INST

ARM instruction compiled. Fields:

DISASS **string**

Disassembly of the instruction.

ISSET **enum**

The instruction set of this instruction.

ITSTATE **unsigned int**

The ITSTATE current for the instruction.

OPCODE **unsigned int**

The opcode of the instruction.

PC unsigned int

The address of the instruction.

SIZE unsigned int

The size of the instruction in bytes.

CONTEXTIDR

Context ID Register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

NS bool

Secure or nonsecure banked register is accessed.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The new value written.

CONTEXT_SYNC

Called for every context synchronization event. Fields:

ARCH_CSE bool

Architectural Context Synchronization Event.

PC unsigned int

The address of the synchronization instruction.

SYSREG_SYNC bool

System register synchronization event.

CORE_ENDIAN

Core BE8 Big-Endian state changed. Fields:

BE8 bool

Core BE8 Big-Endian state.

CORE_INFO

Static processor attributes. Only triggered by a call to DumpState(). Fields:

ARCH_PROFILE enum

The architecture profile of the core.

CLUSTER_ID unsigned int

The cluster ID of this processor.

CORE_NUM unsigned int

The number of this core in an MP processor.

DETAILED_CORE_FEATURES enum

Detailed core features as a bitfield. Enum values are bitmasks to extract the info from the field.

FPU_VERSION enum

The VFP version implemented by the core.

MEM_ARCH enum

The memory architecture of the core.

MIDR_EL1 unsigned int

MIDR_EL1 value.

NUM_CORES unsigned int

The number of cores in this MP processor.

QUANTUM_SIZE unsigned int

The default quantum size of the core (in ticks).

SECURITY_FEATURES bool

Does the core have security features?.

CORE_LOADS

Processor load accesses. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ACQREL enum

Is this an acquire/release.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DATA unsigned int

The data read or written.

ELEMENT_SIZE unsigned int

Width of each element.

ENDIANNESS enum

Byte ordering of this access.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

LOCK enum

Normal, exclusive or locked access.

MECID unsigned int

Memory Encryption Context ID, if applicable.

MECID2 unsigned int

Memory Encryption Context ID of the second page, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PAGE_SIZE unsigned int

Page size as $\log_2(\text{size in bytes})$.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK, 2=Exclusive Failed, 3=Unsupported.

SIZE unsigned int

Width of the access in bytes. Only required if DATA is not traced.

TRANS bool

Is this a translated access.

VADDR unsigned int

The virtual address of the access.

CORE_REGS

Changes of the core registers R0 to R14. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number, 0 to 14.

MODE enum

Bank of the register accessed.

OLD_VALUE unsigned int

The old value overwritten.

PHYS_ID enum

The physical register accessed.

VALUE unsigned int

The new value written to the register.

CORE_REGS64

Changes of the core registers X0..X30, SP_ELn. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ID enum

The register number, 0-30 for X0-X30, >=32 for SP_ELn.

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

CORE_REGS64_READ

General purpose AArch64 register has been read X0..X30, SP_ELn. Fields:

ID enum

The register number, 0-30 for X0-X30, >=32 for SP_ELn.

VALUE unsigned int

The value read from the register.

CORE_STALLED_TRANSITION

Detect when a core is prevented from progressing PChannel states. Fields:

CURRENT_STATE enum

Current power mode of core.

REASON string

The reason a transition was stalled, and what must be done to unblock it.

TARGET_STATE enum

Desired power mode of core after transition.

CORE_STORES

Processor store accesses. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ACQREL enum

Is this an acquire/release.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DATA unsigned int

The data read or written.

ELEMENT_SIZE unsigned int

Width of each element.

ENDIANNESS enum

Byte ordering of this access.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

LOCK enum

Normal, exclusive or locked access.

MECID unsigned int

Memory Encryption Context ID, if applicable.

MECID2 unsigned int

Memory Encryption Context ID of the second page, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PAGE_SIZE unsigned int

Page size as $\log_2(\text{size in bytes})$.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK, 2=Exclusive Failed, 3=Unsupported.

SIZE unsigned int

Width of the access in bytes. Only required if DATA is not traced.

TRANS bool

Is this a translated access.

VADDR unsigned int

The virtual address of the access.

CP14_READ

System Coprocessor register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP14 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc1 unsigned int

opcode 1.

opc2 unsigned int

opcode 2.

CP14_WRITE

System Coprocessor register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP14 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc1 unsigned int

opcode 1.

opc2 unsigned int

opcode 2.

CP15_READ

System Coprocessor register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP15 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc1 unsigned int

opcode 1.

opc2 unsigned int

opcode 2.

CP15_READ64

System Coprocessor register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP15 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc unsigned int
opcode 1.

CP15_WRITE

System Control Coprocessor register write. Fields:

CORE_NUM unsigned int
Core number in a multi processor.

CRm unsigned int
CRm.

CRn unsigned int
CRn.

NS bool
Secure or nonsecure banked register is accessed.

REG_NAME string
Name of the CP15 register accessed.

SECURITY_STATE enum
The current security state.

UNDEF bool
The register accessed is undefined.

UPDATED_VALUE unsigned int
Updated value of the register now it has been written.

VALUE unsigned int
The new value written.

opc1 unsigned int
opcode 1.

opc2 unsigned int
opcode 2.

CP15_WRITE64

System Coprocessor register write. Fields:

CORE_NUM unsigned int
Core number in a multi processor.

CRm unsigned int
CRm.

NS bool
Secure or nonsecure banked register is accessed.

REG_NAME string
Name of the CP15 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc unsigned int

opcode 1.

CPSR

CPSR change. (In AArch64 PSTATE represented in SPSR format). Fields:

CORE_NUM unsigned int

Core number in a multi processor.

OLD_VALUE unsigned int

The old CPSR value.

UNKNOWN unsigned int

Bits within the register that have unknown value.

VALUE unsigned int

The new CPSR value.

CRYPTO_SPEC

Every crypto instruction speculatively executed.

CorePowerStateChange

Core power-up state has changed. DISPLAY Core powered %{PU:(up|down)}. Fields:

PU bool

Whether the core powered up.

DATA_CACHE_SET_ALLOC_TAGS

Set allocation tags from DC GVA instruction. Fields:

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

NSDESC unsigned int

The security state of the access.

PADDR unsigned int

The physical address of the access.

PAS enum

The physical address space of the page.

RESPONSE enum

0=Aborted, 1=OK.

SIZE unsigned int

The size of memory in bytes.

TAG unsigned int

The value of the tag being set.

TAGGING_TYPE enum

Type of Memory tagging (Physical or Virtual).

TAG_VADDR unsigned int

The virtual address of the Virtual Tag if Virtual tagging is used; otherwise 0.

VADDR unsigned int

Virtual address used for setting allocation tags.

DATA_CACHE_ZERO

Zero and invalidate cache line from DC ZVA instruction. Fields:

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

NSDESC unsigned int

The security state of the access.

PADDR unsigned int

The physical address of the access.

PADDR_TAG unsigned int

If tag checked, the physical address tag for the access.

PAS enum

The physical address space of the page.

RESPONSE enum

0=Aborted, 1=OK.

SIZE unsigned int

The size of zero'd memory in bytes.

TAG_CHECKED bool

Is this access tag checked.

VADDR unsigned int

Virtual address of the zero'd cache line.

DEBUG_EVENT

Hardware debug support event. Fields:

EVENT enum

Description of event.

VALUE unsigned int

data value.

DEBUG_TARGET_ISET

Hardware debug target ISA. Fields:

TARGET_ISET enum

The target instruction set of the debug state.

DMI_ALLOCATE

DMI allocated. Fields:

ACCESS_DIRECTION enum

READ / WRITE.

PADDR unsigned int

The physical address of the DMI.

PTR unsigned int

DMI range pointer.

SIDE enum

DSIDE / ISIDE.

SIZE unsigned int

The size of DMI.

VADDR unsigned int

The virtual address of the DMI..

DMI_HIT

DMI hit, access used DMI (only fire for small DMI regions i.e. less than a full page). Fields:

ACCESS_DIRECTION enum

READ / WRITE.

PADDR unsigned int

The physical address of the DMI.

SIDE enum

DSIDE / ISIDE.

SIZE unsigned int

The size of the DMI.

VADDR unsigned int

The virtual address of the DMI..

DMI_REVOKE

DMI revoked. Fields:

ACCESS_DIRECTION enum

READ / WRITE.

PADDR unsigned int

The physical address of the DMI.

SIDE enum

DSIDE / ISIDE.

SIZE unsigned int

The size of DMI.

VADDR unsigned int

The virtual address of the DMI..

END_COMPILE

Compilation end. Fields:

END_OF_PAGE_COUNT unsigned int

Number of basic blocks exited due to page boundary since START_COMPILE.

FETCHFAIL_COUNT unsigned int

Number of basic blocks exited due to ifetch failure START_COMPILE.

INST_COUNT unsigned int

Number of instructions compiled since START_COMPILE.

NONSEQ_COUNT unsigned int

Number of basic blocks exited due to non-sequential instructions since START_COMPILE.

PAGE_STRADDLE_COUNT unsigned int

Number of basic blocks exited due to unaligned instructions crossing page since START_COMPILE.

ESB

Error Synchronization Barrier hit. Fields:

ASYNC bool

ESB is async.

IMPLICIT bool

ESB is implicit.

EXCEPTION

Exception taken. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ESR unsigned int

The value assigned to ESR when exception is taken.

ESR_EC unsigned int

The Value assigned to ESR.EC when exception is taken.

IS_PHYSICAL bool

Physical or Virtual exception.

IS_TRAP bool

Exception coming from a trap.

LR unsigned int

The value assigned to the link register.

PC unsigned int

The location where the exception occurred.

PREFERRED_RETURN unsigned int

The preferred return address for the exception.

TAKEN_LOCALLY bool

Whether exception is taken locally(same exception level).

TARGET_ISET enum

The instruction set of the exception handler code.

TARGET_PC unsigned int

The address the exception branches to.

VECTOR enum

The exception vector.

VECTOR_OFFSET unsigned int

The offset of an exception vector from the base address.

EXCEPTION_END

Every exception completed.

EXCEPTION_INFO

Trace human-readable reason of the current exception for data/instruction aborts, floating point exceptions and unknown exceptions. Fields:

DESCRIPTION string

Reason for current exception.

ESR_EC unsigned int

The value of ESR.EC when exception is taken.

EXCEPTION_RAISE

Every exception raised.

EXCEPTION_RETURN

Branches on leaving exception. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISET enum

The instruction set of the branch instruction.

IS_COND bool

Indicates if this is a conditional branch.

IS_HINTED bool

Whether the conditional instruction is hinted.

IS_ISB bool

Whether the instruction is ISB.

IS_LINK bool

Indicates if this is a branch with link.

PC unsigned int

The address of the branch instruction.

TARGET_ISET enum

The instruction set after the branch.

TARGET_PC unsigned int

The address the instruction branches to.

EXCEPTION_RETURN_PREBRANCH

Exception return event that must occur before PC and processor state have updated.

EXCEPTION_START

Every exception started.

FIQ_TAKEN

FIQ taken exception.

FP_EXCEPTIONS_SIGNALLED

Trace which FP exceptions were signalled as a part of the FP operations performed by an instruction. Fields:

FP_EXCEPTIONS_SIGNALLED_MASK unsigned int

Indicates which FP exceptions were signalled by an instruction. This uses the same encoding as the cumulative floating-point exception bitfields in FPSR.

FP_STATE

Floating point state. Fields:

FPSR **unsigned int**

FPSR register value.

MASK **unsigned int**

The mask indicating updated cumulative bits.

FREQ_CHANGED

Core frequency changed. Fields:

CORE_NUM **unsigned int**

Core number in a multi processor.

INST_COUNT **unsigned int**

The instruction count of this core.

NEW_FREQ **signed int**

The new frequency of this core (expressed in Hz).

OLD_FREQ **signed int**

The old frequency of this core (expressed in Hz).

HLT

HLT instruction occurred. Fields:

IMM16 **unsigned int**

The 16-bit immediate value encoded in HLT instruction.

INST

Every instruction executed. Fields:

CORE_NUM **unsigned int**

Core number in a multi processor.

CURRENT_TIME **unsigned int**

The core's current time, as simulated time plus local time.

DEBUG_STATE **bool**

The instruction is executed in debug state.

DISASS **string**

Disassembly of instruction.

INST_COUNT **unsigned int**

The core's instruction counter, starting at 1 for the first instruction.

ISSET **enum**

The current instruction set.

ITSTATE **unsigned int**

The current ITSTATE.

LOCAL_TIME **unsigned int**

The core's local time, relative to the current quantum.

MODE	enum
	The mode the core is in.
NS	unsigned int
	The core's non-secure bit.
NSDESC	unsigned int
	The physical address non-secure bit.
NSDESC2	unsigned int
	The second page physical address non-secure bit.
OPCODE	unsigned int
	The opcode of the instruction.
PADDR	unsigned int
	The physical address of the instruction.
PADDR2	unsigned int
	If different from PADDR, the physical address of the second page of the instruction.
PAS	enum
	The physical address space of the page.
PAS2	enum
	The physical address space of the page.
PC	unsigned int
	The address of the instruction.
SECURITY_STATE	enum
	The core's security state.
SIZE	unsigned int
	The size of the instruction in bytes.
INST_END	
	Every instruction completed.
INST_START	
	Every instruction started. Fields:
DEBUG_STATE	bool
	The instruction is executed in debug state.
ISSET	enum
	The current instruction set.
MODE	enum
	The mode the core is in.
NS	enum
	The current Secure State.

PC unsigned int

The address of the conditional instruction.

SECURITY_STATE enum

The current Security State.

INST_STRADDLE

Instruction straddles boundary. Fields:

DEBUG_STATE bool

The instruction is executed in debug state.

ISET enum

The current instruction set.

MODE enum

The mode the core is in.

NS enum

The current Secure State.

PC unsigned int

The address of the conditional instruction.

SECURITY_STATE enum

The current Security State.

IRQ_TAKEN

IRQ taken exception.

ISB

ISB instruction occurred. Fields:

PC unsigned int

The address of the ISB instruction.

LOAD_MULTIPLE_REGS_LIST

List of destination registers (not source) of a multiple registers load instruction. Fields:

REG string

Register to add into the list.

LOCAL_MONITOR

Local monitor activity. Fields:

PADDR unsigned int

Local Monitor Address.

State enum

State of the monitor (Open/Exclusive).

MAPPING_CREATE

A memory mapping has been created. This is an implementation detail of the model and so may change without notice. Fields:

ASID **unsigned int**

The ASID used during translation, or 0 if no ASID was used.

IS_VALID **bool**

Whether the mapping is valid or not.

PADDR **unsigned int**

Base physical address of region, or 0 if not known.

SIDE **enum**

I-side or D-side.

SIZE **unsigned int**

Size of the page, or 0 if not known.

VADDR **unsigned int**

Base virtual address of region.

VMID **unsigned int**

The VMID used during translation, or 0 if no VMID was used.

MAPPING_DESTROY

A memory mapping has been destroyed. This is an implementation detail of the model and so may change without notice. Fields:

ASID **unsigned int**

The ASID used during translation.

CAUSE **enum**

The cause of invalidation.

PADDR **unsigned int**

Base physical address of page.

SIDE **enum**

I-side or D-side.

SIZE **unsigned int**

Size of the page.

VADDR **unsigned int**

Base virtual address of page.

VMID **unsigned int**

The VMID used during translation.

MEMMAP_DEBUG_READ

Memory mapped reads to the debug registers. Fields:

ADDR unsigned int

address.

CORE_NUM unsigned int

Core number in a multi processor.

EXT bool

Whether access is from an external device (such as the DAP).

NS enum

Secure state of the access.

REG_NAME string

Name of the debug register accessed.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

MEMMAP_DEBUG_WRITE

Memory mapped writes to the debug registers. Fields:

ADDR unsigned int

address.

CORE_NUM unsigned int

Core number in a multi processor.

EXT bool

Whether access is from an external device (such as the DAP).

NS enum

Secure state of the access.

REG_NAME string

Name of the debug register accessed.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The new value written.

MMU_D128_TTB_READ

This event is triggered by reads caused by a D128 translation table walk. Fields:

ABORTED bool

Was the walk successful.

DATA_HIGH unsigned int

The data upper bits read.

DATA_LOW unsigned int

The data lower bits read.

IPA unsigned int

For stage 1, the IPA of the read.

LEVEL unsigned int

Translation table level.

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

OUTADDR unsigned int

The output physical address.

PADDR unsigned int

The physical address of the read.

PASmem enum

Physical Address Space of the accessed memory.

PASreq enum

Physical Address Space of the requested lookup.

REGIME_EL enum

Entry matches in this translation regime.

REQUESTER enum

The requester of this table walk.

SIDE enum

Inst / Data.

STAGE unsigned int

Translation stage.

MMU_D128_TTB_WRITE

This event is triggered by writes caused by a D128 translation table walk. Fields:

ABORTED bool

Was the walk successful.

DATA_HIGH unsigned int

The data upper bits written.

DATA_LOW unsigned int

The data lower bits written.

IPA unsigned int

For stage 1, the IPA of the write.

IS_HACDBS_ACCESS bool

If the access was due to HACDBS_CLEAN.

LEVEL unsigned int

Translation table level.

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

OUTADDR unsigned int

The output physical address.

PADDR unsigned int

The physical address of the write.

PASmem enum

Physical Address Space of the accessed memory.

PASreq enum

Physical Address Space of the requested lookup.

REGIME_EL enum

Entry matches in this translation regime.

SIDE enum

Inst / Data.

STAGE unsigned int

Translation stage.

MMU_TRANS

Address translation information. Fields:

ASID unsigned int

Address space identifier.

CORE_NUM unsigned int

Core number in a multi processor.

Hyp bool

Entry matches in Hyp state only.

INNERCACHE_RA bool

Is the inner cache allocate on read.

INNERCACHE_TRANSIENT bool

Is the inner write-through transient.

INNERCACHE_TYPE enum

Inner Caching scheme (NC/MB/WA).

INNERCACHE_WA bool

Is the inner cache allocate on write.

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MEMTYPE enum

Memory type.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC enum

Is secure side supposed to access secure or nonsecure memory.

OUTERCACHE_RA bool

Is the outer cache allocate on read.

OUTERCACHE_TRANSIENT bool

Is the outer write-through transient.

OUTERCACHE_TYPE enum

Outer Caching scheme (NC/MB/WA).

OUTERCACHE_WA bool

Is the outer cache allocate on write.

PADDR unsigned int

Physical address of the access.

PAGESIZE unsigned int

Page size as $\log_2(\text{size})$.

PAS enum

PAS supposed to be accessed is secure or nonsecure memory or Root/Realm if RME is implemented.

SH enum

Shareability.

SIDE enum

Inst / Data.

STAGE1_PERM unsigned int

Stage 1 permissions, [5:3] Privileged access (XWR) [2:0] User access (XWR).

STAGE2_PERM unsigned int

Stage 2 permission mask [5:3] Privileged access (XWR) [2:0] User access (XWR).

VADDR unsigned int

Virtual address of the access.

VMID unsigned int

Virtual machine identifier.

XS bool

XS Attribute.

nG enum

Flag indicating whether ASID will be matched.

MMU_TTB_READ

This event is triggered by reads caused by a translation table walk. Fields:

ABORTED bool

Was the walk successful.

DATA unsigned int

The data read.

DESC_KIND enum

The encoding scheme of the descriptors used for this walk.

IPA unsigned int

For stage 1, the IPA of the read.

LEVEL unsigned int

Translation table level.

LPAE bool

Is this for an LPAE translation (*DEPRECATED*, use DESC_KIND).

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSmem bool

Non secure memory access.

NSreq bool

Non secure request.

OUTADDR unsigned int

The output physical address.

PADDR unsigned int

The physical address of the read.

PASmem enum

Physical Address Space of the accessed memory.

PASreq enum

Physical Address Space of the requested lookup.

REGIME_EL enum

Entry matches in this translation regime.

REQUESTER enum

The requester of this table walk.

SIDE enum

Inst / Data.

STAGE unsigned int

Translation stage.

MMU_TTB_WRITE

This event is triggered by writes caused by a translation table walk. Fields:

ABORTED bool

Was the walk successful.

DATA unsigned int

The data written.

DESC_KIND enum

The encoding scheme of the descriptors used for this walk.

IPA unsigned int

For stage 1, the IPA of the write.

IS_HACDBS_ACCESS bool

If the access was due to HACDBS_CLEAN.

LEVEL unsigned int

Translation table level.

LPAE bool

Is this for an LPAE translation (*DEPRECATED*, use DESC_KIND).

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSmem bool

Non secure memory access.

NSreq bool

Non secure request.

OUTADDR unsigned int

The output physical address.

PADDR unsigned int

The physical address of the write.

PASmem enum

Physical Address Space of the accessed memory.

PASreq enum

Physical Address Space of the requested lookup.

REGIME_EL enum

Entry matches in this translation regime.

SIDE enum

Inst / Data.

STAGE unsigned int

Translation stage.

MODE_CHANGE

Mode change. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

MODE enum

The new mode.

NON_SECURE enum

The core's new non-secure bit.

OLD_MODE enum

The old mode.

SECURITY_STATE enum

The new security state.

NOT_TAG_CHECKED_INST

MTE Tag not checked for current instruction.

PERIODIC

Called for every quantum. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The instruction count of this CPU.

PC unsigned int

The address of the next instruction to be executed on this CPU.

PMU_COUNTER_OVERFLOW

PMU counter overflow. Fields:

COUNTER_GROUP enum

Counter group.

COUNTER_RANGE enum

Counter range.

EVENT_ID unsigned int

Event Identifier, field is valid only for Event counter group.

INDEX unsigned int

Counter index (as selected by PMSELR), field is valid only for Event/Cycle counter groups.

INTERRUPT bool

Is interrupt enabled on overflow for this counter.

POTINDEX_UPDATE

Value of POT index for current pc changed. Fields:

NEW_POTINDEX unsigned int

New POTIndex.

OLD_POTINDEX unsigned int

Old POTIndex.

PREFETCH_MEMORY64

Prefetch from PRFM or PRFUM instructions. Fields:

PRFOP enum

Prefetch hint.

VADDR unsigned int

Virtual address of the location that should be prefetched.

PRELOAD64_REQUEST

Preload requests from PRFM or PRFUM instructions. Fields:

PRFOP enum

Prefetch hint.

VADDR unsigned int

Virtual address of the location that should be prefetched.

PRELOAD_DATA

Data preload from PLD instruction. Fields:

VADDR unsigned int

Virtual address of the data that should be preloaded.

PRELOAD_INST

Instruction preload from PLI instruction. Fields:

VADDR unsigned int

Virtual address of the instruction that should be preloaded.

PRE_CORE_LOAD

Trace just before a core load. Fields:

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

PRE_CORE_STORE

Trace just before a core store. Fields:

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

PRE_TTB_READ

This event is triggered before reads caused by a translation table walk. Fields:

ADDR unsigned int

The physical address of the read.

PSTATE

PSTATE fields update. Fields:

ALLINT unsigned int

All interrupt mask bit.

BTYPE unsigned int

Branch target identification bit.

CORE_NUM unsigned int

Core number in a multi processor.

DAIF unsigned int

Exception mask bits.

DIT unsigned int

Data Independent Timing bit.

EL unsigned int

Current Exception level field.

EXLOCK unsigned int

Exception return state lock bit.

IL unsigned int

Illegal Execution state bit.

NZCV unsigned int

Condition flags.

PACM unsigned int

PACM bit.

PAN unsigned int

Privileged Access Never state bit.

PM unsigned int

PMU exception mask bit.

PPEND unsigned int

PMU exception pending bit.

SM unsigned int

StreamingMode status bit.

SP unsigned int

Stack pointer register selection bit.

SS unsigned int

Software Step bit.

SSBS unsigned int

Speculative Store Bypass Safe bit.

TCO unsigned int

Tag Check Override bit.

TOUCHED unsigned int

Mask of the fields that have been touched since this trace has last been emitted. Note that touched does not necessarily mean the value changed. The index of the field is the bit index in the mask plus the index of this field plus one.

UAO unsigned int

User Access Override bit.

UINJ unsigned int

Undefined Instruction exceptions bit.

ZA unsigned int

ZA storage and ZT0 register enable bit.

nRW unsigned int

Current Execution state bit.

PSTATE_PACTIVE_OPMODE_TRANSITION

Trace pactive and Opmode mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Operating mode after the transition.

OLD_STATE enum

Operating mode before the transition.

PSTATE_PACTIVE_OPMODE_TRANSITION_DENIED

Trace the denied pactive and Opmode mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Operating mode after the transition.

OLD_STATE enum

Operating mode before the transition.

REASON string

Reason for the transition being denied.

PSTATE_PACTIVE_POWER_TRANSITION

Trace pactive and power mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Power mode after the transition.

OLD_STATE enum

Power mode before the transition.

PSTATE_PACTIVE_POWER_TRANSITION_DENIED

Trace the denied pactive and power mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Power mode after the transition.

OLD_STATE enum

Power mode before the transition.

REASON string

Reason for the transition being denied.

PSTATE_PACTIVE_THREAD_OPMODE_TRANSITION

Trace pactive and Opmode mode transitions for threads on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Operating mode after the transition.

OLD_STATE enum

Operating mode before the transition.

QUANTUM_START

Called at the start of every quantum.

RANGE_PRELOAD_REQUEST

Preload requests from RPRFM. Fields:

COUNT unsigned int

Number of blocks of data to access.

LENGTH unsigned int

Number of contiguous bytes to be accessed in each block.

OP unsigned int

Range prefetch operation .

REUSE unsigned int

Maximum number of bytes to be accessed before next RPRFM .

STRIDE signed int

Stride to advance after accessing LENGTH bytes.

VADDR unsigned int

Base virtual address of range prefetch request.

RUN_STATE

Run state transition. Fields:

INST_COUNT unsigned int

Ticks count at point of transition.

NEW enum

New run state.

OLD enum

Old run state.

SBP_SAMPLE_COLLISION

DISPLAY Statistical profiling samples collided .

SBP_SAMPLE_DISCARDED

DISPLAY Operation discarded by statistical profiling unit when configured in discard mode.

Fields:

BUFFER_POINTER unsigned int

Value of buffer pointer.

DATA unsigned int

Data discarded.

SIZE unsigned int

Size of sample, in bytes.

SBP_SAMPLE_END

DISPLAY Statistical profiling of an operation ended.

SBP_SAMPLE_POP

DISPLAY Operation which might be sampled by statistical profiling.

SBP_SAMPLE_RECORDED

DISPLAY Operation sampled by statistical profiling has been written to the buffer at %{BUFFER_ADDRESS} with size %{SIZE} and data %{DATA}. Fields:

BUFFER_ADDRESS unsigned int

Virtual Address of start of buffer write.

DATA unsigned int

Data written to the buffer.

SIZE unsigned int

Size of sample written to buffer, in bytes.

SBP_SAMPLE_START

DISPLAY Statistical profiling of an operation started.

SBP_SAMPLE_TAKEN

DISPLAY Operation sampled by statistical profiling after filtering.

SEMIHOSTING_CALL

Call of a semihost function occurred. Fields:

PC unsigned int

The program counter after the semihosting call.

SEMIHOSTING_PRECALL

About to call semihost. Fields:

EL2 bool

Use EL2 translation regime.

NS bool

Is Non-Secure.

REG_WIDTH unsigned int

The current register width in bytes.

SIGNAL

External signal state change. Fields:

SIGNAL enum

Signal that changed.

STATE bool

Signal asserted state.

SOFTWARE_STEP

Return the Debug Software Step state. Fields:

STATE enum

Software Step state.

SPSR

SPSR change. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

MODE enum

Which of the banked SPSR registers is written.

OLD_VALUE unsigned int

The old SPSR value.

VALUE unsigned int

The new SPSR value.

START_COMPILE

Compilation started. Fields:

VADDR unsigned int

Instruction where compilation begins.

STG_MULTIPLE_MICRO_OPS_ADDRESS_RANGE

Address range for MTE stores with multiple uops. Fields:

ADDR_END unsigned int

Final address.

ADDR_START unsigned int

Starting address.

STORE_MULTIPLE_REGS_ADDRESS_RANGE

Address range for stores with multiple registers. Fields:

ADDR_END unsigned int

Final address.

ADDR_START unsigned int

Starting address.

SUBOPTIMAL_LDST_RETIRED

suboptimal load/store. Fields:

ACCESS_SIZE unsigned int

Log2 of the access width in bytes used for alignment checking (1: half-word, 2: word ...).

OPTIMAL_ALIGN_SELECTOR unsigned int

Selector to decide the optimal alignment size (1: access width, 2: 4-bytes, 3: 8-bytes, ..., 12: 4-kbytes).

VADDR unsigned int

The virtual address of the access.

SYNC

Called for every synchronization, that is every quantum break. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The instruction count of this CPU.

LOCAL_QUANTUM unsigned int

The local quantum of this CPU.

LOCAL_TIME unsigned int

The local time of this CPU.

SYNC_REASON enum

Reason for sync.

SYSCALL

System call instruction executed. Fields:

IMM unsigned int

Immediate value of the system call instruction.

TYPE enum

System call type.

VADDR unsigned int

Instruction that caused the system call.

SYSREG_READ128

System Coprocessor 128 bit register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

REGNUM enum

Internal register number.

REGNUM_PHYS enum

Internal register number physically accessed (e.g. in presense of VHE redirection).

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_READ64

System Coprocessor register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REGNUM enum

Internal register number.

REGNUM_OPERAND enum

Register number corresponding to the instruction operand.

REGNUM_PHYS enum

Internal register number physically accessed (e.g. in presense of VHE redirection).

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_UPDATE128

Triggers when the system updates a register. Fields:

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG enum

Register number.

UNKNOWN unsigned int

Bits of the register which became unknown.

VALUE unsigned int

Value written to the register.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int
opcode 2.

SYSREG_UPDATE32

Triggers when the system updates a register. Fields:

CRm unsigned int
CRm.

CRn unsigned int
CRn.

NS bool
Secure or nonsecure banked register is accessed.

REG enum
Register number.

UNKNOWN unsigned int
Bits of the register which became unknown.

VALUE unsigned int
Value written to the register.

opc unsigned int
opcode 1.

opc0 unsigned int
opcode 0.

opc2 unsigned int
opcode 2.

SYSREG_UPDATE64

Triggers when the system updates a register. Fields:

CRm unsigned int
CRm.

CRn unsigned int
CRn.

NS bool
Secure or nonsecure banked register is accessed.

REG enum
Register number.

UNKNOWN unsigned int
Bits of the register which became unknown.

VALUE unsigned int
Value written to the register.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_WRITE128

System Coprocessor 128 bit register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

REGNUM enum

Internal register number.

REGNUM_PHYS enum

Internal register number physically accessed (e.g. in presense of VHE redirection).

UNDEF bool

The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_WRITE64

System Coprocessor register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REGNUM enum

Internal register number.

REGNUM_OPERAND enum

Register number corresponding to the instruction operand.

REGNUM_PHYS enum

Internal register number physically accessed (e.g. in presense of VHE redirection).

UNDEF bool

The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

UNALIGNED_LDST_RETIRED

Processor unaligned load/store. Fields:

ACCESS_SIZE unsigned int

Log2 of the access width in bytes used for alignment checking (1=>half-word, 2->word ...).

NUMBER_OF_BEATS unsigned int

Number of accesses (beats) in this burst (Total size in bytes is ACCESS_SIZE * NUMBER_OF_BEATS).

VADDR unsigned int

The virtual address of the access.

VFP_D_REGS

VFP/NEON D 64 bit register write. Fields:

ALIASING enum

Type of register aliasing used.

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a write to the corresponding bit in the register. For example, 0x1 would indicate a write to VALUE[0].

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

VFP_D_REGS_READ

VFP/NEON D 64 bit register read. Fields:

ID unsigned int

The register number.

VALUE unsigned int

The current value of the register.

VFP_Q_REGS

VFP/NEON Q 128 bit register write. Fields:

ALIASING enum

Type of register aliasing used.

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a write to the corresponding byte in the register. For example, 0x1 would indicate a write to VALUE[7:0].

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

VFP_Q_REGS_READ

VFP/NEON Q 128 bit register read. Fields:

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a read to the corresponding byte in the register. For example, 0x1 would indicate a read to VALUE[7:0].

VALUE unsigned int

The current value of the register.

VFP_SYS_REGS

Writes to the VFP/NEON units system registers. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ID enum

Which VFP system register is written.

MASK unsigned int

The mask indicating updated cumulative bits.

OLD_VALUE unsigned int

The register's old value overwritten.

VALUE unsigned int

The new value written to the VFP system register.

VFP_S_REGS

VFP/NEON S 32 bit register write. Fields:

ALIASING enum

Type of register aliasing used.

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a write to the corresponding bit in the register. For example, 0x1 would indicate a write to VALUE[0].

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

VFP_S_REGS_READ

VFP/NEON S 32 bit register read. Fields:

ID unsigned int

The register number.

VALUE unsigned int

The current value of the register.

VIRTUAL_TAG_ALLOCATED

Virtual tag information when allocated. Fields:

TAG_VADDR unsigned int

Virtual tag virtual address.

VADDR unsigned int

Data Virtual address.

VIRTUAL_TAG unsigned int

Virtual tag value.

WAYPOINT

Signals end of basic block execution. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ISET enum

Origin instruction set.

IS_COND bool

Indicates if this is a conditional waypoint.

PC unsigned int

Origin address (or 0 if unavailable).

TAKEN bool

Indicates if this waypoint was taken.

TARGET unsigned int

Destination address.

TARGET_ISET enum

Destination instruction set.

WFE_END

WFE ended. Fields:

INST_COUNT unsigned int

Ticks count when leaving WFE.

WFE_EVENT_REGISTER

WFE event register status: set/clear, reason. Fields:

INST_COUNT unsigned int

Ticks count.

REASON enum

Reason for set/clear. Only REASON==0 (Cleared by WFE) clears the bit, all other reasons set it.

WFE_IGNORED

WFE ignored. Fields:

EVENT bool

This WFE was ignored because the event register was set.

INST_COUNT unsigned int

Ticks count when ignoring WFE.

TRAPPED bool

This WFE was trapped.

WFE_START

WFE entered. Fields:

INST_COUNT unsigned int

Ticks count when entering WFE.

WFI_END

WFI ended. Fields:

INST_COUNT unsigned int

Ticks count when leaving WFI.

WFI_IGNORED

WFI ignored. Fields:

DISABLED bool

This WFI was ignored because WFI is disabled.

INST_COUNT unsigned int

Ticks count when ignoring WFI.

REASON enum

specifies reason why WFI trace was ignored.

TRAPPED bool

This WFI was trapped.

WFI_START

WFI entered. Fields:

INST_COUNT unsigned int

Ticks count when entering WFI.

WFI_WAKEUP

WFI wakeup. Fields:

INST_COUNT unsigned int

Ticks count when WFI wakeup occurred.

REASON enum

Reason for wakeup.

2.65 ARM_Neoverse-N2

This section describes the trace sources.

AA64_ASE_SVE_REGS

Modification of AArch64 Advanced SIMD or SVE Vector Registers (via SIMD or Scalar operations), SVE Predicate Registers and ZT0. Fields:

ID enum

The register identifier.

MASK unsigned int

Bitmask of the register to signal the modified bits in the VALUE field.

SM bool

Whether the PE is in Streaming Mode.

VALUE unsigned int

Value of AArch64 vector register.

ASYNC_MEMORY_FAULT

Context ID Register write. Fields:

FAULT unsigned int

Fault status present in ESR.ISS encoding for Serror in ARCH64 and present in DFSR encoding for external abort format in ARCH32.

PADDR unsigned int

Physical Address (or 0 if unavailable).

VADDR signed int

Virtual Address (or 0 if unavailable).

ATOMIC_END_ACCESS

Bus trace access for atomics. Fields:

ACCESS_FAIL bool

Memory access failed.

ADDR unsigned int

The virtual address of the access.

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory

attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

COMPARE_VALUE unsigned int

Compare value for CAS.

LOAD_VALUE unsigned int

Loaded values.

NSDESC unsigned int

The security state of the access.

OPERAND_VALUE unsigned int

Operation's operand.

OPERATION enum

Operation type.

PADDR unsigned int

The physical address of the access.

PRIV bool

Is this a privileged access?.

ATOMIC_START_ACCESS

Bus trace access for atomics. Fields:

ADDR unsigned int

The virtual address of the access.

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

COMPARE_VALUE unsigned int

Compare value for CAS.

NSDESC unsigned int

The security state of the access.

OPERAND_VALUE unsigned int

Operation's operand.

OPERATION enum

Operation type.

PADDR unsigned int

The physical address of the access.

PRIV bool

Is this a privileged access?.

ArchMsg.Error.exit_code

DISPLAY exit code: %{EXITCODE} component kind: %{KIND} reason: %{REASON}. Fields:

EXITCODE signed int

Exit code to be returned.

KIND string

Component kind that invoked the exit code trace.

REASON string

Reason why the exit code trace was invoked (optional).

ArchMsg.Warning.IgnoredWarmResetRequest

Event fires when software attempts to request warm reset through write to deprecated DBGPRCR.CWRR.

ArchMsg.Warning.RAS_ErrselrUnpredictableState

DISPLAY ERRSELR is in unpredictable state due to value greater than number_of_error_records written to it.

ArchMsg.Warning.RAS_UnpredictableStateWrite

DISPLAY %{REG_NAME} write ignored due to unpredictable ERRSELR state. Fields:

REG_NAME string

Name of register being written.

ArchMsg.Warning.ReportedRevisionRegisterRead

DISPLAY Reported revision and patch do not match that of the cpu. Fields:

ORIGINAL_VALUE unsigned int

The value the register would have based on the modelled revision.

REGISTER string

The name of the register that has ben read.

REPORTED_VALUE unsigned int

The value the register has been overridden to.

ArchMsg.Warning.Unpredictably Indexed PM Event Register

TAGS UNPREDICTABLE DISPLAY Event %{IsDirect:(Selector value|Register)} %{SEL} is >= %{N}. Fields:

IsDirect unsigned int

Direct Access rather than indirect.

N unsigned int

Number of Accessible Registers.

SEL unsigned int

Selector.

ArchMsg.Warning.cache_contents_unknown

A cache has been activated without being invalidated and may contain **UNKNOWN** content. DISPLAY %{SIDE:(D|I)}-Cache was enabled in %{REGIME:(SECURE|NON_SECURE|HYP)} regime but was not invalidated since power-on: cache lines could contain **UNKNOWN** content %{SIDE:(Note: D-side caches must invalidate SECURE lines even if the cache is only used by NON_SECURE code in order to ensure that unexpected dirty lines tagged SECURE are not naturally evicted)}. Fields:

REGIME unsigned int

Current translation regime.

SIDE unsigned int

0 d-side cache or unified cache, 1 i-side.

ArchMsg.Warning.dcimvac_matches_watchpoint

TAGS **UNPREDICTABLE** DISPLAY Watchpoints matching an AArch32 DCIMVA are implementation defined.

ArchMsg.Warning.decode_fieldmismatch

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with different fields %{FIELD1} and %{FIELD2}. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_fracbitsnegative

TAGS **UNPREDICTABLE** DISPLAY VCVT Instruction is unpredictable with a negative number of fraction bits.

ArchMsg.Warning.decode_initblock

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable in an IT block.

ArchMsg.Warning.decode_initblocknotlast

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable in an IT block when not the last.

ArchMsg.Warning.decode_invalid_condition

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable when the condition is not AL.

ArchMsg.Warning.decode_invalidfieldcombination

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with the values in %{FIELD1} and %{FIELD2}. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_invalidvalue

TAGS UNPREDICTABLE DISPLAY Instruction is unpredictable with the value in %{FIELD}. Fields:

FIELD enum

Field name.

ArchMsg.Warning.decode_mem_hint_unallocated

TAGS UNPREDICTABLE DISPLAY There is no memory hint allocated to this bit pattern.

ArchMsg.Warning.decode_registermatch

TAGS UNPREDICTABLE DISPLAY Instruction is unpredictable with %{FIELD1} and %{FIELD2} as the same register. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_registermismatch

TAGS UNPREDICTABLE DISPLAY Instruction is unpredictable with %{FIELD1} and %{FIELD2} as different registers. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_registeroutofrange

TAGS UNPREDICTABLE DISPLAY Floating-point instruction is unpredictable indexing beyond the end of the register bank.

ArchMsg.Warning.decode_sbzsbo

TAGS UNPREDICTABLE DISPLAY Reserved bits in the instruction are not canonical.

ArchMsg.Warning.decode_transactiontoobig

TAGS UNPREDICTABLE DISPLAY Attempt to perform a memory transaction of more than 128 bytes.

ArchMsg.Warning.decode_unpred_other

TAGS UNPREDICTABLE DISPLAY Instruction is unpredictable.

ArchMsg.Warning.decode_unpreduseofpc

TAGS UNPREDICTABLE DISPLAY Instruction is unpredictable when %{FIELD} is PC. Fields:

FIELD enum

Field name.

ArchMsg.Warning.decode_unpreduseofr13

TAGS UNPREDICTABLE DISPLAY Instruction is unpredictable when %{FIELD} is R13. Fields:

FIELD enum

Field name.

ArchMsg.Warning.decode_unpreduseofr13inreglist

TAGS UNPREDICTABLE DISPLAY Instruction is unpredictable with R13 in register-list.

ArchMsg.Warning.decode_useofd16orhigher

TAGS UNPREDICTABLE DISPLAY Attempt to use register D16 or higher where it is not allowed.

ArchMsg.Warning.decode_writebackandbaseinlist

TAGS UNPREDICTABLE DISPLAY Instruction is unpredictable with write back when the base register is in the transfer list.

ArchMsg.Warning.decode_zeroregistersinlist

TAGS UNPREDICTABLE DISPLAY Instruction is unpredictable when no registers are to be transferred.

ArchMsg.Warning.misaligned_ptrs

DISPLAY TRBPTR_EL1(%{TRBPTR_EL1}) or TRBTRG_EL1(%{TRBTRG_EL1}) are not aligned as per TRBIDR_EL1.Align(%{TRBIDR_EL1}). Fields:

TRBIDR_EL1 unsigned int

TRBIDR_EL1.Align value.

TRBPTR_EL1 unsigned int

Current write pointer.

TRBTRG_EL1 unsigned int

Trigger pointer.

ArchMsg.Warning.out_of_range_ptrs

DISPLAY TRBPTR_EL1(%{TRBPTR_EL1}) is out of range as per TRBBASER_EL1(%{TRBBASER_EL1}) or TRBLIMITR_EL1.LIMIT(%{TRBLIMITR_EL1}). Fields:

TRBBASER_EL1 unsigned int

Base pointer.

TRBLIMITR_EL1 unsigned int

Limit pointer.

TRBPTR_EL1 unsigned int

Current write pointer.

ArchMsg.Warning.recursive_branch

An instruction is performing a branch that targets the same instruction. If this was intended then a WFI might be more effective DISPLAY recursive branch detected at PC=%{PC}. PRIMARY KEY PC. Fields:

PC unsigned int

Address of instruction causing the branch.

ArchMsg.Warning.recursive_exception

An instruction has generated an exception that targets the same instruction. DISPLAY recursive %{TYPE} exception detected at PC=%{PC} (CPSR=%{CPSR}). PRIMARY KEY PC. Fields:

CPSR unsigned int

Processor state.

PC unsigned int

Address of instruction causing the exception.

TYPE enum

Type of exception.

ArchMsg.Warning.reentrant_vector_catch

TAGS **UNPREDICTABLE** DISPLAY Vector catch debug event on Prefetch or Data abort vector %{ADDR}. Fields:

ADDR unsigned int

vector address.

ArchMsg.Warning.reserved_it_state

TAGS **UNPREDICTABLE** DISPLAY Execution with reserved IT state %{ITSTATE}. Fields:

ITSTATE unsigned int

reserved IT state.

ArchMsg.Warning.tlb_contents_unknown

Translation has been enabled without invalidating the TLB. DISPLAY Translation was enabled in the %{PAS:(Secure|Non-secure|Root|Realm)} EL%{EL:d} translation regime (vmid=%{VMID:x}) without invalidating %{INVALIDITY:(|d-side |i-side |)}entries since power-on: %{INVALIDITY:(|D||)}TLB could contain **UNKNOWN** content. Fields:

EL unsigned int

Exception level of translation regime.

INVALIDITY unsigned int

Which TLB may be invalid (if TLBs are separate).

NON_SECURE bool

Whether the translation regime is non-secure.

PAS unsigned int

Physical address space of translation regime.

VMID unsigned int

Current VMID, if applicable.

ArchMsg.Warning.unknown_DLR_DSPSR

Exiting debug state with unknown DLR or DSPSR.

ArchMsg.Warning.unknown_ELR_SPSR

Returning from debug exception with unknown ELR or SPSR.

ArchMsg.Warning.unpred_FAR_on_LRCPC3_non_atomic_mem_access

TAGS **UNPREDICTABLE** Unpredictable FAR/FSTATUS on LRCPC3 pair instructions resulting in two separate memory accesses
 DISPLAY Unpredictable FAR on LRCPC3 non atomic memory access at %{ADDR1} with FSTATUS %{FSTATUS1} or %{ADDR2} with FSTATUS %{FSTATUS2}. Fields:

ADDR1 unsigned int

Possible FAR address 1.

ADDR2 unsigned int

Possible FAR address 2.

FSTATUS1 unsigned int

Possible fault status value 1.

FSTATUS2 unsigned int

Possible fault status value 2.

ArchMsg.Warning.unpred_async_tag_check_fail_on_sve_speculative_load_do_update_ffr

Unpredictable set to FALSE of the FFR element associated to SVE speculative load with an async tag check fail TAGS **UNPREDICTABLE**.

ArchMsg.Warning.unpredictable_a32_breakpoint

TAGS **UNPREDICTABLE** DISPLAY Unpredictable breakpoint on a A32 instruction at %{ADDR1} Offset %{OFFSET} BAS %{BAS} for a %{IS_ADDRESS_MISMATCH_BREAKPOINT:(Mismatch|Match)} Breakpoint number %{N:d} Breakpoint address %{ADDR2}. Fields:

ADDR1 unsigned int

Address.

ADDR2 unsigned int

Breakpoint address.

BAS unsigned int

BAS field.

IS_ADDRESS_MISMATCH_BREAKPOINT bool

Is a Mismatch breakpoint, not a Match breakpoint.

N unsigned int

Breakpoint number.

OFFSET unsigned int

Always has a value of zero.

ArchMsg.Warning.unpredictable_banked_register_access

TAGS **UNPREDICTABLE** DISPLAY Access to banked register (R=%{R:d} SYSm=%{SYSM:2x}) is unpredictable. Fields:

R unsigned int

R bit.

SYSM unsigned int

SYSm.

ArchMsg.Warning.unpredictable_ccfail_undef

TAGS **UNPREDICTABLE** DISPLAY An instruction %{OPCODE} at VA=%{VADDR} failed its condition codes check but would otherwise have caused a non-data-dependent trap or undefined exception. Behaviour is implementation-defined choice of exception or nop. Fields:

OPCODE unsigned int

opcode of the instruction.

VADDR unsigned int

Virtual address of the instruction being executed.

ArchMsg.Warning.unpredictable_ici_bits

TAGS **UNPREDICTABLE** DISPLAY ICI bits do not apply or are architecturally unpredictable for instruction.

ArchMsg.Warning.unpredictable_t32_breakpoint

TAGS **UNPREDICTABLE** DISPLAY Unpredictable breakpoint on a T32 instruction at %{ADDR1} Offset %{OFFSET} BAS %{BAS} for a %{IS_ADDRESS_MISMATCH_BREAKPOINT:(Mismatch|Match)} Breakpoint number %{N:d} Breakpoint address %{ADDR2}. Fields:

ADDR1 unsigned int

Address.

ADDR2 unsigned int

Breakpoint address.

BAS unsigned int

BAS field.

IS_ADDRESS_MISMATCH_BREAKPOINT bool

Is a Mismatch breakpoint, not a Match breakpoint.

N unsigned int

Breakpoint number.

OFFSET unsigned int

Indicates which halfword of a 32-bit Thumb instruction the breakpoint is set on. 0 for the first halfword or if the instruction is 16 bits wide, or 1 for the second halfword.

ArchMsg.Warning.unpredictable_t32_vector_catch

TAGS **UNPREDICTABLE** DISPLAY Unpredictable Vector catch exception on a T32 instruction at %{ADDR} Offset %{OFFSET}. Fields:

ADDR unsigned int

Address.

OFFSET unsigned int

Indicates which halfword of a 32-bit Thumb instruction the vector catch exception is set on. 0 for the first halfword or if the instruction is 16 bits wide, or 1 for the second halfword.

ArchMsg.Warning.unpredictable_watchpoint_far

TAGS **UNPREDICTABLE** DISPLAY The value for FAR/EDWAR for hitting this watchpoint may be anywhere in the range %{LOWER_BOUND} to %{UPPER_BOUND}. Fields:

LOWER_BOUND unsigned int

the lowest address accessed by the instruction that triggered the watchpoint.

UPPER_BOUND unsigned int

the highest watchpointed address accessed by that instruction.

ArchMsg.Warning.unpredictable_watchpoint_far_isncmp

DISPLAY The value for FAR/EDWAR for hitting this watchpoint may be anywhere in the range %{LOWER_BOUND} to %{UPPER_BOUND}. Fields:

LOWER_BOUND unsigned int

the lowest address accessed by the instruction that triggered the watchpoint.

UPPER_BOUND unsigned int

the highest watchpointed address accessed by that instruction.

ArchMsg.Warning.warning_AdvSIMDExpandImmUnexpectedZero

TAGS **UNPREDICTABLE** DISPLAY AdvSIMDExpandImm may treat this immediate value as **UNPREDICTABLE**.

ArchMsg.Warning.warning_ConditionalSMC

TAGS **UNPREDICTABLE** DISPLAY SMC instruction has **UNPREDICTABLE** effects when conditional and when combined with traps.

ArchMsg.Warning.warning_TLBID_out_of_range

TLBID out of range in TLBI according to the configuration of NIS, NVIS, NOS and NVOS. Fields:

NIS unsigned int

NIS.

NOS unsigned int

NOS.

NVIS unsigned int

NVIS.

NVOS unsigned int

NVOS.

TLBID unsigned int

The TLBID value used in the TLBI instruction.

ArchMsg.Warning.warning_TTL_or_TG_mismatches_current_translation_granule

TTL or TG mismatches current translation granule in TLBI VA/RVA/IPA/RIPA. Fields:

CURRENT_TG unsigned int

Current TG as indicated by TCR_ELx or VTCR_EL2.

TLBI_TG unsigned int

TG (or TTL[3:2]) used in TLBI instruction.

ArchMsg.Warning.warning_access_crosses_page_boundary

TAGS **UNPREDICTABLE** DISPLAY Access crosses page boundary.

ArchMsg.Warning.warning_access_crosses_page_boundary_has_fault

TAGS **UNPREDICTABLE** DISPLAY Access that crosses page boundary has fault(s).

ArchMsg.Warning.warning_access_crosses_page_boundary_has_fault_on_both_pages

TAGS **UNPREDICTABLE** DISPLAY Access that crosses page boundary has a fault on each page.
Fields:

ADDR1 unsigned int

FAR address on first page.

ADDR2 unsigned int

FAR address on second page.

FSTATUS1 unsigned int

Fault status on first page.

FSTATUS2 unsigned int

Fault status on second page.

ArchMsg.Warning.warning_access_crosses_page_boundary_spanning_different_memory

TAGS **UNPREDICTABLE** DISPLAY Access at address %{ADDR} crosses page from MEM_TYPE %{MEMTYPE_PAGE1} to %{MEMTYPE_PAGE2}. Fields:

ADDR unsigned int

address of access crossing page.

MEMTYPE_PAGE1 signed int

First Page's memory type (0-Normal, 1-Non-Normal).

MEMTYPE_PAGE2 signed int

Second Page's memory type (0-Normal, 1-Non-Normal).

ArchMsg.Warning.warning_access_wraps_around_memory

TAGS **UNPREDICTABLE** DISPLAY Access wraps around memory in %{WIDTH:d}bit mode.
Fields:

WIDTH unsigned int

Address width.

ArchMsg.Warning.warning_bcr_linking_status

TAGS **UNPREDICTABLE** DISPLAY DBGBCR.{BT2,BT} combination gives unpredictable behaviour.

ArchMsg.Warning.warning_bcr_mask_reserved

TAGS **UNPREDICTABLE** DISPLAY DBGBCR_ELn.MASK is set to a reserved value.

ArchMsg.Warning.warning_bvr_masked_bit_not_zero

TAGS **UNPREDICTABLE** DISPLAY DBGBVRn_EL1.VA contains a bit masked by DBGBCRn_EL1.MASK that is not zero.

ArchMsg.Warning.warning_ccsidr_unimplemented_level

TAGS **UNPREDICTABLE** DISPLAY CCSIDR read while CSSELR %{CSSELR:x} points to an unimplemented cache level. Fields:

CSSELR unsigned int

current effective value of CSSELR.

ArchMsg.Warning.warning_change_to_ns_when_tge_set

TAGS **UNPREDICTABLE** DISPLAY Attempting to change NS to 1 when HCR.TGE = 1.

ArchMsg.Warning.warning_contiguous_bit_check_abort

TAGS **UNPREDICTABLE** DISPLAY An abort occurred whilst attempting to check TLB entry is contiguous at %{ENTRY_ADDR}. Fields:

ENTRY_ADDR unsigned int

address of conflicting TLB entry.

ArchMsg.Warning.warning_contiguous_bit_error

TAGS **UNPREDICTABLE** DISPLAY TLB entry at %{ENTRY_ADDR} was not contiguous with entry at %{CONTIG_ADDR} Contents are %{ENTRY_DATA} but expected %{CONTIG_DATA}. Fields:

CONTIG_ADDR unsigned int

address of the TLB first read, with which this entry is expected to be contiguous.

CONTIG_DATA unsigned int

expected contents based on the entry at CONTIG_ADDR.

ENTRY_ADDR unsigned int

address of conflicting TLB entry.

ENTRY_DATA unsigned int

contents of conflicting TLB entry.

ArchMsg.Warning.warning_contiguous_bit_error_D128

TAGS **UNPREDICTABLE** DISPLAY TLB entry at %{ENTRY_ADDR} was not contiguous with entry at %{CONTIG_ADDR} Contents are %{ENTRY_DATA_HI} %{ENTRY_DATA_LO} but expected %{CONTIG_DATA}. Fields:

CONTIG_ADDR unsigned int

address of the TLB first read, with which this entry is expected to be contiguous.

CONTIG_DATA unsigned int

expected contents based on the entry at CONTIG_ADDR.

ENTRY_ADDR unsigned int

address of conflicting TLB entry.

ENTRY_DATA_HI unsigned int

Higher bit contents of conflicting TLB entry.

ENTRY_DATA_LO unsigned int

Lower bit contents of conflicting TLB entry.

ArchMsg.Warning.warning_cp10_cp11_mismatch

TAGS **UNPREDICTABLE** DISPLAY Unpredictable configuration of CP10 and CP11 controls at EL
%{EL:d}. Fields:

EL unsigned int

EL owning the control.

ArchMsg.Warning.warning_cp10_cp11_reserved_value

TAGS **UNPREDICTABLE** DISPLAY Unpredictable configuration of CPACR CP10 (%{CP10}) or
CP11 (%{CP11}). Fields:

CP10 unsigned int

CP10 access.

CP11 unsigned int

CP11 access.

ArchMsg.Warning.warning_csselr_level_out_of_range

TAGS **UNPREDICTABLE** DISPLAY CSSELR.Level out of range, written value %{LEVEL} with only
%{IMPLEMENTED:d} levels implemented. Fields:

IMPLEMENTED unsigned int

number of cache levels implemented.

LEVEL unsigned int

written value of CSSELR.Level.

TYPE enum

The type of cache selected by CSSELR.InD and TnD.

ArchMsg.Warning.warning_debug_flow_control_bits_not_obeyed

TAGS **UNPREDICTABLE** DISPLAY Flow control bits for %{REGISTER} were not obeyed, causing
%{ERROR}. Fields:

ERROR enum

type of error resulting.

REGISTER enum

register under consideration.

ArchMsg.Warning.warning_debug_register_access_during_reset

TAGS **UNPREDICTABLE** DISPLAY Unpredictable %{IS_WRITE:(read|write)} access to debug register offset %{OFFSET} during reset. Fields:

IS_WRITE unsigned int

Write Not Read.

OFFSET unsigned int

Register Offset.

ArchMsg.Warning.warning_decode_cps_inconsistent_fields

TAGS **UNPREDICTABLE** DISPLAY The mode setting fields in the CPS instruction are inconsistent.

ArchMsg.Warning.warning_decode_invalid_state

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable in current exception-level/security state.

ArchMsg.Warning.warning_default_cacheable_mmu_on

TAGS **UNPREDICTABLE** DISPLAY Default cacheable is enabled (HCR.DC = 1) while MMU is enabled.

ArchMsg.Warning.warning_default_cacheable_vmmu_off

TAGS **UNPREDICTABLE** DISPLAY Default cacheable is enabled (HCR.DC = 1) while HCR.VM = 0.

ArchMsg.Warning.warning_deprecated_wvr_bit_2_set

TAGS **UNPREDICTABLE** DISPLAY DBGWVR%{N:d}_EL1=%{WVR:h} has bit 2 set, deprecated in ARMv8. Fields:

N unsigned int

index.

WVR unsigned int

DBGWVR.

ArchMsg.Warning.warning_eret_while_software_step_active_pending

Missing ISB between setting MDSCR_EL1.SS and ERET TAGS **UNPREDICTABLE**.

ArchMsg.Warning.warning_exclusive_to_non_normal

TAGS **UNPREDICTABLE** DISPLAY Exclusive Access to Non-Normal Memory %{ADDR}. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_exclusive_to_non_writeback

TAGS **UNPREDICTABLE** DISPLAY Exclusive Access to Non-Writeback-Cacheable Memory %{ADDR}. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_execute_from_device_memory

TAGS **UNPREDICTABLE** DISPLAY An attempt was made to execute from Device memory.

ArchMsg.Warning.warning_illegal_cpsr_mode

TAGS **UNPREDICTABLE** DISPLAY Writing an illegal or unimplement mode (%{NEW_MODE:h}) to CPSR was unpredictable. Fields:

NEW_MODE unsigned int

New value of CPSR.M.

ArchMsg.Warning.warning_illegal_srs_mode

TAGS **UNPREDICTABLE** DISPLAY Illegal or **UNPREDICTABLE** mode (%{MODE:h}) used for SRS instruction. Fields:

MODE unsigned int

mode for Banked SP.

ArchMsg.Warning.warning_implementation_defined_read_debug_register_in_SCS

Read Access to Debug registers from the processor is **IMPLEMENTATION DEFINED** TAGS **IMP_DEF**. Fields:

PPB_OFFSET unsigned int

debug register offset.

ArchMsg.Warning.warning_implementation_defined_sequential_security_transitions_supported

The behavior of sequential instruction fetches that cross from non-secure to secure memory and contain an SG instruction is **CONSTRAINED UNPREDICTABLE** TAGS **UNPREDICTABLE**.

ArchMsg.Warning.warning_implementation_defined_stack_limit_check_supported

It is **IMPLEMENTATION DEFINED** whether stack pointer limit checking is performed for this instruction TAGS **IMP_DEF**. Fields:

ADDRESS unsigned int

Address of instruction.

OPCODE unsigned int

instruction opcode.

ArchMsg.Warning.warning_implementation_defined_taggednonsharedmemorytype

TAGS **IMP_DEF** DISPLAY Memory attribute has Tagged, non shared memory type; it is **IMPLEMENTATION DEFINED** whether the memory location is treated as Tagged or non-Tagged.

ArchMsg.Warning.warning_implementation_defined_write_debug_register_in_SCS

Write Access to Debug registers from the processor is **IMPLEMENTATION DEFINED** TAGS **IMP_DEF**. Fields:

DATA unsigned int

data attempted to be written.

PPB_OFFSET unsigned int

debug register offset.

ArchMsg.Warning.warning_invalid_tcr_granule

TAGS UNPREDICTABLE DISPLAY TCR.TG{%TG_ID:(0|1)} bits have been set to a granule size {%REQUEST:d}K not implemented - using {%SUBSTITUTE:d}K. Fields:

REQUEST signed int

page size requested (or 0 for reserved).

SUBSTITUTE signed int

best guess available page size.

TG_ID bool

bits TG0 or TG1.

ArchMsg.Warning.warning_load_multiple_user_registers_from_user_mode

TAGS UNPREDICTABLE DISPLAY An LDM(user registers) instruction executed from user mode.

ArchMsg.Warning.warning_mte_in_mem_attr_not_supported

TAGS UNPREDICTABLE DISPLAY Memory attribute {%ATTR} has Tagged memory type but Armv8.5-MTE is not implemented. Fields:

ATTR unsigned int

Memory attribute.

ArchMsg.Warning.warning_reserved_breakpoint_state_match

TAGS UNPREDICTABLE DISPLAY {%IS_BREAKPOINT:(Watchpoint|Breakpoint)} programmed with a reserved combination of HMC, SSC and {%IS_BREAKPOINT:(PAC|PMC)}. Fields:

IS_BREAKPOINT unsigned int

Is a breakpoint, not a watchpoint.

ArchMsg.Warning.warning_shareability

TAGS UNPREDICTABLE DISPLAY Unpredictable: combination of the 1st and 2nd stages of translation is Normal Inner Non-Cacheable, Outer Non-Cacheable.

ArchMsg.Warning.warning_software_step_set_while_enabled

TAGS UNPREDICTABLE DISPLAY MDSCR_EL1.SS set to 1 while software step debug exceptions are enabled.

ArchMsg.Warning.warning_thumb_instruction_wraps_around_memory

TAGS UNPREDICTABLE DISPLAY Thumb instruction wraps around memory.

ArchMsg.Warning.warning_ttbr_sbz_bits_are_not_zero

TAGS UNPREDICTABLE DISPLAY {%TTBR}[{%MSB:d}:0] should be zero, but are not. Fields:

MSB unsigned int

x-1.

TTBR enum

which TTBR register.

ArchMsg.Warning.warning_unaligned_address_dbgdtrrx_write

TAGS **UNPREDICTABLE** DISPLAY Unaligned address %{ADDR} in X0/R0 when DBGDTRRX written in memory access mode. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unaligned_address_dbgdtrtx_read

TAGS **UNPREDICTABLE** DISPLAY Unaligned address %{ADDR} in X0/R0 when DBGDTRTX read in memory access mode. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unaligned_to_device

TAGS **UNPREDICTABLE** DISPLAY Unaligned Access to Device Memory %{ADDR}. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unaligned_to_strongly_ordered

TAGS **UNPREDICTABLE** DISPLAY Unaligned Access to Strongly Ordered Memory %{ADDR}. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unknown_sau_rnr

TAGS **UNPREDICTABLE** DISPLAY SAU_RNR was set to an unsupported value.

ArchMsg.Warning.warning_unpredictable_AIRCR_PRIS_and_BFHFNMINs

TAGS **UNPREDICTABLE** DISPLAY The effect of setting both AIRCR.BFHFNMINs and AIRCR.PRIS to 1 is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_AIRCR_VECTCLRACTIVE_when_not_in_debug

TAGS **UNPREDICTABLE** DISPLAY The effect of writing a 1 to AIRCR.VECTCLRACTIVE if the processor is not halted in Debug state is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_AIRCR_VECTRESET_and_SYSRESETREQ

TAGS **UNPREDICTABLE** DISPLAY When the processor is halted in Debug state, if a write to the register writes a 1 to both VECTRESET and SYSRESETREQ, the behavior is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_AIRCR_VECTRESET_when_not_in_debug

TAGS **UNPREDICTABLE** DISPLAY The effect of writing a 1 to AIRCR.VECTRESET if the processor is not halted in Debug state is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_AIRCR_incorrect_VKEY

TAGS **UNPREDICTABLE** DISPLAY The value 0x05FA must be written to AIRCR.VKEY, otherwise the register write is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_EXC_RETURN_Reserved_Bit

EXC_RETURN[23:7] are reserved with the special condition that all bits should be written as one. Values other than all 1s are **UNPREDICTABLE**. TAGS **UNPREDICTABLE**. Fields:

EXC_RETURN unsigned int

Special PC value.

SBO_MASK unsigned int

Which bits Should Be One to not UNPRED.

ArchMsg.Warning.warning_unpredictable_PMEVTYPEPn_ELO_VS_reserved_value

TAGS **UNPREDICTABLE** DISPLAY Unpredictable reserved value of PMEVTYPEn<n>_ELO.VS.

ArchMsg.Warning.warning_unpredictable_change_to_DEMCR_MON_STEP_at_insufficient_priority

TAGS **UNPREDICTABLE** DISPLAY The effect of changing DEMCR.MON_STEP at an execution priority that is lower than the priority of the DebugMonitor exception is **CONSTRAINED UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_change_to_priority_of_active_exception

TAGS **UNPREDICTABLE** DISPLAY Changing the priority of an active exception is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_clear_lspact

TAGS **UNPREDICTABLE** DISPLAY **UNPREDICTABLE** state on context switch. Fields:

DRVEXC_TAKEN unsigned int

Derived exception taken, not pended.

REACHED_LAST_STORE unsigned int

Only the last faulted.

ArchMsg.Warning.warning_unpredictable_dbg_exit_unaligned_dlr

TAGS **UNPREDICTABLE** DISPLAY Debug exit to AArch32 with DLR[0] set to 1. PC[0] can be set to 0 or to DLR[0].

ArchMsg.Warning.warning_unpredictable_exception_catch

TAGS **UNPREDICTABLE** DISPLAY Generation of an exception catch event is unpredictable.

ArchMsg.Warning.warning_unpredictable_exception_return_inconsistent_state

TAGS **UNPREDICTABLE** DISPLAY state on exception return is unpredictable.

ArchMsg.Warning.warning_unpredictable_exception_return_instruction

TAGS **UNPREDICTABLE** DISPLAY exception return instruction is unpredictable.

ArchMsg.Warning.warning_unpredictable_in_debug_state

TAGS **UNPREDICTABLE** DISPLAY Instruction is **UNPREDICTABLE** when executed in debug state.

ArchMsg.Warning.warning_unpredictable_mair_encoding

TAGS **UNPREDICTABLE** DISPLAY Memory attribute %{ATTR} has invalid MAIR encoding; this may lead to unpredictable results. Fields:

ATTR unsigned int

Memory attribute.

ArchMsg.Warning.warning_unpredictable_pmu_counter_access

TAGS **UNPREDICTABLE** DISPLAY Access to PMU counters from non-secure EL0 or EL1 is **UNPREDICTABLE** with MDCR_EL2.HPMN set to 0.

ArchMsg.Warning.warning_unpredictable_prioritization_breakpoint_match_vector_catch

TAGS **UNPREDICTABLE** DISPLAY Two events with the same priority occurred at the same instruction: Address Matching Vector Catch debug event, and Breakpoint debug event. It is constrained unpredictable which is taken.

ArchMsg.Warning.warning_unpredictable_stack_selection

TAGS **UNPREDICTABLE** DISPLAY **UNPREDICTABLE** state on context switch.

ArchMsg.Warning.warning_unpredictable_tsize_out_of_range

TAGS **UNPREDICTABLE** DISPLAY %{TCR}.T%{TTBR:d}SZ (%{TSIZE:d} bits) is out of range. Must be between 25 and %{TSIZE_MAX:d}. Fields:

TCR enum

Which TCR.

TSIZE signed int

Value of TSize.

TSIZE_MAX signed int

The maximum allowed value.

TTBR unsigned int

Which TTBR region.

ArchMsg.Warning.warning_unpredictable_unaligned_pc_as_base_register

TAGS **UNPREDICTABLE** DISPLAY The PC value must be word-aligned, otherwise the behavior of the instruction is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_unaligned_pop_stack

TAGS **UNPREDICTABLE** DISPLAY **CONSTRAINED UNPREDICTABLE** unaligned stack frame %{FRAME_ADDR:x} in exception-return. Fields:

FRAME_ADDR unsigned int

unaligned stack frame address.

ArchMsg.Warning.warning_unpredictable_unaligned_to_non_writeback

TAGS **UNPREDICTABLE** DISPLAY Unaligned Access to Non-Writeback-Cacheable Memory.

ArchMsg.Warning.warning_unpredictable_vmsa_memattrib

TAGS **UNPREDICTABLE** DISPLAY Unpredictable vmsa memory attribute with texcb value programmed to %{TEXCB} when tex remap is %{TEX_REMAP}. Fields:

TEXCB unsigned int

texcb value.

TEX_REMAP bool

use tex remap.

ArchMsg.Warning.warning_unpredictable_vtcr_t0sz_sl0

TAGS **UNPREDICTABLE** DISPLAY The combination of VTCR.TOSZ=%{TOSZ:d} and VTCR.SLO=%{SLO:d} is unpredictable. Fields:

SLO signed int

VTCR.SLO, starting level for stage 2 translation table walks.

TOSZ signed int

VTCR.TOSZ, required input address range.

ArchMsg.Warning.warning_unpredictable_write_DHCSR

TAGS **UNPREDICTABLE** DISPLAY **UNPREDICTABLE** set of bit-changes in DHCSR. Fields:

NEW unsigned int

data attempted to be written.

OLD unsigned int

previous value.

ArchMsg.Warning.warning_unpredictable_write_SHCSR

TAGS **UNPREDICTABLE** DISPLAY **UNPREDICTABLE** setting a pending bit to 1 for an exception with priority >= execution priority.

ArchMsg.Warning.warning_unsupported_access_to_memory_mapped_register

TAGS **UNPREDICTABLE** DISPLAY Access to memory mapped register is unsupported.

ArchMsg.Warning.warning_user_jmcr_access

TAGS **UNPREDICTABLE** DISPLAY User %{WRITE:(write|read)} access to JMCR is unpredictable. Fields:

WRITE unsigned int

access is write.

ArchMsg.Warning.warning_wcr_mask_and_bas

TAGS **UNPREDICTABLE** DISPLAY DBGWCR_EL1.MASK is non-zero and BAS is not 0xff.

ArchMsg.Warning.warning_wcr_mask_reserved

TAGS **UNPREDICTABLE** DISPLAY DBGWCR_ELn.MASK is set to a reserved value.

ArchMsg.Warning.warning_wcr_non_configuous_bas

TAGS **UNPREDICTABLE** DISPLAY DBGWCR_EL1.BAS has non-contiguous set of ones.

ArchMsg.Warning.warning_wvr_masked_bit_not_zero

TAGS **UNPREDICTABLE** DISPLAY DBGWVRn_EL1.VA contains a bit masked by DBGWCRn_EL1.MASK that is not zero.

ArchMsg.Why.why_branch_target_exception

A branch target exception has occurred. DISPLAY branch target exception: target pc=%{PC}, target opcode=%{OPCODE}. PRIMARY KEY PC. Fields:

BTYPE unsigned int

pstate.BTYPE at time branch taken.

OPCODE unsigned int

opcode at branch target.

PC unsigned int

Virtual Address of branch target.

ArchMsg.Why.why_illegal_state

An illegal mode change has occurred. DISPLAY illegal mode change: %{MESSAGE}. Fields:

MESSAGE string

The message.

ArchMsg.warning_invalid_interval_counter_reload_value

Invalid interval counter reload value in PMSIRR_EL1.INTERVAL: software should set this to a value greater than or equal to the minimum indicated by PMSIDR_EL1.Interval.

ArchMsg.warning_invalid_spe_buffer_write_pointer

Invalid value programmed in PMBPTR_EL1.PTR or PMBLIMITR_EL1.LIMIT: Current write pointer must be at least one sample record below the write limit pointer and PMBPTR_EL1.PTR[63:56] must be equal to PMBLIMITR_EL1.LIMIT[63:56].

BRANCH_MISPREDICT

Simulating branch mispredict. Fields:

PC unsigned int

Origin address (or 0 if unavailable).

BRA_DIR

Direct branches, to immediate address. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISSET enum

The instruction set of the branch instruction.

IS_COND bool

Indicates if this is a conditional branch.

IS_HINTED bool

Whether the conditional instruction is hinted.

IS_ISB bool

Whether the instruction is ISB.

IS_LINK bool

Indicates if this is a branch with link.

PC unsigned int

The address of the branch instruction.

TARGET_ISET enum

The instruction set after the branch.

TARGET_PC unsigned int

The address the instruction branches to.

BRA_INDIR

Indirect branches, perhaps to a register. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISET enum

The instruction set of the branch instruction.

IS_COND bool

Indicates if this is a conditional branch.

IS_HINTED bool

Whether the conditional instruction is hinted.

IS_ISB bool

Whether the instruction is ISB.

IS_LINK bool

Indicates if this is a branch with link.

PC unsigned int

The address of the branch instruction.

TARGET_ISET enum

The instruction set after the branch.

TARGET_PC unsigned int

The address the instruction branches to.

BRA_INDIR_CCFAIL

Indirect branches fail, perhaps to a register.

BRA_RET_CCFAIL

Branch return fail, perhaps to a register.

CACHE_MAINTENANCE_OP

Cache Maintenance Operation. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

DATA unsigned int

Specified MVA or Set/way.

FUNCTION enum

Clean / Invalidate.

SCOPE enum

Affected region.

SIDE enum

Inst / Data.

CCFAIL

Conditional instruction condition check fail. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CCFAIL_UNC

Unconditional instruction condition check fail. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CCPASS

Conditional instruction condition check pass. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CCPASS_UNC

Unconditional instruction condition check pass. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CODE_CACHE_MAINT

Code cache maintenance. Fields:

END_INCL unsigned int

Inclusive end address for VA or PA requests.

NS bool

Non-secure world for PA.

START unsigned int

Start address for VA or PA requests.

TYPE enum

Request type.

COMPILE_BLOCK_END

Last instruction of basic block translated. Fields:

VADDR unsigned int

Address of next instruction after this basic block.

COMPILE_BLOCK_START

First instruction of basic block translated. Fields:

VADDR unsigned int

Address of first instruction in basic block.

COMPILE_INST

ARM instruction compiled. Fields:

DISASS string

Disassembly of the instruction.

ISSET enum

The instruction set of this instruction.

ITSTATE unsigned int

The ITSTATE current for the instruction.

OPCODE unsigned int

The opcode of the instruction.

PC unsigned int

The address of the instruction.

SIZE unsigned int

The size of the instruction in bytes.

CONTEXTIDR

Context ID Register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

NS bool

Secure or nonsecure banked register is accessed.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The new value written.

CONTEXT_SYNC

Called for every context synchronization event. Fields:

ARCH_CSE bool

Architectural Context Synchronization Event.

PC unsigned int

The address of the synchronization instruction.

SYSREG_SYNC bool

System register synchronization event.

CORE_ENDIAN

Core BE8 Big-Endian state changed. Fields:

BE8 bool

Core BE8 Big-Endian state.

CORE_INFO

Static processor attributes. Only triggered by a call to DumpState(). Fields:

ARCH_PROFILE enum

The architecture profile of the core.

CLUSTER_ID unsigned int

The cluster ID of this processor.

CORE_NUM unsigned int

The number of this core in an MP processor.

DETAILED_CORE_FEATURES enum

Detailed core features as a bitfield. Enum values are bitmasks to extract the info from the field.

FPU_VERSION enum

The VFP version implemented by the core.

MEM_ARCH enum

The memory architecture of the core.

MIDR_EL1 unsigned int

MIDR_EL1 value.

NUM_CORES unsigned int

The number of cores in this MP processor.

QUANTUM_SIZE unsigned int

The default quantum size of the core (in ticks).

SECURITY_FEATURES bool

Does the core have security features?.

CORE_LOADS

Processor load accesses. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ACQREL enum

Is this an acquire/release.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DATA unsigned int

The data read or written.

ELEMENT_SIZE unsigned int

Width of each element.

ENDIANNESS enum

Byte ordering of this access.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

LOCK enum

Normal, exclusive or locked access.

MECID unsigned int

Memory Encryption Context ID, if applicable.

MECID2 unsigned int

Memory Encryption Context ID of the second page, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PADDR_TAG unsigned int

If tag checked, the physical address tag for the access.

PAGE_SIZE unsigned int

Page size as $\log_2(\text{size in bytes})$.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK, 2=Exclusive Failed, 3=Unsupported.

SIZE unsigned int

Width of the access in bytes. Only required if DATA is not traced.

TAG_CHECKED bool

Is this access tag checked.

TAG_CHECKED2 bool

Is the second page access tag checked.

TRANS bool

Is this a translated access.

VADDR unsigned int

The virtual address of the access.

CORE_REGS

Changes of the core registers R0 to R14. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number, 0 to 14.

MODE enum

Bank of the register accessed.

OLD_VALUE unsigned int

The old value overwritten.

PHYS_ID enum

The physical register accessed.

VALUE unsigned int

The new value written to the register.

CORE_REGS64

Changes of the core registers X0..X30, SP_ELn. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ID enum

The register number, 0-30 for X0-X30, >=32 for SP_ELn.

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

CORE_REGS64_READ

General purpose AArch64 register has been read X0..X30, SP_ELn. Fields:

ID enum

The register number, 0-30 for X0-X30, >=32 for SP_ELn.

VALUE unsigned int

The value read from the register.

CORE_STALLED_TRANSITION

Detect when a core is prevented from progressing PChannel states. Fields:

CURRENT_STATE enum

Current power mode of core.

REASON string

The reason a transition was stalled, and what must be done to unblock it.

TARGET_STATE enum

Desired power mode of core after transition.

CORE_STORES

Processor store accesses. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ACQREL enum

Is this an acquire/release.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DATA unsigned int

The data read or written.

ELEMENT_SIZE unsigned int

Width of each element.

ENDIANNESS enum

Byte ordering of this access.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

LOCK enum

Normal, exclusive or locked access.

MECID unsigned int

Memory Encryption Context ID, if applicable.

MECID2 unsigned int

Memory Encryption Context ID of the second page, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PADDR_TAG unsigned int

If tag checked, the physical address tag for the access.

PAGE_SIZE unsigned int

Page size as $\log_2(\text{size in bytes})$.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK, 2=Exclusive Failed, 3=Unsupported.

SIZE unsigned int

Width of the access in bytes. Only required if DATA is not traced.

TAG_CHECKED bool

Is this access tag checked.

TAG_CHECKED2 bool

Is the second page access tag checked.

TRANS bool

Is this a translated access.

VADDR unsigned int

The virtual address of the access.

CP14_READ

System Coprocessor register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP14 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc1 unsigned int

opcode 1.

opc2 unsigned int

opcode 2.

CP14_WRITE

System Coprocessor register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP14 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc1 unsigned int

opcode 1.

opc2 unsigned int
opcode 2.

CP15_READ

System Coprocessor register read. Fields:

CORE_NUM unsigned int
Core number in a multi processor.

CRm unsigned int
CRm.

CRn unsigned int
CRn.

NS bool
Secure or nonsecure banked register is accessed.

REG_NAME string
Name of the CP15 register accessed.

SECURITY_STATE enum
The current security state.

UNDEF bool
The register accessed is undefined.

VALUE unsigned int
The value read.

opc1 unsigned int
opcode 1.

opc2 unsigned int
opcode 2.

CP15_READ64

System Coprocessor register read. Fields:

CORE_NUM unsigned int
Core number in a multi processor.

CRm unsigned int
CRm.

NS bool
Secure or nonsecure banked register is accessed.

REG_NAME string
Name of the CP15 register accessed.

SECURITY_STATE enum
The current security state.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc unsigned int

opcode 1.

CP15_WRITE

System Control Coprocessor register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP15 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc1 unsigned int

opcode 1.

opc2 unsigned int

opcode 2.

CP15_WRITE64

System Coprocessor register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP15 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc unsigned int

opcode 1.

CPSR

CPSR change. (In AArch64 PSTATE represented in SPSR format). Fields:

CORE_NUM unsigned int

Core number in a multi processor.

OLD_VALUE unsigned int

The old CPSR value.

UNKNOWN unsigned int

Bits within the register that have unknown value.

VALUE unsigned int

The new CPSR value.

CRYPTO_SPEC

Every crypto instruction speculatively executed.

CorePowerStateChange

Core power-up state has changed. DISPLAY Core powered %{PU:(up|down)}. Fields:

PU bool

Whether the core powered up.

DATA_CACHE_SET_ALLOC_TAGS

Set allocation tags from DC GVA instruction. Fields:

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate

when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

NSDESC unsigned int

The security state of the access.

PADDR unsigned int

The physical address of the access.

PAS enum

The physical address space of the page.

RESPONSE enum

0=Aborted, 1=OK.

SIZE unsigned int

The size of memory in bytes.

TAG unsigned int

The value of the tag being set.

TAGGING_TYPE enum

Type of Memory tagging (Physical or Virtual).

TAG_VADDR unsigned int

The virtual address of the Virtual Tag if Virtual tagging is used; otherwise 0.

VADDR unsigned int

Virtual address used for setting allocation tags.

DATA_CACHE_ZERO

Zero and invalidate cache line from DC ZVA instruction. Fields:

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

NSDESC unsigned int

The security state of the access.

PADDR unsigned int

The physical address of the access.

PADDR_TAG unsigned int

If tag checked, the physical address tag for the access.

PAS enum

The physical address space of the page.

RESPONSE enum

0=Aborted, 1=OK.

SIZE unsigned int

The size of zero'd memory in bytes.

TAG_CHECKED bool

Is this access tag checked.

VADDR unsigned int

Virtual address of the zero'd cache line.

DEBUG_EVENT

Hardware debug support event. Fields:

EVENT enum

Description of event.

VALUE unsigned int

data value.

DEBUG_TARGET_ISET

Hardware debug target ISA. Fields:

TARGET_ISET enum

The target instruction set of the debug state.

DMI_ALLOCATE

DMI allocated. Fields:

ACCESS_DIRECTION enum

READ / WRITE.

PADDR unsigned int

The physical address of the DMI.

PTR unsigned int

DMI range pointer.

SIDE enum

DSIDE / ISIDE.

SIZE unsigned int

The size of DMI.

VADDR unsigned int

The virtual address of the DMI..

DMI_HIT

DMI hit, access used DMI (only fire for small DMI regions i.e. less than a full page). Fields:

ACCESS_DIRECTION enum

READ / WRITE.

PADDR unsigned int

The physical address of the DMI.

SIDE enum

DSIDE / ISIDE.

SIZE unsigned int

The size of the DMI.

VADDR unsigned int

The virtual address of the DMI..

DMI_REVOKE

DMI revoked. Fields:

ACCESS_DIRECTION enum

READ / WRITE.

PADDR unsigned int

The physical address of the DMI.

SIDE enum

DSIDE / ISIDE.

SIZE unsigned int

The size of DMI.

VADDR unsigned int

The virtual address of the DMI..

END_COMPILE

Compilation end. Fields:

END_OF_PAGE_COUNT unsigned int

Number of basic blocks exited due to page boundary since START_COMPILE.

FETCHFAIL_COUNT unsigned int

Number of basic blocks exited due to ifetch failure START_COMPILE.

INST_COUNT unsigned int

Number of instructions compiled since START_COMPILE.

NONSEQ_COUNT unsigned int

Number of basic blocks exited due to non-sequential instructions since START_COMPILE.

PAGE_STRADDLE_COUNT unsigned int

Number of basic blocks exited due to unaligned instructions crossing page since START_COMPILE.

ESB

Error Synchronization Barrier hit. Fields:

ASYNC bool

ESB is async.

IMPLICIT bool

ESB is implicit.

EXCEPTION

Exception taken. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ESR unsigned int

The value assigned to ESR when exception is taken.

ESR_EC unsigned int

The Value assigned to ESR.EC when exception is taken.

IS_PHYSICAL bool

Physical or Virtual exception.

IS_TRAP bool

Exception coming from a trap.

LR unsigned int

The value assigned to the link register.

PC unsigned int

The location where the exception occurred.

PREFERRED_RETURN unsigned int

The preferred return address for the exception.

TAKEN_LOCALLY bool

Whether exception is taken locally(same exception level).

TARGET_ISET enum

The instruction set of the exception handler code.

TARGET_PC unsigned int

The address the exception branches to.

VECTOR enum

The exception vector.

VECTOR_OFFSET unsigned int

The offset of an exception vector from the base address.

EXCEPTION_END

Every exception completed.

EXCEPTION_INFO

Trace human-readable reason of the current exception for data/instruction aborts, floating point exceptions and unknown exceptions. Fields:

DESCRIPTION string

Reason for current exception.

ESR_EC unsigned int

The value of ESR.EC when exception is taken.

EXCEPTION_RAISE

Every exception raised.

EXCEPTION_RETURN

Branches on leaving exception. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISET enum

The instruction set of the branch instruction.

IS_COND bool

Indicates if this is a conditional branch.

IS_HINTED bool

Whether the conditional instruction is hinted.

IS_ISB bool

Whether the instruction is ISB.

IS_LINK bool

Indicates if this is a branch with link.

PC unsigned int

The address of the branch instruction.

TARGET_ISET enum

The instruction set after the branch.

TARGET_PC unsigned int

The address the instruction branches to.

EXCEPTION_RETURN_PREBRANCH

Exception return event that must occur before PC and processor state have updated.

EXCEPTION_START

Every exception started.

FIQ_TAKEN

FIQ taken exception.

FP_EXCEPTIONS_SIGNALLED

Trace which FP exceptions were signalled as a part of the FP operations performed by an instruction. Fields:

FP_EXCEPTIONS_SIGNALLED_MASK unsigned int

Indicates which FP exceptions were signalled by an instruction. This uses the same encoding as the cumulative floating-point exception bitfields in FPSR.

FP_STATE

Floating point state. Fields:

FPSR unsigned int

FPSR register value.

MASK unsigned int

The mask indicating updated cumulative bits.

FREQ_CHANGED

Core frequency changed. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The instruction count of this core.

NEW_FREQ signed int

The new frequency of this core (expressed in Hz).

OLD_FREQ signed int

The old frequency of this core (expressed in Hz).

HLT

HLT instruction occurred. Fields:

IMM16 unsigned int

The 16-bit immediate value encoded in HLT instruction.

INST

Every instruction executed. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DEBUG_STATE bool

The instruction is executed in debug state.

DISASS string

Disassembly of instruction.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISSET **enum**

The current instruction set.

ITSTATE **unsigned int**

The current ITSTATE.

LOCAL_TIME **unsigned int**

The core's local time, relative to the current quantum.

MODE **enum**

The mode the core is in.

NS **unsigned int**

The core's non-secure bit.

NSDESC **unsigned int**

The physical address non-secure bit.

NSDESC2 **unsigned int**

The second page physical address non-secure bit.

OPCODE **unsigned int**

The opcode of the instruction.

PADDR **unsigned int**

The physical address of the instruction.

PADDR2 **unsigned int**

If different from PADDR, the physical address of the second page of the instruction.

PAS **enum**

The physical address space of the page.

PAS2 **enum**

The physical address space of the page.

PC **unsigned int**

The address of the instruction.

SECURITY_STATE **enum**

The core's security state.

SIZE **unsigned int**

The size of the instruction in bytes.

INST_END

Every instruction completed.

INST_START

Every instruction started. Fields:

DEBUG_STATE **bool**

The instruction is executed in debug state.

ISSET enum

The current instruction set.

MODE enum

The mode the core is in.

NS enum

The current Secure State.

PC unsigned int

The address of the conditional instruction.

SECURITY_STATE enum

The current Security State.

INST_STRADDLE

Instruction straddles boundary. Fields:

DEBUG_STATE bool

The instruction is executed in debug state.

ISSET enum

The current instruction set.

MODE enum

The mode the core is in.

NS enum

The current Secure State.

PC unsigned int

The address of the conditional instruction.

SECURITY_STATE enum

The current Security State.

IRQ_TAKEN

IRQ taken exception.

ISB

ISB instruction occurred. Fields:

PC unsigned int

The address of the ISB instruction.

LOAD_MULTIPLE_REGS_LIST

List of destination registers (not source) of a multiple registers load instruction. Fields:

REG string

Register to add into the list.

LOCAL_MONITOR

Local monitor activity. Fields:

PADDR unsigned int

Local Monitor Address.

State enum

State of the monitor (Open/Exclusive).

MAPPING_CREATE

A memory mapping has been created. This is an implementation detail of the model and so may change without notice. Fields:

ASID unsigned int

The ASID used during translation, or 0 if no ASID was used.

IS_VALID bool

Whether the mapping is valid or not.

PADDR unsigned int

Base physical address of region, or 0 if not known.

SIDE enum

I-side or D-side.

SIZE unsigned int

Size of the page, or 0 if not known.

VADDR unsigned int

Base virtual address of region.

VMID unsigned int

The VMID used during translation, or 0 if no VMID was used.

MAPPING_DESTROY

A memory mapping has been destroyed. This is an implementation detail of the model and so may change without notice. Fields:

ASID unsigned int

The ASID used during translation.

CAUSE enum

The cause of invalidation.

PADDR unsigned int

Base physical address of page.

SIDE enum

I-side or D-side.

SIZE unsigned int

Size of the page.

VADDR unsigned int

Base virtual address of page.

VMID **unsigned int**

The VMID used during translation.

MEMMAP_DEBUG_READ

Memory mapped reads to the debug registers. Fields:

ADDR **unsigned int**

address.

CORE_NUM **unsigned int**

Core number in a multi processor.

EXT **bool**

Whether access is from an external device (such as the DAP).

NS **enum**

Secure state of the access.

REG_NAME **string**

Name of the debug register accessed.

UNDEF **bool**

The register accessed is undefined.

VALUE **unsigned int**

The value read.

MEMMAP_DEBUG_WRITE

Memory mapped writes to the debug registers. Fields:

ADDR **unsigned int**

address.

CORE_NUM **unsigned int**

Core number in a multi processor.

EXT **bool**

Whether access is from an external device (such as the DAP).

NS **enum**

Secure state of the access.

REG_NAME **string**

Name of the debug register accessed.

UNDEF **bool**

The register accessed is undefined.

VALUE **unsigned int**

The new value written.

MEMTAG_LOADS

Allocation tag memory read; when attributes and system configuration allow tag accesses.

Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

NUM_GRANULES unsigned int

Number of tags to be Loaded/stored.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK.

TAGGING_TYPE enum

Type of Memory tagging (Physical or Virtual).

TAGS unsigned int

The tags read or written. Tags for subsequent granules would be found at subsequent bytes in this field (One byte for each tag).

TAG_VADDR unsigned int

The virtual address of the Virtual Tag if Virtual tagging is used; otherwise 0.

VADDR unsigned int

The virtual address of the access.

MEMTAG_LOAD_INST

Allocation tag load instruction; regardless of tags being actually accessed. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

NUM_GRANULES unsigned int

Number of tags to be Loaded/stored.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK.

TAGGING_TYPE enum

Type of Memory tagging (Physical or Virtual).

TAGS unsigned int

The tags read or written. Tags for subsequent granules would be found at subsequent bytes in this field (One byte for each tag).

TAG_VADDR unsigned int

The virtual address of the Virtual Tag if Virtual tagging is used; otherwise 0.

VADDR unsigned int

The virtual address of the access.

MEMTAG_STORES

Allocation tag memory write; when attributes and system configuration allow tag accesses.

Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

NUM_GRANULES unsigned int

Number of tags to be Loaded/stored.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK.

TAGGING_TYPE enum

Type of Memory tagging (Physical or Virtual).

TAGS unsigned int

The tags read or written. Tags for subsequent granules would be found at subsequent bytes in this field (One byte for each tag).

TAG_VADDR unsigned int

The virtual address of the Virtual Tag if Virtual tagging is used; otherwise 0.

VADDR unsigned int

The virtual address of the access.

MEMTAG_STORE_INST

Allocation tag store instruction; regardless of tags being actually accessed. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

NUM_GRANULES unsigned int

Number of tags to be Loaded/stored.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK.

TAGGING_TYPE enum

Type of Memory tagging (Physical or Virtual).

TAGS unsigned int

The tags read or written. Tags for subsequent granules would be found at subsequent bytes in this field (One byte for each tag).

TAG_VADDR unsigned int

The virtual address of the Virtual Tag if Virtual tagging is used; otherwise 0.

VADDR unsigned int

The virtual address of the access.

MMU_D128_TTB_READ

This event is triggered by reads caused by a D128 translation table walk. Fields:

ABORTED bool

Was the walk successful.

DATA_HIGH unsigned int

The data upper bits read.

DATA_LOW unsigned int

The data lower bits read.

IPA unsigned int

For stage 1, the IPA of the read.

LEVEL unsigned int

Translation table level.

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

OUTADDR unsigned int

The output physical address.

PADDR unsigned int

The physical address of the read.

PASmem enum

Physical Address Space of the accessed memory.

PASreq enum

Physical Address Space of the requested lookup.

REGIME_EL enum

Entry matches in this translation regime.

REQUESTER enum

The requester of this table walk.

SIDE enum

Inst / Data.

STAGE unsigned int

Translation stage.

MMU_D128_TTB_WRITE

This event is triggered by writes caused by a D128 translation table walk. Fields:

ABORTED bool

Was the walk successful.

DATA_HIGH unsigned int

The data upper bits written.

DATA_LOW unsigned int

The data lower bits written.

IPA unsigned int

For stage 1, the IPA of the write.

IS_HACDBS_ACCESS bool

If the access was due to HACDBS_CLEAN.

LEVEL unsigned int

Translation table level.

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

OUTADDR unsigned int

The output physical address.

PADDR unsigned int

The physical address of the write.

PASmem enum

Physical Address Space of the accessed memory.

PASreq enum

Physical Address Space of the requested lookup.

REGIME_EL enum

Entry matches in this translation regime.

SIDE enum

Inst / Data.

STAGE unsigned int

Translation stage.

MMU_TRANS

Address translation information. Fields:

ASID unsigned int

Address space identifier.

CORE_NUM unsigned int

Core number in a multi processor.

Hyp bool

Entry matches in Hyp state only.

INNERCACHE_RA bool

Is the inner cache allocate on read.

INNERCACHE_TRANSIENT bool

Is the inner write-through transient.

INNERCACHE_TYPE enum

Inner Caching scheme (NC/MB/WA).

INNERCACHE_WA bool

Is the inner cache allocate on write.

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MEMTYPE enum

Memory type.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC enum

Is secure side supposed to access secure or nonsecure memory.

OUTERCACHE_RA bool

Is the outer cache allocate on read.

OUTERCACHE_TRANSIENT bool

Is the outer write-through transient.

OUTERCACHE_TYPE enum

Outer Caching scheme (NC/MB/WA).

OUTERCACHE_WA bool

Is the outer cache allocate on write.

PADDR unsigned int

Physical address of the access.

PAGESIZE unsigned int

Page size as $\log_2(\text{size})$.

PAS enum

PAS supposed to be accessed is secure or nonsecure memory or Root/Realm if RME is implemented.

SH enum

Shareability.

SIDE enum

Inst / Data.

STAGE1_PERM unsigned int

Stage 1 permissions, [5:3] Privileged access (XWR) [2:0] User access (XWR).

STAGE2_PERM unsigned int

Stage 2 permission mask [5:3] Privileged access (XWR) [2:0] User access (XWR).

VADDR unsigned int

Virtual address of the access.

VMID unsigned int

Virtual machine identifier.

XS bool

XS Attribute.

nG enum

Flag indicating whether ASID will be matched.

MMU_TTB_READ

This event is triggered by reads caused by a translation table walk. Fields:

ABORTED bool

Was the walk successful.

DATA unsigned int

The data read.

DESC_KIND enum

The encoding scheme of the descriptors used for this walk.

IPA unsigned int

For stage 1, the IPA of the read.

LEVEL unsigned int

Translation table level.

LPAE bool

Is this for an LPAE translation (*DEPRECATED*, use DESC_KIND).

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSmem bool

Non secure memory access.

NSreq bool

Non secure request.

OUTADDR unsigned int

The output physical address.

PADDR unsigned int

The physical address of the read.

PASmem enum

Physical Address Space of the accessed memory.

PASreq enum

Physical Address Space of the requested lookup.

REGIME_EL enum

Entry matches in this translation regime.

REQUESTER enum

The requester of this table walk.

SIDE enum

Inst / Data.

STAGE unsigned int

Translation stage.

MMU_TTB_WRITE

This event is triggered by writes caused by a translation table walk. Fields:

ABORTED bool

Was the walk successful.

DATA unsigned int

The data written.

DESC_KIND enum

The encoding scheme of the descriptors used for this walk.

IPA unsigned int

For stage 1, the IPA of the write.

IS_HACDBS_ACCESS bool

If the access was due to HACDBS_CLEAN.

LEVEL unsigned int

Translation table level.

LPAE bool

Is this for an LPAE translation (*DEPRECATED*, use DESC_KIND).

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSmem bool

Non secure memory access.

NSreq bool

Non secure request.

OUTADDR unsigned int

The output physical address.

PADDR unsigned int

The physical address of the write.

PASmem enum

Physical Address Space of the accessed memory.

PASreq enum

Physical Address Space of the requested lookup.

REGIME_EL enum

Entry matches in this translation regime.

SIDE enum

Inst / Data.

STAGE unsigned int

Translation stage.

MODE_CHANGE

Mode change. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

MODE enum

The new mode.

NON_SECURE enum

The core's new non-secure bit.

OLD_MODE enum

The old mode.

SECURITY_STATE enum

The new security state.

NOT_TAG_CHECKED_INST

MTE Tag not checked for current instruction.

NV2_CORE_128_LOADS

Processor load accesses for nested virtualization of v9.4 128-bit registers. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ACQREL enum

Is this an acquire/release.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DATA unsigned int

The data read or written.

ELEMENT_SIZE unsigned int

Width of each element.

ENDIANNESS enum

Byte ordering of this access.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

LOCK enum

Normal, exclusive or locked access.

MECID unsigned int

Memory Encryption Context ID, if applicable.

MECID2 unsigned int

Memory Encryption Context ID of the second page, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PADDR_TAG unsigned int

If tag checked, the physical address tag for the access.

PAGE_SIZE unsigned int

Page size as $\log_2(\text{size in bytes})$.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

REGNUM_OPERAND unsigned int

Register number corresponding to the instruction operand.

RESPONSE enum

0=Aborted, 1=OK, 2=Exclusive Failed, 3=Unsupported.

SIZE unsigned int

Width of the access in bytes. Only required if DATA is not traced.

TAG_CHECKED bool

Is this access tag checked.

TAG_CHECKED2 bool

Is the second page access tag checked.

TRANS bool

Is this a translated access.

VADDR unsigned int

The virtual address of the access.

NV2_CORE_128_STORES

Processor store accesses for nested virtualization of v9.4 128-bit registers. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ACQREL enum

Is this an acquire/release.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DATA unsigned int

The data read or written.

ELEMENT_SIZE unsigned int

Width of each element.

ENDIANNESS enum

Byte ordering of this access.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

LOCK enum

Normal, exclusive or locked access.

MECID unsigned int

Memory Encryption Context ID, if applicable.

MECID2 unsigned int

Memory Encryption Context ID of the second page, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PADDR_TAG unsigned int

If tag checked, the physical address tag for the access.

PAGE_SIZE unsigned int

Page size as $\log_2(\text{size in bytes})$.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

REGNUM_OPERAND unsigned int

Register number corresponding to the instruction operand.

RESPONSE enum

0=Aborted, 1=OK, 2=Exclusive Failed, 3=Unsupported.

SIZE unsigned int

Width of the access in bytes. Only required if DATA is not traced.

TAG_CHECKED bool

Is this access tag checked.

TAG_CHECKED2 bool

Is the second page access tag checked.

TRANS bool

Is this a translated access.

VADDR unsigned int

The virtual address of the access.

NV2_CORE_LOADS

Processor load accesses for v8.4 nested virtualization. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ACQREL enum

Is this an acquire/release.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DATA unsigned int

The data read or written.

ELEMENT_SIZE unsigned int

Width of each element.

ENDIANNESS enum

Byte ordering of this access.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

LOCK enum

Normal, exclusive or locked access.

MECID unsigned int

Memory Encryption Context ID, if applicable.

MECID2 unsigned int

Memory Encryption Context ID of the second page, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PADDR_TAG unsigned int

If tag checked, the physical address tag for the access.

PAGE_SIZE unsigned int

Page size as $\log_2(\text{size in bytes})$.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

REGNUM_OPERAND unsigned int

Register number corresponding to the instruction operand.

RESPONSE enum

0=Aborted, 1=OK, 2=Exclusive Failed, 3=Unsupported.

SIZE unsigned int

Width of the access in bytes. Only required if DATA is not traced.

TAG_CHECKED bool

Is this access tag checked.

TAG_CHECKED2 bool

Is the second page access tag checked.

TRANS bool

Is this a translated access.

VADDR unsigned int

The virtual address of the access.

NV2_CORE_STORES

Processor store accesses for v8.4 nested virtualization. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ACQREL enum

Is this an acquire/release.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DATA unsigned int

The data read or written.

ELEMENT_SIZE unsigned int

Width of each element.

ENDIANNESS enum

Byte ordering of this access.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

LOCK enum

Normal, exclusive or locked access.

MECID unsigned int

Memory Encryption Context ID, if applicable.

MECID2 unsigned int

Memory Encryption Context ID of the second page, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PADDR_TAG unsigned int

If tag checked, the physical address tag for the access.

PAGE_SIZE unsigned int

Page size as $\log_2(\text{size in bytes})$.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

REGNUM_OPERAND unsigned int

Register number corresponding to the instruction operand.

RESPONSE enum

0=Aborted, 1=OK, 2=Exclusive Failed, 3=Unsupported.

SIZE unsigned int

Width of the access in bytes. Only required if DATA is not traced.

TAG_CHECKED bool

Is this access tag checked.

TAG_CHECKED2 bool

Is the second page access tag checked.

TRANS bool

Is this a translated access.

VADDR unsigned int

The virtual address of the access.

PERIODIC

Called for every quantum. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The instruction count of this CPU.

PC unsigned int

The address of the next instruction to be executed on this CPU.

PMU_COUNTER_OVERFLOW

PMU counter overflow. Fields:

COUNTER_GROUP enum

Counter group.

COUNTER_RANGE enum

Counter range.

EVENT_ID unsigned int

Event Identifier, field is valid only for Event counter group.

INDEX unsigned int

Counter index (as selected by PMSELR), field is valid only for Event/Cycle counter groups.

INTERRUPT bool

Is interrupt enabled on overflow for this counter.

POINTER_AUTH

Called when execution hits a pointer authentication check. Fields:

AUTH_STATE bool

Authentication result.

KEY_USED enum

Pointer authentication key.

POTINDEX_UPDATE

Value of POT index for current pc changed. Fields:

NEW_POTINDEX unsigned int

New POTIndex.

OLD_POTINDEX unsigned int

Old POTIndex.

PREFETCH_MEMORY64

Prefetch from PRFM or PRFUM instructions. Fields:

PRFOP enum

Prefetch hint.

VADDR unsigned int

Virtual address of the location that should be prefetched.

PRELOAD64_REQUEST

Preload requests from PRFM or PRFUM instructions. Fields:

PRFOP enum

Prefetch hint.

VADDR unsigned int

Virtual address of the location that should be prefetched.

PRELOAD_DATA

Data preload from PLD instruction. Fields:

VADDR unsigned int

Virtual address of the data that should be preloaded.

PRELOAD_INST

Instruction preload from PLI instruction. Fields:

VADDR unsigned int

Virtual address of the instruction that should be preloaded.

PRE_CORE_LOAD

Trace just before a core load. Fields:

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

PRE_CORE_STORE

Trace just before a core store. Fields:

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

PRE_TTB_READ

This event is triggered before reads caused by a translation table walk. Fields:

ADDR unsigned int

The physical address of the read.

PSTATE

PSTATE fields update. Fields:

ALLINT unsigned int

All interrupt mask bit.

BTYPE unsigned int

Branch target identification bit.

CORE_NUM unsigned int

Core number in a multi processor.

DAIF unsigned int

Exception mask bits.

DIT unsigned int

Data Independent Timing bit.

EL unsigned int

Current Exception level field.

EXLOCK unsigned int

Exception return state lock bit.

IL unsigned int

Illegal Execution state bit.

NZCV unsigned int

Condition flags.

PACM unsigned int

PACM bit.

PAN unsigned int

Privileged Access Never state bit.

PM unsigned int

PMU exception mask bit.

PPEND unsigned int

PMU exception pending bit.

SM unsigned int

StreamingMode status bit.

SP unsigned int

Stack pointer register selection bit.

SS unsigned int

Software Step bit.

SSBS unsigned int

Speculative Store Bypass Safe bit.

TCO unsigned int

Tag Check Override bit.

TOUCHED unsigned int

Mask of the fields that have been touched since this trace has last been emitted. Note that touched does not necessarily mean the value changed. The index of the field is the bit index in the mask plus the index of this field plus one.

UAO unsigned int

User Access Override bit.

UINJ unsigned int

Undefined Instruction exceptions bit.

ZA unsigned int

ZA storage and ZT0 register enable bit.

nRW unsigned int

Current Execution state bit.

PSTATE_PACTIVE_OPMODE_TRANSITION

Trace pactive and Opmode mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Operating mode after the transition.

OLD_STATE enum

Operating mode before the transition.

PSTATE_PACTIVE_OPMODE_TRANSITION_DENIED

Trace the denied pactive and Opmode mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Operating mode after the transition.

OLD_STATE enum

Operating mode before the transition.

REASON string

Reason for the transition being denied.

PSTATE_PACTIVE_POWER_TRANSITION

Trace pactive and power mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Power mode after the transition.

OLD_STATE enum

Power mode before the transition.

PSTATE_PACTIVE_POWER_TRANSITION_DENIED

Trace the denied pactive and power mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Power mode after the transition.

OLD_STATE enum

Power mode before the transition.

REASON string

Reason for the transition being denied.

PSTATE_PACTIVE_THREAD_OPMODE_TRANSITION

Trace pactive and Opmode mode transitions for threads on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Operating mode after the transition.

OLD_STATE enum

Operating mode before the transition.

QUANTUM_START

Called at the start of every quantum.

RANGE_PRELOAD_REQUEST

Preload requests from RPRFM. Fields:

COUNT unsigned int

Number of blocks of data to access.

LENGTH unsigned int

Number of contiguous bytes to be accessed in each block.

OP unsigned int

Range prefetch operation .

REUSE unsigned int

Maximum number of bytes to be accessed before next RPRFM .

STRIDE signed int

Stride to advance after accessing LENGTH bytes.

VADDR unsigned int

Base virtual address of range prefetch request.

RUN_STATE

Run state transition. Fields:

INST_COUNT unsigned int

Ticks count at point of transition.

NEW enum

New run state.

OLD enum

Old run state.

SBP_SAMPLE_COLLISION

DISPLAY Statistical profiling samples collided .

SBP_SAMPLE_DISCARDED

DISPLAY Operation discarded by statistical profiling unit when configured in discard mode.
Fields:

BUFFER_POINTER unsigned int

Value of buffer pointer.

DATA unsigned int

Data discarded.

SIZE unsigned int

Size of sample, in bytes.

SBP_SAMPLE_END

DISPLAY Statistical profiling of an operation ended.

SBP_SAMPLE_POP

DISPLAY Operation which might be sampled by statistical profiling.

SBP_SAMPLE_RECORDED

DISPLAY Operation sampled by statistical profiling has been written to the buffer at %{BUFFER_ADDRESS} with size %{SIZE} and data %{DATA}. Fields:

BUFFER_ADDRESS unsigned int

Virtual Address of start of buffer write.

DATA unsigned int

Data written to the buffer.

SIZE unsigned int

Size of sample written to buffer, in bytes.

SBP_SAMPLE_START

DISPLAY Statistical profiling of an operation started.

SBP_SAMPLE_TAKEN

DISPLAY Operation sampled by statistical profiling after filtering.

SEMIHOSTING_CALL

Call of a semihost function occurred. Fields:

PC unsigned int

The program counter after the semihosting call.

SEMIHOSTING_PRECALL

About to call semihost. Fields:

EL2 bool

Use EL2 translation regime.

NS bool

Is Non-Secure.

REG_WIDTH unsigned int

The current register width in bytes.

SIGNAL

External signal state change. Fields:

SIGNAL *enum*

Signal that changed.

STATE *bool*

Signal asserted state.

SOFTWARE_STEP

Return the Debug Software Step state. Fields:

STATE *enum*

Software Step state.

SPSR

SPSR change. Fields:

CORE_NUM *unsigned int*

Core number in a multi processor.

MODE *enum*

Which of the banked SPSR registers is written.

OLD_VALUE *unsigned int*

The old SPSR value.

VALUE *unsigned int*

The new SPSR value.

START_COMPILE

Compilation started. Fields:

VADDR *unsigned int*

Instruction where compilation begins.

STG_MULTIPLE_MICRO_OPS_ADDRESS_RANGE

Address range for MTE stores with multiple uops. Fields:

ADDR_END *unsigned int*

Final address.

ADDR_START *unsigned int*

Starting address.

STORE_MULTIPLE_REGS_ADDRESS_RANGE

Address range for stores with multiple registers. Fields:

ADDR_END *unsigned int*

Final address.

ADDR_START *unsigned int*

Starting address.

SUBOPTIMAL_LDST_RETIRED

suboptimal load/store. Fields:

ACCESS_SIZE unsigned int

Log2 of the access width in bytes used for alignment checking (1: half-word, 2: word ...).

OPTIMAL_ALIGN_SELECTOR unsigned int

Selector to decide the optimal alignment size (1: access width, 2: 4-bytes, 3: 8-bytes, ..., 12: 4-kbytes).

VADDR unsigned int

The virtual address of the access.

SVE_INST_SPEC

SVE operations speculatively executed.

SVE_LD_RETIRED

SVE load instructions architecturally executed.

SVE_LOADS

SVE load accesses. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ACQREL enum

Is this an acquire/release.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DATA unsigned int

The data read or written.

ELEMENT_SIZE unsigned int

Width of each element.

ENDIANNESS enum

Byte ordering of this access.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

LOCK enum

Normal, exclusive or locked access.

MECID unsigned int

Memory Encryption Context ID, if applicable.

MECID2 unsigned int

Memory Encryption Context ID of the second page, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PADDR_TAG unsigned int

If tag checked, the physical address tag for the access.

PAGE_SIZE unsigned int

Page size as $\log_2(\text{size in bytes})$.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK, 2=Exclusive Failed, 3=Unsupported.

SIZE unsigned int

Width of the access in bytes. Only required if DATA is not traced.

TAG_CHECKED bool

Is this access tag checked.

TAG_CHECKED2 bool

Is the second page access tag checked.

TRANS bool

Is this a translated access.

VADDR unsigned int

The virtual address of the access.

SVE_MEM_PREFETCH

Scalable Vector Extension memory prefetch. Fields:

CACHE_LEVEL enum

The preferred cache level.

HINT enum

What the memory is expected to be used for.

SIZE unsigned int

The prefetch size in bytes.

TEMPORAL bool

Whether the data is temporal.

VADDR unsigned int

The virtual address of the prefetch.

SVE_REG_READ

Scalable Vector Extension register read. Fields:

ID enum

The register identifier.

SM bool

Whether the PE is in Streaming Mode.

VALUE unsigned int

The read data.

SVE_REG_UPDATE

Scalable Vector Extension register update. Fields:

ID enum

The register identifier.

SM bool

Whether the PE is in Streaming Mode.

VALUE unsigned int

The written data.

SVE_REG_WRITE

Scalable Vector Extension register write. Fields:

ID enum

The register identifier.

SM bool

Whether the PE is in Streaming Mode.

VALUE unsigned int

The written data.

SVE_STORES

SVE store accesses. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ACQREL enum

Is this an acquire/release.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DATA unsigned int

The data read or written.

ELEMENT_SIZE unsigned int

Width of each element.

ENDIANNESS enum

Byte ordering of this access.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

LOCK enum

Normal, exclusive or locked access.

MECID unsigned int

Memory Encryption Context ID, if applicable.

MECID2 unsigned int

Memory Encryption Context ID of the second page, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PADDR_TAG unsigned int

If tag checked, the physical address tag for the access.

PAGE_SIZE unsigned int

Page size as $\log_2(\text{size in bytes})$.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK, 2=Exclusive Failed, 3=Unsupported.

SIZE unsigned int

Width of the access in bytes. Only required if DATA is not traced.

TAG_CHECKED bool

Is this access tag checked.

TAG_CHECKED2 bool

Is the second page access tag checked.

TRANS bool

Is this a translated access.

VADDR unsigned int

The virtual address of the access.

SVE_ST_RETIRED

SVE store instructions architecturally executed.

SVE_UNALIGNED_LDST_RETIRED

SVE load or store instructions architecturally executed that would cause alignment fault if A bit is set.

SVE_Z_PRED_MEM_ACCESS_INFO

Information about SVE/SME predicated accesses to Z registers (non Gather/Scatter). Fields:

MEM_ELEM_SIZE unsigned int

Width of elements accessed in memory (in bytes).

RnW bool

Whether this is a Read not a Write.

VA unsigned int

Virtual base address computed for accesses.

SYNC

Called for every synchronization, that is every quantum break. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The instruction count of this CPU.

LOCAL_QUANTUM unsigned int

The local quantum of this CPU.

LOCAL_TIME unsigned int

The local time of this CPU.

SYNC_REASON enum

Reason for sync.

SYSCALL

System call instruction executed. Fields:

IMM unsigned int

Immediate value of the system call instruction.

TYPE enum

System call type.

VADDR unsigned int

Instruction that caused the system call.

SYSREG_READ128

System Coprocessor 128 bit register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

REGNUM enum

Internal register number.

REGNUM_PHYS enum

Internal register number physically accessed (e.g. in presense of VHE redirection).

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_READ64

System Coprocessor register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REGNUM enum

Internal register number.

REGNUM_OPERAND enum

Register number corresponding to the instruction operand.

REGNUM_PHYS enum

Internal register number physically accessed (e.g. in presense of VHE redirection).

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_UPDATE128

Triggers when the system updates a register. Fields:

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG enum

Register number.

UNKNOWN unsigned int

Bits of the register which became unknown.

VALUE unsigned int

Value written to the register.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_UPDATE32

Triggers when the system updates a register. Fields:

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG enum

Register number.

UNKNOWN unsigned int

Bits of the register which became unknown.

VALUE unsigned int

Value written to the register.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_UPDATE64

Triggers when the system updates a register. Fields:

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG enum

Register number.

UNKNOWN unsigned int

Bits of the register which became unknown.

VALUE unsigned int

Value written to the register.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_WRITE128

System Coprocessor 128 bit register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

REGNUM enum

Internal register number.

REGNUM_PHYS enum

Internal register number physically accessed (e.g. in presense of VHE redirection).

UNDEF bool

The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_WRITE64

System Coprocessor register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REGNUM enum

Internal register number.

REGNUM_OPERAND enum

Register number corresponding to the instruction operand.

REGNUM_PHYS enum

Internal register number physically accessed (e.g. in presense of VHE redirection).

UNDEF bool

The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

TRBU_BUFFER_WRAP

DISPLAY Trace buffer current write pointer wrapped.

TRBU_TRIGGER_EVENT

DISPLAY Trace buffer Trigger Event.

TRBU_WRITE

DISPLAY PETU trace data has been written to the TRBU memory buffer at `%{BUFFER_ADDRESS}` with size `%{SIZE}` and data `%{DATA}`. Fields:

BUFFER_ADDRESS unsigned int

Virtual Address of start of buffer write.

DATA unsigned int

Data written to the buffer.

SIZE unsigned int

Size of sample written to buffer, in bytes.

UNALIGNED_LDST_RETIRED

Processor unaligned load/store. Fields:

ACCESS_SIZE unsigned int

Log2 of the access width in bytes used for alignment checking (1=>half-word, 2->word ...).

NUMBER_OF_BEATS unsigned int

Number of accesses (beats) in this burst (Total size in bytes is `ACCESS_SIZE * NUMBER_OF_BEATS`).

VADDR unsigned int

The virtual address of the access.

VFP_D_REGS

VFP/NEON D 64 bit register write. Fields:

ALIASING enum

Type of register aliasing used.

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a write to the corresponding bit in the register. For example, 0x1 would indicate a write to VALUE[0].

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

VFP_D_REGS_READ

VFP/NEON D 64 bit register read. Fields:

ID unsigned int

The register number.

VALUE unsigned int

The current value of the register.

VFP_Q_REGS

VFP/NEON Q 128 bit register write. Fields:

ALIASING enum

Type of register aliasing used.

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a write to the corresponding byte in the register. For example, 0x1 would indicate a write to VALUE[7:0].

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

VFP_Q_REGS_READ

VFP/NEON Q 128 bit register read. Fields:

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a read to the corresponding byte in the register. For example, 0x1 would indicate a read to VALUE[7:0].

VALUE unsigned int

The current value of the register.

VFP_SYS_REGS

Writes to the VFP/NEON units system registers. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ID enum

Which VFP system register is written.

MASK unsigned int

The mask indicating updated cumulative bits.

OLD_VALUE unsigned int

The register's old value overwritten.

VALUE unsigned int

The new value written to the VFP system register.

VFP_S_REGS

VFP/NEON S 32 bit register write. Fields:

ALIASING enum

Type of register aliasing used.

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a write to the corresponding bit in the register. For example, 0x1 would indicate a write to VALUE[0].

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

VFP_S_REGS_READ

VFP/NEON S 32 bit register read. Fields:

ID unsigned int

The register number.

VALUE unsigned int

The current value of the register.

VIRTUAL_TAG_ALLOCATED

Virtual tag information when allocated. Fields:

TAG_VADDR unsigned int

Virtual tag virtual address.

VADDR unsigned int

Data Virtual address.

VIRTUAL_TAG unsigned int

Virtual tag value.

WAYPOINT

Signals end of basic block execution. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ISET enum

Origin instruction set.

IS_COND bool

Indicates if this is a conditional waypoint.

PC unsigned int

Origin address (or 0 if unavailable).

TAKEN bool

Indicates if this waypoint was taken.

TARGET unsigned int

Destination address.

TARGET_ISET enum

Destination instruction set.

WFE_END

WFE ended. Fields:

INST_COUNT unsigned int

Ticks count when leaving WFE.

WFE_EVENT_REGISTER

WFE event register status: set/clear, reason. Fields:

INST_COUNT unsigned int

Ticks count.

REASON enum

Reason for set/clear. Only REASON==0 (Cleared by WFE) clears the bit, all other reasons set it.

WFE_IGNORED

WFE ignored. Fields:

EVENT bool

This WFE was ignored because the event register was set.

INST_COUNT unsigned int

Ticks count when ignoring WFE.

TRAPPED bool

This WFE was trapped.

WFE_START

WFE entered. Fields:

INST_COUNT unsigned int

Ticks count when entering WFE.

WFI_END

WFI ended. Fields:

INST_COUNT unsigned int

Ticks count when leaving WFI.

WFI_IGNORED

WFI ignored. Fields:

DISABLED bool

This WFI was ignored because WFI is disabled.

INST_COUNT unsigned int

Ticks count when ignoring WFI.

REASON enum

specifies reason why WFI trace was ignored.

TRAPPED bool

This WFI was trapped.

WFI_START

WFI entered. Fields:

INST_COUNT unsigned int

Ticks count when entering WFI.

WFI_WAKEUP

WFI wakeup. Fields:

INST_COUNT unsigned int

Ticks count when WFI wakeup occurred.

REASON enum

Reason for wakeup.

2.66 ARM_Neoverse-N3

This section describes the trace sources.

AA64_ASE_SVE_REGS

Modification of AArch64 Advanced SIMD or SVE Vector Registers (via SIMD or Scalar operations), SVE Predicate Registers and ZT0. Fields:

ID **enum**

The register identifier.

MASK **unsigned int**

Bitmask of the register to signal the modified bits in the VALUE field.

SM **bool**

Whether the PE is in Streaming Mode.

VALUE **unsigned int**

Value of AArch64 vector register.

ASYNC_MEMORY_FAULT

Context ID Register write. Fields:

FAULT **unsigned int**

Fault status present in ESR.ISS encoding for Serror in ARCH64 and present in DFSR encoding for external abort format in ARCH32.

PADDR **unsigned int**

Physical Address (or 0 if unavailable).

VADDR **signed int**

Virtual Address (or 0 if unavailable).

ATOMIC_END_ACCESS

Bus trace access for atomics. Fields:

ACCESS_FAIL **bool**

Memory access failed.

ADDR **unsigned int**

The virtual address of the access.

ATTR **unsigned int**

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

COMPARE_VALUE **unsigned int**

Compare value for CAS.

LOAD_VALUE unsigned int

Loaded values.

NSDESC unsigned int

The security state of the access.

OPERAND_VALUE unsigned int

Operation's operand.

OPERATION enum

Operation type.

PADDR unsigned int

The physical address of the access.

PRIV bool

Is this a privileged access?.

ATOMIC_START_ACCESS

Bus trace access for atomics. Fields:

ADDR unsigned int

The virtual address of the access.

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

COMPARE_VALUE unsigned int

Compare value for CAS.

NSDESC unsigned int

The security state of the access.

OPERAND_VALUE unsigned int

Operation's operand.

OPERATION enum

Operation type.

PADDR unsigned int

The physical address of the access.

PRIV bool

Is this a privileged access?.

ArchMsg.Error.exit_code

DISPLAY exit code: %{EXITCODE} component kind: %{KIND} reason: %{REASON}. Fields:

EXITCODE signed int

Exit code to be returned.

KIND string

Component kind that invoked the exit code trace.

REASON string

Reason why the exit code trace was invoked (optional).

ArchMsg.Warning.IgnoredWarmResetRequest

Event fires when software attempts to request warm reset through write to deprecated DBGPRCR.CWRR.

ArchMsg.Warning.RAS_ErrselrUnpredictableState

DISPLAY ERRSELR is in unpredictable state due to value greater than number_of_error_records written to it.

ArchMsg.Warning.RAS_UnpredictableStateWrite

DISPLAY %{REG_NAME} write ignored due to unpredictable ERRSELR state. Fields:

REG_NAME string

Name of register being written.

ArchMsg.Warning.ReportedRevisionRegisterRead

DISPLAY Reported revision and patch do not match that of the cpu. Fields:

ORIGINAL_VALUE unsigned int

The value the register would have based on the modelled revision.

REGISTER string

The name of the register that has ben read.

REPORTED_VALUE unsigned int

The value the register has been overridden to.

ArchMsg.Warning.Unpredictably Indexed PM Event Register

TAGS UNPREDICTABLE DISPLAY Event %{IsDirect:(Selector value|Register)} %{SEL} is >= %{N}. Fields:

IsDirect unsigned int

Direct Access rather than indirect.

N unsigned int

Number of Accessible Registers.

SEL unsigned int

Selector.

ArchMsg.Warning.cache_contents_unknown

A cache has been activated without being invalidated and may contain **UNKNOWN** content. DISPLAY %{SIDE:(D|I)}-Cache was enabled in %{REGIME:(SECURE|NON_SECURE|HYP)} regime but was not invalidated since power-on: cache lines could contain **UNKNOWN** content %{SIDE:(Note: D-side caches must invalidate SECURE lines even if the cache is only used by NON_SECURE code in order to ensure that unexpected dirty lines tagged SECURE are not naturally evicted)}. Fields:

REGIME unsigned int

Current translation regime.

SIDE unsigned int

0 d-side cache or unified cache, 1 i-side.

ArchMsg.Warning.dcimvac_matches_watchpoint

TAGS **UNPREDICTABLE** DISPLAY Watchpoints matching an AArch32 DCIMVA are implementation defined.

ArchMsg.Warning.decode_fieldmismatch

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with different fields %{FIELD1} and %{FIELD2}. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_fracbitsnegative

TAGS **UNPREDICTABLE** DISPLAY VCVT Instruction is unpredictable with a negative number of fraction bits.

ArchMsg.Warning.decode_initblock

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable in an IT block.

ArchMsg.Warning.decode_initblocknotlast

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable in an IT block when not the last.

ArchMsg.Warning.decode_invalid_condition

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable when the condition is not AL.

ArchMsg.Warning.decode_invalidfieldcombination

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with the values in %{FIELD1} and %{FIELD2}. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_invalidvalue

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with the value in %{FIELD}. Fields:

FIELD enum

Field name.

ArchMsg.Warning.decode_mem_hint_unallocated

TAGS **UNPREDICTABLE** DISPLAY There is no memory hint allocated to this bit pattern.

ArchMsg.Warning.decode_registermatch

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with %{FIELD1} and %{FIELD2} as the same register. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_registermismatch

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with %{FIELD1} and %{FIELD2} as different registers. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_registeroutofrange

TAGS **UNPREDICTABLE** DISPLAY Floating-point instruction is unpredictable indexing beyond the end of the register bank.

ArchMsg.Warning.decode_sbzsbo

TAGS **UNPREDICTABLE** DISPLAY Reserved bits in the instruction are not canonical.

ArchMsg.Warning.decode_transactiontoobig

TAGS **UNPREDICTABLE** DISPLAY Attempt to perform a memory transaction of more than 128 bytes.

ArchMsg.Warning.decode_unpred_other

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable.

ArchMsg.Warning.decode_unpreduseofpc

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable when %{FIELD} is PC. Fields:

FIELD enum

Field name.

ArchMsg.Warning.decode_unpreduseofr13

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable when %{FIELD} is R13. Fields:

FIELD enum

Field name.

ArchMsg.Warning.decode_unpreduseofr13inreglist

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with R13 in register-list.

ArchMsg.Warning.decode_useofd16orhigher

TAGS **UNPREDICTABLE** DISPLAY Attempt to use register D16 or higher where it is not allowed.

ArchMsg.Warning.decode_writebackandbaseinlist

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with write back when the base register is in the transfer list.

ArchMsg.Warning.decode_zeroregistersinlist

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable when no registers are to be transferred.

ArchMsg.Warning.misaligned_ptrs

DISPLAY TRBPTR_EL1(%{TRBPTR_EL1}) or TRBTRG_EL1(%{TRBTRG_EL1}) are not aligned as per TRBIDR_EL1.Align(%{TRBIDR_EL1}). Fields:

TRBIDR_EL1 unsigned int

TRBIDR_EL1.Align value.

TRBPTR_EL1 unsigned int

Current write pointer.

TRBTRG_EL1 unsigned int

Trigger pointer.

ArchMsg.Warning.out_of_range_ptrs

DISPLAY TRBPTR_EL1(%{TRBPTR_EL1}) is out of range as per TRBBASER_EL1(%{TRBBASER_EL1}) or TRBLIMITR_EL1.LIMIT(%{TRBLIMITR_EL1}). Fields:

TRBBASER_EL1 unsigned int

Base pointer.

TRBLIMITR_EL1 unsigned int

Limit pointer.

TRBPTR_EL1 unsigned int

Current write pointer.

ArchMsg.Warning.recursive_branch

An instruction is performing a branch that targets the same instruction. If this was intended then a WFI might be more effective DISPLAY recursive branch detected at PC=%{PC}.

PRIMARY KEY PC. Fields:

PC unsigned int

Address of instruction causing the branch.

ArchMsg.Warning.recursive_exception

An instruction has generated an exception that targets the same instruction. DISPLAY recursive %{TYPE} exception detected at PC=%{PC} (CPSR=%{CPSR}). PRIMARY KEY PC. Fields:

CPSR unsigned int

Processor state.

PC unsigned int

Address of instruction causing the exception.

TYPE enum

Type of exception.

ArchMsg.Warning.reentrant_vector_catch

TAGS **UNPREDICTABLE** DISPLAY Vector catch debug event on Prefetch or Data abort vector %{ADDR}. Fields:

ADDR unsigned int

vector address.

ArchMsg.Warning.reserved_it_state

TAGS **UNPREDICTABLE** DISPLAY Execution with reserved IT state %{ITSTATE}. Fields:

ITSTATE unsigned int

reserved IT state.

ArchMsg.Warning.tlb_contents_unknown

Translation has been enabled without invalidating the TLB. DISPLAY Translation was enabled in the %{PAS:(Secure|Non-secure|Root|Realm)} EL%{EL:d} translation regime (vmid=%{VMID:x}) without invalidating %{INVALIDITY:(|d-side |i-side |)}entries since power-on: %{INVALIDITY:(|D||)}TLB could contain **UNKNOWN** content. Fields:

EL unsigned int

Exception level of translation regime.

INVALIDITY unsigned int

Which TLB may be invalid (if TLBs are separate).

NON_SECURE bool

Whether the translation regime is non-secure.

PAS unsigned int

Physical address space of translation regime.

VMID unsigned int

Current VMID, if applicable.

ArchMsg.Warning.unknown_DLR_DSPSR

Exiting debug state with unknown DLR or DSPSR.

ArchMsg.Warning.unknown_ELR_SPSR

Returning from debug exception with unknown ELR or SPSR.

ArchMsg.Warning.unpred_FAR_on_LRCPC3_non_atomic_mem_access

TAGS **UNPREDICTABLE** Unpredictable FAR/FSTATUS on LRCPC3 pair instructions resulting in two separate memory accessesDISPLAY Unpredictable FAR on LRCPC3 non atomic memory access at %{ADDR1} with FSTATUS %{FSTATUS1} or %{ADDR2} with FSTATUS %{FSTATUS2}. Fields:

ADDR1 unsigned int

Possible FAR address 1.

ADDR2 unsigned int

Possible FAR address 2.

FSTATUS1 unsigned int

Possible fault status value 1.

FSTATUS2 unsigned int

Possible fault status value 2.

ArchMsg.Warning.unpred_async_tag_check_fail_on_sve_speculative_load_do_update_ffr

Unpredictable set to FALSE of the FFR element associated to SVE speculative load with an async tag check fail TAGS **UNPREDICTABLE**.

ArchMsg.Warning.unpredictable_a32_breakpoint

TAGS **UNPREDICTABLE** DISPLAY Unpredictable breakpoint on a A32 instruction at %{ADDR1} Offset %{OFFSET} BAS %{BAS} for a %{IS_ADDRESS_MISMATCH_BREAKPOINT:(Mismatch|Match)} Breakpoint number %{N:d} Breakpoint address %{ADDR2}. Fields:

ADDR1 unsigned int

Address.

ADDR2 unsigned int

Breakpoint address.

BAS unsigned int

BAS field.

IS_ADDRESS_MISMATCH_BREAKPOINT bool

Is a Mismatch breakpoint, not a Match breakpoint.

N unsigned int

Breakpoint number.

OFFSET unsigned int

Always has a value of zero.

ArchMsg.Warning.unpredictable_banked_register_access

TAGS **UNPREDICTABLE** DISPLAY Access to banked register (R=%{R:d} SYSm=%{SYSM:2x}) is unpredictable. Fields:

R unsigned int

R bit.

SYSM unsigned int

SYSm.

ArchMsg.Warning.unpredictable_ccfail_undef

TAGS **UNPREDICTABLE** DISPLAY An instruction %{OPCODE} at VA=%{VADDR} failed its condition codes check but would otherwise have caused a non-data-dependent trap or undefined exception. Behaviour is implementation-defined choice of exception or nop. Fields:

OPCODE unsigned int

opcode of the instruction.

VADDR unsigned int

Virtual address of the instruction being executed.

ArchMsg.Warning.unpredictable_ici_bits

TAGS **UNPREDICTABLE** DISPLAY ICI bits do not apply or are architecturally unpredictable for instruction.

ArchMsg.Warning.unpredictable_t32_breakpoint

TAGS **UNPREDICTABLE** DISPLAY Unpredictable breakpoint on a T32 instruction at %{ADDR1} Offset %{OFFSET} BAS %{BAS} for a %{IS_ADDRESS_MISMATCH_BREAKPOINT:(Mismatch|Match)} Breakpoint number %{N:d} Breakpoint address %{ADDR2}. Fields:

ADDR1 unsigned int

Address.

ADDR2 unsigned int

Breakpoint address.

BAS unsigned int

BAS field.

IS_ADDRESS_MISMATCH_BREAKPOINT bool

Is a Mismatch breakpoint, not a Match breakpoint.

N unsigned int

Breakpoint number.

OFFSET unsigned int

Indicates which halfword of a 32-bit Thumb instruction the breakpoint is set on. 0 for the first halfword or if the instruction is 16 bits wide, or 1 for the second halfword.

ArchMsg.Warning.unpredictable_t32_vector_catch

TAGS **UNPREDICTABLE** DISPLAY Unpredictable Vector catch exception on a T32 instruction at %{ADDR} Offset %{OFFSET}. Fields:

ADDR unsigned int

Address.

OFFSET unsigned int

Indicates which halfword of a 32-bit Thumb instruction the vector catch exception is set on. 0 for the first halfword or if the instruction is 16 bits wide, or 1 for the second halfword.

ArchMsg.Warning.unpredictable_watchpoint_far

TAGS **UNPREDICTABLE** DISPLAY The value for FAR/EDWAR for hitting this watchpoint may be anywhere in the range %{LOWER_BOUND} to %{UPPER_BOUND}. Fields:

LOWER_BOUND unsigned int

the lowest address accessed by the instruction that triggered the watchpoint.

UPPER_BOUND unsigned int

the highest watchpointed address accessed by that instruction.

ArchMsg.Warning.unpredictable_watchpoint_far_isscmp

DISPLAY The value for FAR/EDWAR for hitting this watchpoint may be anywhere in the range %{LOWER_BOUND} to %{UPPER_BOUND}. Fields:

LOWER_BOUND unsigned int

the lowest address accessed by the instruction that triggered the watchpoint.

UPPER_BOUND unsigned int

the highest watchpointed address accessed by that instruction.

ArchMsg.Warning.warning_AdvSIMDEExpandImmUnexpectedZero

TAGS UNPREDICTABLE DISPLAY AdvSIMDEExpandImm may treat this immediate value as UNPREDICTABLE.

ArchMsg.Warning.warning_ConditionalSMC

TAGS UNPREDICTABLE DISPLAY SMC instruction has UNPREDICTABLE effects when conditional and when combined with traps.

ArchMsg.Warning.warning_TLBID_out_of_range

TLBID out of range in TLBI according to the configuration of NIS, NVIS, NOS and NVOS. Fields:

NIS unsigned int

NIS.

NOS unsigned int

NOS.

NVIS unsigned int

NVIS.

NVOS unsigned int

NVOS.

TLBID unsigned int

The TLBID value used in the TLBI instruction.

ArchMsg.Warning.warning_TTL_or_TG_mismatches_current_translation_granule

TTL or TG mismatches current translation granule in TLBI VA/RVA/IPA/RIPA. Fields:

CURRENT_TG unsigned int

Current TG as indicated by TCR_ELx or VTCR_EL2.

TLBI_TG unsigned int

TG (or TTL[3:2]) used in TLBI instruction.

ArchMsg.Warning.warning_access_crosses_page_boundary

TAGS UNPREDICTABLE DISPLAY Access crosses page boundary.

ArchMsg.Warning.warning_access_crosses_page_boundary_has_fault

TAGS UNPREDICTABLE DISPLAY Access that crosses page boundary has fault(s).

ArchMsg.Warning.warning_access_crosses_page_boundary_has_fault_on_both_pages

TAGS **UNPREDICTABLE** DISPLAY Access that crosses page boundary has a fault on each page.

Fields:

ADDR1 unsigned int

FAR address on first page.

ADDR2 unsigned int

FAR address on second page.

FSTATUS1 unsigned int

Fault status on first page.

FSTATUS2 unsigned int

Fault status on second page.

ArchMsg.Warning.warning_access_crosses_page_boundary_spanning_different_memory

TAGS **UNPREDICTABLE** DISPLAY Access at address %{ADDR} crosses page from MEM_TYPE %{MEMTYPE_PAGE1} to %{MEMTYPE_PAGE2}. Fields:

ADDR unsigned int

address of access crossing page.

MEMTYPE_PAGE1 signed int

First Page's memory type (0-Normal, 1-Non-Normal).

MEMTYPE_PAGE2 signed int

Second Page's memory type (0-Normal, 1-Non-Normal).

ArchMsg.Warning.warning_access_wraps_around_memory

TAGS **UNPREDICTABLE** DISPLAY Access wraps around memory in %{WIDTH:d}bit mode.

Fields:

WIDTH unsigned int

Address width.

ArchMsg.Warning.warning_bcr_linking_status

TAGS **UNPREDICTABLE** DISPLAY DBGBCR.{BT2,BT} combination gives unpredictable behaviour.

ArchMsg.Warning.warning_bcr_mask_reserved

TAGS **UNPREDICTABLE** DISPLAY DBGBCR_ELn.MASK is set to a reserved value.

ArchMsg.Warning.warning_bvr_masked_bit_not_zero

TAGS **UNPREDICTABLE** DISPLAY DBGBVRn_EL1.VA contains a bit masked by DBGBCRn_EL1.MASK that is not zero.

ArchMsg.Warning.warning_ccsidr_unimplemented_level

TAGS **UNPREDICTABLE** DISPLAY CCSIDR read while CSSELR %{CSSELR:x} points to an unimplemented cache level. Fields:

CSSELR unsigned int

current effective value of CSSELR.

ArchMsg.Warning.warning_change_to_ns_when_tge_set

TAGS **UNPREDICTABLE** DISPLAY Attempting to change NS to 1 when HCR.TGE = 1.

ArchMsg.Warning.warning_contiguous_bit_check_abort

TAGS **UNPREDICTABLE** DISPLAY An abort occurred whilst attempting to check TLB entry is contiguous at %{ENTRY_ADDR}. Fields:

ENTRY_ADDR unsigned int

address of conflicting TLB entry.

ArchMsg.Warning.warning_contiguous_bit_error

TAGS **UNPREDICTABLE** DISPLAY TLB entry at %{ENTRY_ADDR} was not contiguous with entry at %{CONTIG_ADDR} Contents are %{ENTRY_DATA} but expected %{CONTIG_DATA}. Fields:

CONTIG_ADDR unsigned int

address of the TLB first read, with which this entry is expected to be contiguous.

CONTIG_DATA unsigned int

expected contents based on the entry at CONTIG_ADDR.

ENTRY_ADDR unsigned int

address of conflicting TLB entry.

ENTRY_DATA unsigned int

contents of conflicting TLB entry.

ArchMsg.Warning.warning_contiguous_bit_error_D128

TAGS **UNPREDICTABLE** DISPLAY TLB entry at %{ENTRY_ADDR} was not contiguous with entry at %{CONTIG_ADDR} Contents are %{ENTRY_DATA_HI} %{ENTRY_DATA_LO} but expected %{CONTIG_DATA}. Fields:

CONTIG_ADDR unsigned int

address of the TLB first read, with which this entry is expected to be contiguous.

CONTIG_DATA unsigned int

expected contents based on the entry at CONTIG_ADDR.

ENTRY_ADDR unsigned int

address of conflicting TLB entry.

ENTRY_DATA_HI unsigned int

Higher bit contents of conflicting TLB entry.

ENTRY_DATA_LO unsigned int

Lower bit contents of conflicting TLB entry.

ArchMsg.Warning.warning_cp10_cp11_mismatch

TAGS **UNPREDICTABLE** DISPLAY Unpredictable configuration of CP10 and CP11 controls at EL %{EL:d}. Fields:

EL unsigned int

EL owning the control.

ArchMsg.Warning.warning_cp10_cp11_reserved_value

TAGS **UNPREDICTABLE** DISPLAY Unpredictable configuration of CPACR CP10 ({CP10}) or CP11 ({CP11}). Fields:

CP10 unsigned int

CP10 access.

CP11 unsigned int

CP11 access.

ArchMsg.Warning.warning_csselr_level_out_of_range

TAGS **UNPREDICTABLE** DISPLAY CSSELR.Level out of range, written value {LEVEL} with only {IMPLEMENTED:d} levels implemented. Fields:

IMPLEMENTED unsigned int

number of cache levels implemented.

LEVEL unsigned int

written value of CSSELR.Level.

TYPE enum

The type of cache selected by CSSELR.InD and TnD.

ArchMsg.Warning.warning_debug_flow_control_bits_not_obeyed

TAGS **UNPREDICTABLE** DISPLAY Flow control bits for {REGISTER} were not obeyed, causing {ERROR}. Fields:

ERROR enum

type of error resulting.

REGISTER enum

register under consideration.

ArchMsg.Warning.warning_debug_register_access_during_reset

TAGS **UNPREDICTABLE** DISPLAY Unpredictable {IS_WRITE:(read|write)} access to debug register offset {OFFSET} during reset. Fields:

IS_WRITE unsigned int

Write Not Read.

OFFSET unsigned int

Register Offset.

ArchMsg.Warning.warning_decode_cps_inconsistent_fields

TAGS **UNPREDICTABLE** DISPLAY The mode setting fields in the CPS instruction are inconsistent.

ArchMsg.Warning.warning_decode_invalid_state

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable in current exception-level/security state.

ArchMsg.Warning.warning_default_cacheable_mmu_on

TAGS **UNPREDICTABLE** DISPLAY Default cacheable is enabled (HCR.DC = 1) while MMU is enabled.

ArchMsg.Warning.warning_default_cacheable_vmmu_off

TAGS **UNPREDICTABLE** DISPLAY Default cacheable is enabled (HCR.DC = 1) while HCR.VM = 0.

ArchMsg.Warning.warning_deprecated_wvr_bit_2_set

TAGS **UNPREDICTABLE** DISPLAY DBGWVR%{N:d}_EL1=%{WVR:h} has bit 2 set, deprecated in ARMv8. Fields:

N unsigned int

index.

WVR unsigned int

DBGWVR.

ArchMsg.Warning.warning_eret_while_software_step_active_pending

Missing ISB between setting MDSCR_EL1.SS and ERET TAGS **UNPREDICTABLE**.

ArchMsg.Warning.warning_exclusive_to_non_normal

TAGS **UNPREDICTABLE** DISPLAY Exclusive Access to Non-Normal Memory %{ADDR}. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_exclusive_to_non_writeback

TAGS **UNPREDICTABLE** DISPLAY Exclusive Access to Non-Writeback-Cacheable Memory %{ADDR}. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_execute_from_device_memory

TAGS **UNPREDICTABLE** DISPLAY An attempt was made to execute from Device memory.

ArchMsg.Warning.warning_illegal_cpsr_mode

TAGS **UNPREDICTABLE** DISPLAY Writing an illegal or unimplement mode (%{NEW_MODE:h}) to CPSR was unpredictable. Fields:

NEW_MODE unsigned int

New value of CPSR.M.

ArchMsg.Warning.warning_illegal_srs_mode

TAGS **UNPREDICTABLE** DISPLAY Illegal or **UNPREDICTABLE** mode (%{MODE:h}) used for SRS instruction. Fields:

MODE unsigned int

mode for Banked SP.

ArchMsg.Warning.warning_implementation_defined_read_debug_register_in_SCS

Read Access to Debug registers from the processor is IMPLEMENTATION DEFINED TAGS IMP_DEF. Fields:

PPB_OFFSET unsigned int
debug register offset.

ArchMsg.Warning.warning_implementation_defined_sequential_security_transitions_supported

The behavior of sequential instruction fetches that cross from non-secure to secure memory and contain an SG instruction is CONSTRAINED UNPREDICTABLE TAGS UNPREDICTABLE.

ArchMsg.Warning.warning_implementation_defined_stack_limit_check_supported

It is IMPLEMENTATION DEFINED whether stack pointer limit checking is performed for this instruction TAGS IMP_DEF. Fields:

ADDRESS unsigned int
Address of instruction.

OPCODE unsigned int
instruction opcode.

ArchMsg.Warning.warning_implementation_defined_taggednonsharedmemorytype

TAGS IMP_DEF DISPLAY Memory attribute has Tagged, non shared memory type; it is IMPLEMENTATION DEFINED whether the memory location is treated as Tagged or non-Tagged.

ArchMsg.Warning.warning_implementation_defined_write_debug_register_in_SCS

Write Access to Debug registers from the processor is IMPLEMENTATION DEFINED TAGS IMP_DEF. Fields:

DATA unsigned int
data attempted to be written.

PPB_OFFSET unsigned int
debug register offset.

ArchMsg.Warning.warning_invalid_tcr_granule

TAGS UNPREDICTABLE DISPLAY TCR.TG%{TG_ID:(0|1)} bits have been set to a granule size %{{REQUEST:d}}K not implemented - using %{{SUBSTITUTE:d}}K. Fields:

REQUEST signed int
page size requested (or 0 for reserved).

SUBSTITUTE signed int
best guess available page size.

TG_ID bool
bits TG0 or TG1.

ArchMsg.Warning.warning_load_multiple_user_registers_from_user_mode

TAGS UNPREDICTABLE DISPLAY An LDM(user registers) instruction executed from user mode.

ArchMsg.Warning.warning_mte_in_mem_attr_not_supported

TAGS UNPREDICTABLE DISPLAY Memory attribute %{ATTR} has Tagged memory type but Armv8.5-MTE is not implemented. Fields:

ATTR unsigned int

Memory attribute.

ArchMsg.Warning.warning_reserved_breakpoint_state_match

TAGS UNPREDICTABLE DISPLAY %{IS_BREAKPOINT:(Watchpoint|Breakpoint)} programmed with a reserved combination of HMC, SSC and %{IS_BREAKPOINT:(PAC|PMC)}. Fields:

IS_BREAKPOINT unsigned int

Is a breakpoint, not a watchpoint.

ArchMsg.Warning.warning_shareability

TAGS UNPREDICTABLE DISPLAY Unpredictable: combination of the 1st and 2nd stages of translation is Normal Inner Non-Cacheable, Outer Non-Cacheable.

ArchMsg.Warning.warning_software_step_set_while_enabled

TAGS UNPREDICTABLE DISPLAY MDSCR_EL1.SS set to 1 while software step debug exceptions are enabled.

ArchMsg.Warning.warning_thumb_instruction_wraps_around_memory

TAGS UNPREDICTABLE DISPLAY Thumb instruction wraps around memory.

ArchMsg.Warning.warning_ttbr_sbz_bits_are_not_zero

TAGS UNPREDICTABLE DISPLAY %{TTBR}[%{MSB:d};0] should be zero, but are not. Fields:

MSB unsigned int

x-1.

TTBR enum

which TTBR register.

ArchMsg.Warning.warning_unaligned_address_dbgdtrrx_write

TAGS UNPREDICTABLE DISPLAY Unaligned address %{ADDR} in X0/R0 when DBGDTRRX written in memory access mode. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unaligned_address_dbgdtrtx_read

TAGS UNPREDICTABLE DISPLAY Unaligned address %{ADDR} in X0/R0 when DBGDTRTX read in memory access mode. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unaligned_to_device

TAGS UNPREDICTABLE DISPLAY Unaligned Access to Device Memory %{ADDR}. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unaligned_to_strongly_ordered

TAGS **UNPREDICTABLE** DISPLAY Unaligned Access to Strongly Ordered Memory %{ADDR}.

Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unknown_sau_rnr

TAGS **UNPREDICTABLE** DISPLAY SAU_RNR was set to an unsupported value.

ArchMsg.Warning.warning_unpredictable_AIRCR_PRIS_and_BFHFNMINs

TAGS **UNPREDICTABLE** DISPLAY The effect of setting both AIRCR.BFHFNMINs and AIRCR.PRIS to 1 is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_AIRCR_VECTCLRACTIVE_when_not_in_debug

TAGS **UNPREDICTABLE** DISPLAY The effect of writing a 1 to AIRCR.VECTCLRACTIVE if the processor is not halted in Debug state is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_AIRCR_VECTRESET_and_SYSRESETREQ

TAGS **UNPREDICTABLE** DISPLAY When the processor is halted in Debug state, if a write to the register writes a 1 to both VECTRESET and SYSRESETREQ, the behavior is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_AIRCR_VECTRESET_when_not_in_debug

TAGS **UNPREDICTABLE** DISPLAY The effect of writing a 1 to AIRCR.VECTRESET if the processor is not halted in Debug state is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_AIRCR_incorrect_VKEY

TAGS **UNPREDICTABLE** DISPLAY The value 0x05FA must be written to AIRCR.VKEY, otherwise the register write is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_EXC_RETURN_Reserved_Bit

EXC_RETURN[23:7] are reserved with the special condition that all bits should be written as one. Values other than all 1s are **UNPREDICTABLE**. TAGS **UNPREDICTABLE**. Fields:

EXC_RETURN unsigned int

Special PC value.

SBO_MASK unsigned int

Which bits Should Be One to not UNPRED.

ArchMsg.Warning.warning_unpredictable_PMEVTYPEPn_ELO_VS_reserved_value

TAGS **UNPREDICTABLE** DISPLAY Unpredictable reserved value of PMEVTYPEn<n>_ELO.VS.

ArchMsg.Warning.warning_unpredictable_change_to_DEMCR_MON_STEP_at_insufficient_priority

TAGS **UNPREDICTABLE** DISPLAY The effect of changing DEMCR.MON_STEP at an execution priority that is lower than the priority of the DebugMonitor exception is **CONSTRAINED UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_change_to_priority_of_active_exception

TAGS UNPREDICTABLE DISPLAY Changing the priority of an active exception is UNPREDICTABLE.

ArchMsg.Warning.warning_unpredictable_clear_lspact

TAGS UNPREDICTABLE DISPLAY UNPREDICTABLE state on context switch. Fields:

DRVEXC_TAKEN unsigned int

Derived exception taken, not pended.

REACHED_LAST_STORE unsigned int

Only the last faulted.

ArchMsg.Warning.warning_unpredictable_dbg_exit_unaligned_dlr

TAGS UNPREDICTABLE DISPLAY Debug exit to AArch32 with DLR[0] set to 1. PC[0] can be set to 0 or to DLR[0].

ArchMsg.Warning.warning_unpredictable_exception_catch

TAGS UNPREDICTABLE DISPLAY Generation of an exception catch event is unpredictable.

ArchMsg.Warning.warning_unpredictable_exception_return_inconsistent_state

TAGS UNPREDICTABLE DISPLAY state on exception return is unpredictable.

ArchMsg.Warning.warning_unpredictable_exception_return_instruction

TAGS UNPREDICTABLE DISPLAY exception return instruction is unpredictable.

ArchMsg.Warning.warning_unpredictable_in_debug_state

TAGS UNPREDICTABLE DISPLAY Instruction is UNPREDICTABLE when executed in debug state.

ArchMsg.Warning.warning_unpredictable_mair_encoding

TAGS UNPREDICTABLE DISPLAY Memory attribute %{ATTR} has invalid MAIR encoding; this may lead to unpredictable results. Fields:

ATTR unsigned int

Memory attribute.

ArchMsg.Warning.warning_unpredictable_pmu_counter_access

TAGS UNPREDICTABLE DISPLAY Access to PMU counters from non-secure EL0 or EL1 is UNPREDICTABLE with MDCR_EL2.HPMN set to 0.

ArchMsg.Warning.warning_unpredictable_prioritization_breakpoint_match_vector_catch

TAGS UNPREDICTABLE DISPLAY Two events with the same priority occurred at the same instruction: Address Matching Vector Catch debug event, and Breakpoint debug event. It is constrained unpredictable which is taken.

ArchMsg.Warning.warning_unpredictable_stack_selection

TAGS UNPREDICTABLE DISPLAY UNPREDICTABLE state on context switch.

ArchMsg.Warning.warning_unpredictable_tsize_out_of_range

TAGS UNPREDICTABLE DISPLAY %{TCR}.T%{TTBR:d}SZ (%{TSIZE:d} bits) is out of range. Must be between 25 and %{TSIZE_MAX:d}. Fields:

TCR enum

Which TCR.

TSize signed int

Value of TSize.

TSize_MAX signed int

The maximum allowed value.

TTBR unsigned int

Which TTBR region.

ArchMsg.Warning.warning_unpredictable_unaligned_pc_as_base_register

TAGS **UNPREDICTABLE** DISPLAY The PC value must be word-aligned, otherwise the behavior of the instruction is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_unaligned_pop_stack

TAGS **UNPREDICTABLE** DISPLAY **CONSTRAINED UNPREDICTABLE** unaligned stack frame `%{FRAME_ADDR:x}` in exception-return. Fields:

FRAME_ADDR unsigned int

unaligned stack frame address.

ArchMsg.Warning.warning_unpredictable_unaligned_to_non_writeback

TAGS **UNPREDICTABLE** DISPLAY Unaligned Access to Non-Writeback-Cacheable Memory.

ArchMsg.Warning.warning_unpredictable_vmsa_memattrib

TAGS **UNPREDICTABLE** DISPLAY Unpredictable vmsa memory attribute with texcb value programmed to `%{TEXCB}` when tex remap is `%{TEX_REMAP}`. Fields:

TEXCB unsigned int

texcb value.

TEX_REMAP bool

use tex remap.

ArchMsg.Warning.warning_unpredictable_vtcr_t0sz_sl0

TAGS **UNPREDICTABLE** DISPLAY The combination of VTCR.TOSZ=`%{TOSZ:d}` and VTCR.SL0=`%{SL0:d}` is unpredictable. Fields:

SL0 signed int

VTCR.SL0, starting level for stage 2 translation table walks.

TOSZ signed int

VTCR.TOSZ, required input address range.

ArchMsg.Warning.warning_unpredictable_write_DHCSR

TAGS **UNPREDICTABLE** DISPLAY **UNPREDICTABLE** set of bit-changes in DHCSR. Fields:

NEW unsigned int

data attempted to be written.

OLD unsigned int

previous value.

ArchMsg.Warning.warning_unpredictable_write_SHCSR

TAGS UNPREDICTABLE DISPLAY UNPREDICTABLE setting a pending bit to 1 for an exception with priority \geq execution priority.

ArchMsg.Warning.warning_unsupported_access_to_memory_mapped_register

TAGS UNPREDICTABLE DISPLAY Access to memory mapped register is unsupported.

ArchMsg.Warning.warning_user_jmcr_access

TAGS UNPREDICTABLE DISPLAY User %{WRITE:(write|read)} access to JMCR is unpredictable. Fields:

WRITE unsigned int
access is write.

ArchMsg.Warning.warning_wcr_mask_and_bas

TAGS UNPREDICTABLE DISPLAY DBGWCR_EL1.MASK is non-zero and BAS is not 0xff.

ArchMsg.Warning.warning_wcr_mask_reserved

TAGS UNPREDICTABLE DISPLAY DBGWCR_ELn.MASK is set to a reserved value.

ArchMsg.Warning.warning_wcr_non_configuous_bas

TAGS UNPREDICTABLE DISPLAY DBGWCR_EL1.BAS has non-contiguous set of ones.

ArchMsg.Warning.warning_wvr_masked_bit_not_zero

TAGS UNPREDICTABLE DISPLAY DBGWVRn_EL1.VA contains a bit masked by DBGWCRn_EL1.MASK that is not zero.

ArchMsg.Why.why_branch_target_exception

A branch target exception has occurred. DISPLAY branch target exception: target pc=%{PC}, target opcode=%{OPCODE}. PRIMARY KEY PC. Fields:

BTYPE unsigned int
pstate.BTYPE at time branch taken.

OPCODE unsigned int
opcode at branch target.

PC unsigned int
Virtual Address of branch target.

ArchMsg.Why.why_illegal_state

An illegal mode change has occurred. DISPLAY illegal mode change: %{MESSAGE}. Fields:

MESSAGE string
The message.

BRANCH_MISPREDICT

Simulating branch mispredict. Fields:

PC unsigned int
Origin address (or 0 if unavailable).

BRA_DIR

Direct branches, to immediate address. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISSET enum

The instruction set of the branch instruction.

IS_COND bool

Indicates if this is a conditional branch.

IS_HINTED bool

Whether the conditional instruction is hinted.

IS_ISB bool

Whether the instruction is ISB.

IS_LINK bool

Indicates if this is a branch with link.

PC unsigned int

The address of the branch instruction.

TARGET_ISET enum

The instruction set after the branch.

TARGET_PC unsigned int

The address the instruction branches to.

BRA_INDIR

Indirect branches, perhaps to a register. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISSET enum

The instruction set of the branch instruction.

IS_COND bool

Indicates if this is a conditional branch.

IS_HINTED bool

Whether the conditional instruction is hinted.

IS_ISB bool

Whether the instruction is ISB.

IS_LINK bool

Indicates if this is a branch with link.

PC unsigned int

The address of the branch instruction.

TARGET_ISET enum

The instruction set after the branch.

TARGET_PC unsigned int

The address the instruction branches to.

BRA_INDIR_CCFAIL

Indirect branches fail, perhaps to a register.

BRA_RET_CCFAIL

Branch return fail, perhaps to a register.

CACHE_MAINTENANCE_OP

Cache Maintenance Operation. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

DATA unsigned int

Specified MVA or Set/way.

FUNCTION enum

Clean / Invalidate.

SCOPE enum

Affected region.

SIDE enum

Inst / Data.

CCFAIL

Conditional instruction condition check fail. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CCFAIL_UNC

Unconditional instruction condition check fail. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CCPASS

Conditional instruction condition check pass. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CCPASS_UNC

Unconditional instruction condition check pass. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CODE_CACHE_MAINT

Code cache maintenance. Fields:

END_INCL unsigned int

Inclusive end address for VA or PA requests.

NS bool

Non-secure world for PA.

START unsigned int

Start address for VA or PA requests.

TYPE enum

Request type.

COMPILE_BLOCK_END

Last instruction of basic block translated. Fields:

VADDR unsigned int

Address of next instruction after this basic block.

COMPILE_BLOCK_START

First instruction of basic block translated. Fields:

VADDR unsigned int

Address of first instruction in basic block.

COMPILE_INST

ARM instruction compiled. Fields:

DISASS string

Disassembly of the instruction.

ISET enum

The instruction set of this instruction.

ITSTATE unsigned int

The ITSTATE current for the instruction.

OPCODE unsigned int

The opcode of the instruction.

PC unsigned int

The address of the instruction.

SIZE unsigned int

The size of the instruction in bytes.

CONTEXTIDR

Context ID Register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

NS bool

Secure or nonsecure banked register is accessed.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The new value written.

CONTEXT_SYNC

Called for every context synchronization event. Fields:

ARCH_CSE bool

Architectural Context Synchronization Event.

PC unsigned int

The address of the synchronization instruction.

SYSREG_SYNC bool

System register synchronization event.

CORE_ENDIAN

Core BE8 Big-Endian state changed. Fields:

BE8 bool

Core BE8 Big-Endian state.

CORE_INFO

Static processor attributes. Only triggered by a call to DumpState(). Fields:

ARCH_PROFILE enum

The architecture profile of the core.

CLUSTER_ID unsigned int

The cluster ID of this processor.

CORE_NUM unsigned int

The number of this core in an MP processor.

DETAILED_CORE_FEATURES enum

Detailed core features as a bitfield. Enum values are bitmasks to extract the info from the field.

FPU_VERSION enum

The VFP version implemented by the core.

MEM_ARCH enum

The memory architecture of the core.

MIDR_EL1 unsigned int

MIDR_EL1 value.

NUM_CORES unsigned int

The number of cores in this MP processor.

QUANTUM_SIZE unsigned int

The default quantum size of the core (in ticks).

SECURITY_FEATURES bool

Does the core have security features?.

CORE_LOADS

Processor load accesses. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ACQREL enum

Is this an acquire/release.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DATA unsigned int

The data read or written.

ELEMENT_SIZE unsigned int

Width of each element.

ENDIANNESS enum

Byte ordering of this access.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

LOCK enum

Normal, exclusive or locked access.

MECID unsigned int

Memory Encryption Context ID, if applicable.

MECID2 unsigned int

Memory Encryption Context ID of the second page, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PADDR_TAG unsigned int

If tag checked, the physical address tag for the access.

PAGE_SIZE unsigned intPage size as \log_2 (size in bytes).**PAS enum**

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK, 2=Exclusive Failed, 3=Unsupported.

SIZE unsigned int

Width of the access in bytes. Only required if DATA is not traced.

TAG_CHECKED bool

Is this access tag checked.

TAG_CHECKED2 bool

Is the second page access tag checked.

TRANS bool

Is this a translated access.

VADDR unsigned int

The virtual address of the access.

CORE_REGS

Changes of the core registers R0 to R14. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number, 0 to 14.

MODE enum

Bank of the register accessed.

OLD_VALUE unsigned int

The old value overwritten.

PHYS_ID enum

The physical register accessed.

VALUE unsigned int

The new value written to the register.

CORE_REGS64

Changes of the core registers X0..X30, SP_ELn. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ID enum

The register number, 0-30 for X0-X30, >=32 for SP_ELn.

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

CORE_REGS64_READ

General purpose AArch64 register has been read X0..X30, SP_ELn. Fields:

ID enum

The register number, 0-30 for X0-X30, >=32 for SP_ELn.

VALUE unsigned int

The value read from the register.

CORE_STALLED_TRANSITION

Detect when a core is prevented from progressing PChannel states. Fields:

CURRENT_STATE enum

Current power mode of core.

REASON string

The reason a transition was stalled, and what must be done to unblock it.

TARGET_STATE enum

Desired power mode of core after transition.

CORE_STORES

Processor store accesses. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ACQREL enum

Is this an acquire/release.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DATA unsigned int

The data read or written.

ELEMENT_SIZE unsigned int

Width of each element.

ENDIANNESS enum

Byte ordering of this access.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

LOCK enum

Normal, exclusive or locked access.

MECID unsigned int

Memory Encryption Context ID, if applicable.

MECID2 unsigned int

Memory Encryption Context ID of the second page, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PADDR_TAG unsigned int

If tag checked, the physical address tag for the access.

PAGE_SIZE unsigned int

Page size as $\log_2(\text{size in bytes})$.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK, 2=Exclusive Failed, 3=Unsupported.

SIZE unsigned int

Width of the access in bytes. Only required if DATA is not traced.

TAG_CHECKED bool

Is this access tag checked.

TAG_CHECKED2 bool

Is the second page access tag checked.

TRANS bool

Is this a translated access.

VADDR unsigned int

The virtual address of the access.

CP14_READ

System Coprocessor register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP14 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc1 unsigned int

opcode 1.

opc2 unsigned int

opcode 2.

CP14_WRITE

System Coprocessor register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP14 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc1 unsigned int

opcode 1.

opc2 unsigned int

opcode 2.

CP15_READ

System Coprocessor register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP15 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc1 unsigned int

opcode 1.

opc2 unsigned int

opcode 2.

CP15_READ64

System Coprocessor register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP15 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc unsigned int

opcode 1.

CP15_WRITE

System Control Coprocessor register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP15 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc1 unsigned int

opcode 1.

opc2 unsigned int

opcode 2.

CP15_WRITE64

System Coprocessor register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP15 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc unsigned int

opcode 1.

CPSR

CPSR change. (In AArch64 PSTATE represented in SPSR format). Fields:

CORE_NUM unsigned int

Core number in a multi processor.

OLD_VALUE unsigned int

The old CPSR value.

UNKNOWN unsigned int

Bits within the register that have unknown value.

VALUE unsigned int

The new CPSR value.

CRYPTO_SPEC

Every crypto instruction speculatively executed.

CorePowerStateChange

Core power-up state has changed. DISPLAY Core powered %{PU:(up|down)}. Fields:

PU bool

Whether the core powered up.

DATA_CACHE_SET_ALLOC_TAGS

Set allocation tags from DC GVA instruction. Fields:

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

NSDESC unsigned int

The security state of the access.

PADDR unsigned int

The physical address of the access.

PAS enum

The physical address space of the page.

RESPONSE enum

0=Aborted, 1=OK.

SIZE unsigned int

The size of memory in bytes.

TAG unsigned int

The value of the tag being set.

TAGGING_TYPE enum

Type of Memory tagging (Physical or Virtual).

TAG_VADDR unsigned int

The virtual address of the Virtual Tag if Virtual tagging is used; otherwise 0.

VADDR unsigned int

Virtual address used for setting allocation tags.

DATA_CACHE_ZERO

Zero and invalidate cache line from DC ZVA instruction. Fields:

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

NSDESC unsigned int

The security state of the access.

PADDR unsigned int

The physical address of the access.

PADDR_TAG unsigned int

If tag checked, the physical address tag for the access.

PAS enum

The physical address space of the page.

RESPONSE enum

0=Aborted, 1=OK.

SIZE unsigned int

The size of zero'd memory in bytes.

TAG_CHECKED bool

Is this access tag checked.

VADDR unsigned int

Virtual address of the zero'd cache line.

DEBUG_EVENT

Hardware debug support event. Fields:

EVENT enum

Description of event.

VALUE unsigned int

data value.

DEBUG_TARGET_ISET

Hardware debug target ISA. Fields:

TARGET_ISET enum

The target instruction set of the debug state.

DMI_ALLOCATE

DMI allocated. Fields:

ACCESS_DIRECTION enum

READ / WRITE.

PADDR unsigned int

The physical address of the DMI.

PTR unsigned int

DMI range pointer.

SIDE enum

DSIDE / ISIDE.

SIZE unsigned int

The size of DMI.

VADDR unsigned int

The virtual address of the DMI..

DMI_HIT

DMI hit, access used DMI (only fire for small DMI regions i.e. less than a full page). Fields:

ACCESS_DIRECTION enum

READ / WRITE.

PADDR unsigned int

The physical address of the DMI.

SIDE enum

DSIDE / ISIDE.

SIZE unsigned int

The size of the DMI.

VADDR unsigned int

The virtual address of the DMI..

DMI_REVOKE

DMI revoked. Fields:

ACCESS_DIRECTION enum

READ / WRITE.

PADDR unsigned int

The physical address of the DMI.

SIDE enum

DSIDE / ISIDE.

SIZE unsigned int

The size of DMI.

VADDR unsigned int

The virtual address of the DMI..

END_COMPILE

Compilation end. Fields:

END_OF_PAGE_COUNT unsigned int

Number of basic blocks exited due to page boundary since START_COMPILE.

FETCHFAIL_COUNT unsigned int

Number of basic blocks exited due to ifetch failure START_COMPILE.

INST_COUNT unsigned int

Number of instructions compiled since START_COMPILE.

NONSEQ_COUNT unsigned int

Number of basic blocks exited due to non-sequential instructions since START_COMPILE.

PAGE_STRADDLE_COUNT unsigned int

Number of basic blocks exited due to unaligned instructions crossing page since START_COMPILE.

ESB

Error Synchronization Barrier hit. Fields:

ASYNC bool

ESB is async.

IMPLICIT bool

ESB is implicit.

EXCEPTION

Exception taken. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ESR unsigned int

The value assigned to ESR when exception is taken.

ESR_EC unsigned int

The Value assigned to ESR.EC when exception is taken.

IS_PHYSICAL bool

Physical or Virtual exception.

IS_TRAP bool

Exception coming from a trap.

LR unsigned int

The value assigned to the link register.

PC unsigned int

The location where the exception occurred.

PREFERRED_RETURN unsigned int

The preferred return address for the exception.

TAKEN_LOCALLY bool

Whether exception is taken locally(same exception level).

TARGET_ISET enum

The instruction set of the exception handler code.

TARGET_PC unsigned int

The address the exception branches to.

VECTOR enum

The exception vector.

VECTOR_OFFSET unsigned int

The offset of an exception vector from the base address.

EXCEPTION_END

Every exception completed.

EXCEPTION_INFO

Trace human-readable reason of the current exception for data/instruction aborts, floating point exceptions and unknown exceptions. Fields:

DESCRIPTION string

Reason for current exception.

ESR_EC unsigned int

The value of ESR.EC when exception is taken.

EXCEPTION_RAISE

Every exception raised.

EXCEPTION_RETURN

Branches on leaving exception. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISET enum

The instruction set of the branch instruction.

IS_COND bool

Indicates if this is a conditional branch.

IS_HINTED bool

Whether the conditional instruction is hinted.

IS_ISB bool

Whether the instruction is ISB.

IS_LINK bool

Indicates if this is a branch with link.

PC unsigned int

The address of the branch instruction.

TARGET_ISET enum

The instruction set after the branch.

TARGET_PC unsigned int

The address the instruction branches to.

EXCEPTION_RETURN_PREBRANCH

Exception return event that must occur before PC and processor state have updated.

EXCEPTION_START

Every exception started.

FIQ_TAKEN

FIQ taken exception.

FP_EXCEPTIONS_SIGNALLED

Trace which FP exceptions were signalled as a part of the FP operations performed by an instruction. Fields:

FP_EXCEPTIONS_SIGNALLED_MASK unsigned int

Indicates which FP exceptions were signalled by an instruction. This uses the same encoding as the cumulative floating-point exception bitfields in FPSR.

FP_STATE

Floating point state. Fields:

FPSR unsigned int

FPSR register value.

MASK unsigned int

The mask indicating updated cumulative bits.

FREQ_CHANGED

Core frequency changed. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The instruction count of this core.

NEW_FREQ signed int

The new frequency of this core (expressed in Hz).

OLD_FREQ signed int

The old frequency of this core (expressed in Hz).

HLT

HLT instruction occurred. Fields:

IMM16 unsigned int

The 16-bit immediate value encoded in HLT instruction.

INST

Every instruction executed. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DEBUG_STATE bool

The instruction is executed in debug state.

DISASS string

Disassembly of instruction.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISSET enum

The current instruction set.

ITSTATE unsigned int

The current ITSTATE.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

MODE enum

The mode the core is in.

NS unsigned int

The core's non-secure bit.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

OPCODE unsigned int

The opcode of the instruction.

PADDR unsigned int

The physical address of the instruction.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the instruction.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the page.

PC unsigned int

The address of the instruction.

SECURITY_STATE enum

The core's security state.

SIZE unsigned int

The size of the instruction in bytes.

INST_END

Every instruction completed.

INST_START

Every instruction started. Fields:

DEBUG_STATE bool

The instruction is executed in debug state.

ISET enum

The current instruction set.

MODE enum

The mode the core is in.

NS enum

The current Secure State.

PC unsigned int

The address of the conditional instruction.

SECURITY_STATE enum

The current Security State.

INST_STRADDLE

Instruction straddles boundary. Fields:

DEBUG_STATE bool

The instruction is executed in debug state.

ISET enum

The current instruction set.

MODE enum

The mode the core is in.

NS enum

The current Secure State.

PC unsigned int

The address of the conditional instruction.

SECURITY_STATE enum

The current Security State.

IRQ_TAKEN

IRQ taken exception.

ISB

ISB instruction occurred. Fields:

PC unsigned int

The address of the ISB instruction.

LOAD_MULTIPLE_REGS_LIST

List of destination registers (not source) of a multiple registers load instruction. Fields:

REG string

Register to add into the list.

LOCAL_MONITOR

Local monitor activity. Fields:

PADDR unsigned int

Local Monitor Address.

State enum

State of the monitor (Open/Exclusive).

MAPPING_CREATE

A memory mapping has been created. This is an implementation detail of the model and so may change without notice. Fields:

ASID unsigned int

The ASID used during translation, or 0 if no ASID was used.

IS_VALID bool

Whether the mapping is valid or not.

PADDR unsigned int

Base physical address of region, or 0 if not known.

SIDE enum

I-side or D-side.

SIZE unsigned int

Size of the page, or 0 if not known.

VADDR unsigned int

Base virtual address of region.

VMID unsigned int

The VMID used during translation, or 0 if no VMID was used.

MAPPING_DESTROY

A memory mapping has been destroyed. This is an implementation detail of the model and so may change without notice. Fields:

ASID unsigned int

The ASID used during translation.

CAUSE enum

The cause of invalidation.

PADDR unsigned int

Base physical address of page.

SIDE enum

I-side or D-side.

SIZE unsigned int

Size of the page.

VADDR unsigned int

Base virtual address of page.

VMID unsigned int

The VMID used during translation.

MEMMAP_DEBUG_READ

Memory mapped reads to the debug registers. Fields:

ADDR unsigned int

address.

CORE_NUM unsigned int

Core number in a multi processor.

EXT bool

Whether access is from an external device (such as the DAP).

NS enum

Secure state of the access.

REG_NAME string

Name of the debug register accessed.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

MEMMAP_DEBUG_WRITE

Memory mapped writes to the debug registers. Fields:

ADDR **unsigned int**

address.

CORE_NUM **unsigned int**

Core number in a multi processor.

EXT **bool**

Whether access is from an external device (such as the DAP).

NS **enum**

Secure state of the access.

REG_NAME **string**

Name of the debug register accessed.

UNDEF **bool**

The register accessed is undefined.

VALUE **unsigned int**

The new value written.

MEMTAG_LOADS

Allocation tag memory read; when attributes and system configuration allow tag accesses.

Fields:

ACCESS_TYPE **enum**

The type of instruction performing the access.

ATTR **unsigned int**

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 **unsigned int**

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

NSDESC **unsigned int**

The physical address non-secure bit.

NSDESC2 **unsigned int**

The second page physical address non-secure bit.

NUM_GRANULES **unsigned int**

Number of tags to be Loaded/stored.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK.

TAGGING_TYPE enum

Type of Memory tagging (Physical or Virtual).

TAGS unsigned int

The tags read or written. Tags for subsequent granules would be found at subsequent bytes in this field (One byte for each tag).

TAG_VADDR unsigned int

The virtual address of the Virtual Tag if Virtual tagging is used; otherwise 0.

VADDR unsigned int

The virtual address of the access.

MEMTAG_LOAD_INST

Allocation tag load instruction; regardless of tags being actually accessed. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

NUM_GRANULES unsigned int

Number of tags to be Loaded/stored.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK.

TAGGING_TYPE enum

Type of Memory tagging (Physical or Virtual).

TAGS unsigned int

The tags read or written. Tags for subsequent granules would be found at subsequent bytes in this field (One byte for each tag).

TAG_VADDR unsigned int

The virtual address of the Virtual Tag if Virtual tagging is used; otherwise 0.

VADDR unsigned int

The virtual address of the access.

MEMTAG_STORES

Allocation tag memory write; when attributes and system configuration allow tag accesses.

Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

NUM_GRANULES unsigned int

Number of tags to be Loaded/stored.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK.

TAGGING_TYPE enum

Type of Memory tagging (Physical or Virtual).

TAGS unsigned int

The tags read or written. Tags for subsequent granules would be found at subsequent bytes in this field (One byte for each tag).

TAG_VADDR unsigned int

The virtual address of the Virtual Tag if Virtual tagging is used; otherwise 0.

VADDR unsigned int

The virtual address of the access.

MEMTAG_STORE_INST

Allocation tag store instruction; regardless of tags being actually accessed. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

NUM_GRANULES unsigned int

Number of tags to be Loaded/stored.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK.

TAGGING_TYPE enum

Type of Memory tagging (Physical or Virtual).

TAGS unsigned int

The tags read or written. Tags for subsequent granules would be found at subsequent bytes in this field (One byte for each tag).

TAG_VADDR unsigned int

The virtual address of the Virtual Tag if Virtual tagging is used; otherwise 0.

VADDR unsigned int

The virtual address of the access.

MMU_D128_TTB_READ

This event is triggered by reads caused by a D128 translation table walk. Fields:

ABORTED bool

Was the walk successful.

DATA_HIGH unsigned int

The data upper bits read.

DATA_LOW unsigned int

The data lower bits read.

IPA unsigned int

For stage 1, the IPA of the read.

LEVEL unsigned int

Translation table level.

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

OUTADDR unsigned int

The output physical address.

PADDR unsigned int

The physical address of the read.

PASmem enum

Physical Address Space of the accessed memory.

PASreq enum

Physical Address Space of the requested lookup.

REGIME_EL enum

Entry matches in this translation regime.

REQUESTER enum

The requester of this table walk.

SIDE enum

Inst / Data.

STAGE unsigned int

Translation stage.

MMU_D128_TTB_WRITE

This event is triggered by writes caused by a D128 translation table walk. Fields:

ABORTED bool

Was the walk successful.

DATA_HIGH unsigned int

The data upper bits written.

DATA_LOW unsigned int

The data lower bits written.

IPA unsigned int

For stage 1, the IPA of the write.

IS_HACDBS_ACCESS bool

If the access was due to HACDBS_CLEAN.

LEVEL unsigned int

Translation table level.

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

OUTADDR unsigned int

The output physical address.

PADDR unsigned int

The physical address of the write.

PASmem enum

Physical Address Space of the accessed memory.

PASreq enum

Physical Address Space of the requested lookup.

REGIME_EL enum

Entry matches in this translation regime.

SIDE enum

Inst / Data.

STAGE unsigned int

Translation stage.

MMU_TRANS

Address translation information. Fields:

ASID unsigned int

Address space identifier.

CORE_NUM unsigned int

Core number in a multi processor.

Hyp bool

Entry matches in Hyp state only.

INNERCACHE_RA bool

Is the inner cache allocate on read.

INNERCACHE_TRANSIENT bool

Is the inner write-through transient.

INNERCACHE_TYPE enum

Inner Caching scheme (NC/MB/WA).

INNERCACHE_WA bool

Is the inner cache allocate on write.

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MEMTYPE enum

Memory type.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC enum

Is secure side supposed to access secure or nonsecure memory.

OUTERCACHE_RA bool

Is the outer cache allocate on read.

OUTERCACHE_TRANSIENT bool

Is the outer write-through transient.

OUTERCACHE_TYPE enum

Outer Caching scheme (NC/MB/WA).

OUTERCACHE_WA bool

Is the outer cache allocate on write.

PADDR unsigned int

Physical address of the access.

PAGESIZE unsigned int

Page size as log2(size).

PAS enum

PAS supposed to be accessed is secure or nonsecure memory or Root/Realm if RME is implemented.

SH enum

Shareability.

SIDE enum

Inst / Data.

STAGE1_PERM unsigned int

Stage 1 permissions, [5:3] Privileged access (XWR) [2:0] User access (XWR).

STAGE2_PERM unsigned int

Stage 2 permission mask [5:3] Privileged access (XWR) [2:0] User access (XWR).

VADDR unsigned int

Virtual address of the access.

VMID unsigned int

Virtual machine identifier.

XS bool

XS Attribute.

nG enum

Flag indicating whether ASID will be matched.

MMU_TTB_READ

This event is triggered by reads caused by a translation table walk. Fields:

ABORTED bool

Was the walk successful.

DATA unsigned int

The data read.

DESC_KIND enum

The encoding scheme of the descriptors used for this walk.

IPA unsigned int

For stage 1, the IPA of the read.

LEVEL unsigned int

Translation table level.

LPAE bool

Is this for an LPAE translation (*DEPRECATED*, use DESC_KIND).

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSmem bool

Non secure memory access.

NSreq bool

Non secure request.

OUTADDR **unsigned int**

The output physical address.

PADDR **unsigned int**

The physical address of the read.

PASmem **enum**

Physical Address Space of the accessed memory.

PASreq **enum**

Physical Address Space of the requested lookup.

REGIME_EL **enum**

Entry matches in this translation regime.

REQUESTER **enum**

The requester of this table walk.

SIDE **enum**

Inst / Data.

STAGE **unsigned int**

Translation stage.

MMU_TTB_WRITE

This event is triggered by writes caused by a translation table walk. Fields:

ABORTED **bool**

Was the walk successful.

DATA **unsigned int**

The data written.

DESC_KIND **enum**

The encoding scheme of the descriptors used for this walk.

IPA **unsigned int**

For stage 1, the IPA of the write.

IS_HACDBS_ACCESS **bool**

If the access was due to HACDBS_CLEAN.

LEVEL **unsigned int**

Translation table level.

LPAAE **bool**

Is this for an LPAAE translation (*DEPRECATED*, use DESC_KIND).

MECID **unsigned int**

Memory Encryption Context(MEC) ID, if applicable.

MPAM_PARTID **unsigned int**

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSmem bool

Non secure memory access.

NSreq bool

Non secure request.

OUTADDR unsigned int

The output physical address.

PADDR unsigned int

The physical address of the write.

PASmem enum

Physical Address Space of the accessed memory.

PASreq enum

Physical Address Space of the requested lookup.

REGIME_EL enum

Entry matches in this translation regime.

SIDE enum

Inst / Data.

STAGE unsigned int

Translation stage.

MODE_CHANGE

Mode change. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

MODE enum

The new mode.

NON_SECURE enum

The core's new non-secure bit.

OLD_MODE enum

The old mode.

SECURITY_STATE enum

The new security state.

NOT_TAG_CHECKED_INST

MTE Tag not checked for current instruction.

PERIODIC

Called for every quantum. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The instruction count of this CPU.

PC unsigned int

The address of the next instruction to be executed on this CPU.

PMU_COUNTER_OVERFLOW

PMU counter overflow. Fields:

COUNTER_GROUP enum

Counter group.

COUNTER_RANGE enum

Counter range.

EVENT_ID unsigned int

Event Identifier, field is valid only for Event counter group.

INDEX unsigned int

Counter index (as selected by PMSELR), field is valid only for Event/Cycle counter groups.

INTERRUPT bool

Is interrupt enabled on overflow for this counter.

POINTER_AUTH

Called when execution hits a pointer authentication check. Fields:

AUTH_STATE bool

Authentication result.

KEY_USED enum

Pointer authentication key.

POTINDEX_UPDATE

Value of POT index for current pc changed. Fields:

NEW_POTINDEX unsigned int

New POTIndex.

OLD_POTINDEX unsigned int

Old POTIndex.

PREFETCH_MEMORY64

Prefetch from PRFM or PRFUM instructions. Fields:

PRFOP enum

Prefetch hint.

VADDR unsigned int

Virtual address of the location that should be prefetched.

PRELOAD64_REQUEST

Preload requests from PRFM or PRFUM instructions. Fields:

PRFOP enum

Prefetch hint.

VADDR unsigned int

Virtual address of the location that should be prefetched.

PRELOAD_DATA

Data preload from PLD instruction. Fields:

VADDR unsigned int

Virtual address of the data that should be preloaded.

PRELOAD_INST

Instruction preload from PLI instruction. Fields:

VADDR unsigned int

Virtual address of the instruction that should be preloaded.

PRE_CORE_LOAD

Trace just before a core load. Fields:

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

PRE_CORE_STORE

Trace just before a core store. Fields:

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

PRE_TTB_READ

This event is triggered before reads caused by a translation table walk. Fields:

ADDR unsigned int

The physical address of the read.

PSTATE

PSTATE fields update. Fields:

ALLINT unsigned int

All interrupt mask bit.

BTYPE unsigned int

Branch target identification bit.

CORE_NUM unsigned int

Core number in a multi processor.

DAIF unsigned int

Exception mask bits.

DIT unsigned int

Data Independent Timing bit.

EL unsigned int

Current Exception level field.

EXLOCK unsigned int

Exception return state lock bit.

IL unsigned int

Illegal Execution state bit.

NZCV unsigned int

Condition flags.

PACM unsigned int

PACM bit.

PAN unsigned int

Privileged Access Never state bit.

PM unsigned int

PMU exception mask bit.

PPEND unsigned int

PMU exception pending bit.

SM unsigned int

StreamingMode status bit.

SP unsigned int

Stack pointer register selection bit.

SS unsigned int

Software Step bit.

SSBS unsigned int

Speculative Store Bypass Safe bit.

TCO unsigned int

Tag Check Override bit.

TOUCHED unsigned int

Mask of the fields that have been touched since this trace has last been emitted. Note that touched does not necessarily mean the value changed. The index of the field is the bit index in the mask plus the index of this field plus one.

UAO unsigned int

User Access Override bit.

UINJ unsigned int

Undefined Instruction exceptions bit.

ZA unsigned int

ZA storage and ZT0 register enable bit.

nRW unsigned int

Current Execution state bit.

PSTATE_PACTIVE_OPMODE_TRANSITION

Trace pactive and Opmode mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Operating mode after the transition.

OLD_STATE enum

Operating mode before the transition.

PSTATE_PACTIVE_OPMODE_TRANSITION_DENIED

Trace the denied pactive and Opmode mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Operating mode after the transition.

OLD_STATE enum

Operating mode before the transition.

REASON string

Reason for the transition being denied.

PSTATE_PACTIVE_POWER_TRANSITION

Trace pactive and power mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Power mode after the transition.

OLD_STATE enum

Power mode before the transition.

PSTATE_PACTIVE_POWER_TRANSITION_DENIED

Trace the denied pactive and power mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Power mode after the transition.

OLD_STATE enum

Power mode before the transition.

REASON string

Reason for the transition being denied.

PSTATE_PACTIVE_THREAD_OPMODE_TRANSITION

Trace pactive and Opmode mode transitions for threads on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Operating mode after the transition.

OLD_STATE enum

Operating mode before the transition.

QUANTUM_START

Called at the start of every quantum.

RANGE_PRELOAD_REQUEST

Preload requests from RPRFM. Fields:

COUNT unsigned int

Number of blocks of data to access.

LENGTH unsigned int

Number of contiguous bytes to be accessed in each block.

OP unsigned int

Range prefetch operation .

REUSE unsigned int

Maximum number of bytes to be accessed before next RPRFM .

STRIDE signed int

Stride to advance after accessing LENGTH bytes.

VADDR unsigned int

Base virtual address of range prefetch request.

RUN_STATE

Run state transition. Fields:

INST_COUNT **unsigned int**

Ticks count at point of transition.

NEW **enum**

New run state.

OLD **enum**

Old run state.

SEMIHOSTING_CALL

Call of a semihost function occurred. Fields:

PC **unsigned int**

The program counter after the semihosting call.

SEMIHOSTING_PRECALL

About to call semihost. Fields:

EL2 **bool**

Use EL2 translation regime.

NS **bool**

Is Non-Secure.

REG_WIDTH **unsigned int**

The current register width in bytes.

SIGNAL

External signal state change. Fields:

SIGNAL **enum**

Signal that changed.

STATE **bool**

Signal asserted state.

SOFTWARE_STEP

Return the Debug Software Step state. Fields:

STATE **enum**

Software Step state.

SPSR

SPSR change. Fields:

CORE_NUM **unsigned int**

Core number in a multi processor.

MODE **enum**

Which of the banked SPSR registers is written.

OLD_VALUE unsigned int

The old SPSR value.

VALUE unsigned int

The new SPSR value.

START_COMPILE

Compilation started. Fields:

VADDR unsigned int

Instruction where compilation begins.

STG_MULTIPLE_MICRO_OPS_ADDRESS_RANGE

Address range for MTE stores with multiple uops. Fields:

ADDR_END unsigned int

Final address.

ADDR_START unsigned int

Starting address.

STORE_MULTIPLE_REGS_ADDRESS_RANGE

Address range for stores with multiple registers. Fields:

ADDR_END unsigned int

Final address.

ADDR_START unsigned int

Starting address.

SUBOPTIMAL_LDST_RETIRED

suboptimal load/store. Fields:

ACCESS_SIZE unsigned int

Log2 of the access width in bytes used for alignment checking (1: half-word, 2: word ...).

OPTIMAL_ALIGN_SELECTOR unsigned int

Selector to decide the optimal alignment size (1: access width, 2: 4-bytes, 3: 8-bytes, ..., 12: 4-kbytes).

VADDR unsigned int

The virtual address of the access.

SVE_INST_SPEC

SVE operations speculatively executed.

SVE_LD_RETIRED

SVE load instructions architecturally executed.

SVE_LOADS

SVE load accesses. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ACQREL enum

Is this an acquire/release.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DATA unsigned int

The data read or written.

ELEMENT_SIZE unsigned int

Width of each element.

ENDIANNESS enum

Byte ordering of this access.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

LOCK enum

Normal, exclusive or locked access.

MECID unsigned int

Memory Encryption Context ID, if applicable.

MECID2 unsigned int

Memory Encryption Context ID of the second page, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PADDR_TAG unsigned int

If tag checked, the physical address tag for the access.

PAGE_SIZE unsigned int

Page size as $\log_2(\text{size in bytes})$.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK, 2=Exclusive Failed, 3=Unsupported.

SIZE unsigned int

Width of the access in bytes. Only required if DATA is not traced.

TAG_CHECKED bool

Is this access tag checked.

TAG_CHECKED2 bool

Is the second page access tag checked.

TRANS bool

Is this a translated access.

VADDR unsigned int

The virtual address of the access.

SVE_MEM_PREFETCH

Scalable Vector Extension memory prefetch. Fields:

CACHE_LEVEL enum

The preferred cache level.

HINT enum

What the memory is expected to be used for.

SIZE unsigned int

The prefetch size in bytes.

TEMPORAL bool

Whether the data is temporal.

VADDR unsigned int

The virtual address of the prefetch.

SVE_REG_READ

Scalable Vector Extension register read. Fields:

ID enum

The register identifier.

SM bool

Whether the PE is in Streaming Mode.

VALUE unsigned int

The read data.

SVE_REG_UPDATE

Scalable Vector Extension register update. Fields:

ID enum

The register identifier.

SM bool

Whether the PE is in Streaming Mode.

VALUE unsigned int

The written data.

SVE_REG_WRITE

Scalable Vector Extension register write. Fields:

ID enum

The register identifier.

SM bool

Whether the PE is in Streaming Mode.

VALUE unsigned int

The written data.

SVE_STORES

SVE store accesses. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ACQREL enum

Is this an acquire/release.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DATA unsigned int

The data read or written.

ELEMENT_SIZE unsigned int

Width of each element.

ENDIANNESS enum

Byte ordering of this access.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

LOCK enum

Normal, exclusive or locked access.

MECID unsigned int

Memory Encryption Context ID, if applicable.

MECID2 unsigned int

Memory Encryption Context ID of the second page, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PADDR_TAG unsigned int

If tag checked, the physical address tag for the access.

PAGE_SIZE unsigned intPage size as \log_2 (size in bytes).**PAS enum**

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK, 2=Exclusive Failed, 3=Unsupported.

SIZE unsigned int

Width of the access in bytes. Only required if DATA is not traced.

TAG_CHECKED bool

Is this access tag checked.

TAG_CHECKED2 bool

Is the second page access tag checked.

TRANS bool

Is this a translated access.

VADDR unsigned int

The virtual address of the access.

SVE_ST_RETIRED

SVE store instructions architecturally executed.

SVE_UNALIGNED_LDST_RETIRED

SVE load or store instructions architecturally executed that would cause alignment fault if A bit is set.

SVE_Z_PRED_MEM_ACCESS_INFO

Information about SVE/SME predicated accesses to Z registers (non Gather/Scatter). Fields:

MEM_ELEM_SIZE unsigned int

Width of elements accessed in memory (in bytes).

RnW bool

Whether this is a Read not a Write.

VA unsigned int

Virtual base address computed for accesses.

SYNC

Called for every synchronization, that is every quantum break. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The instruction count of this CPU.

LOCAL_QUANTUM unsigned int

The local quantum of this CPU.

LOCAL_TIME unsigned int

The local time of this CPU.

SYNC_REASON enum

Reason for sync.

SYSCALL

System call instruction executed. Fields:

IMM unsigned int

Immediate value of the system call instruction.

TYPE enum

System call type.

VADDR unsigned int

Instruction that caused the system call.

SYSREG_READ128

System Coprocessor 128 bit register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

REGNUM enum

Internal register number.

REGNUM_PHYS enum

Internal register number physically accessed (e.g. in presense of VHE redirection).

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_READ64

System Coprocessor register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REGNUM enum

Internal register number.

REGNUM_OPERAND enum

Register number corresponding to the instruction operand.

REGNUM_PHYS enum

Internal register number physically accessed (e.g. in presense of VHE redirection).

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_UPDATE128

Triggers when the system updates a register. Fields:

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG enum

Register number.

UNKNOWN unsigned int

Bits of the register which became unknown.

VALUE unsigned int

Value written to the register.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_UPDATE32

Triggers when the system updates a register. Fields:

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG enum

Register number.

UNKNOWN unsigned int

Bits of the register which became unknown.

VALUE unsigned int

Value written to the register.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_UPDATE64

Triggers when the system updates a register. Fields:

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG enum

Register number.

UNKNOWN unsigned int

Bits of the register which became unknown.

VALUE unsigned int

Value written to the register.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_WRITE128

System Coprocessor 128 bit register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

REGNUM enum

Internal register number.

REGNUM_PHYS enum

Internal register number physically accessed (e.g. in presense of VHE redirection).

UNDEF bool

The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int
opcode 2.

SYSREG_WRITE64

System Coprocessor register write. Fields:

CORE_NUM unsigned int
Core number in a multi processor.

CRm unsigned int
CRm.

CRn unsigned int
CRn.

NS bool
Secure or nonsecure banked register is accessed.

REGNUM enum
Internal register number.

REGNUM_OPERAND enum
Register number corresponding to the instruction operand.

REGNUM_PHYS enum
Internal register number physically accessed (e.g. in presense of VHE redirection).

UNDEF bool
The register accessed is undefined.

UPDATED_VALUE unsigned int
Updated value of the register now it has been written.

VALUE unsigned int
The new value written.

opc unsigned int
opcode 1.

opc0 unsigned int
opcode 0.

opc2 unsigned int
opcode 2.

TRBU_BUFFER_WRAP

DISPLAY Trace buffer current write pointer wrapped.

TRBU_TRIGGER_EVENT

DISPLAY Trace buffer Trigger Event.

TRBU_WRITE

DISPLAY PETU trace data has been written to the TRBU memory buffer at
%{BUFFER_ADDRESS} with size %{SIZE} and data %{DATA}. Fields:

BUFFER_ADDRESS unsigned int

Virtual Address of start of buffer write.

DATA unsigned int

Data written to the buffer.

SIZE unsigned int

Size of sample written to buffer, in bytes.

UNALIGNED_LDST_RETIRED

Processor unaligned load/store. Fields:

ACCESS_SIZE unsigned int

Log2 of the access width in bytes used for alignment checking (1=>half-word, 2->word ...).

NUMBER_OF_BEATS unsigned int

Number of accesses (beats) in this burst (Total size in bytes is ACCESS_SIZE * NUMBER_OF_BEATS).

VADDR unsigned int

The virtual address of the access.

VFP_D_REGS

VFP/NEON D 64 bit register write. Fields:

ALIASING enum

Type of register aliasing used.

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a write to the corresponding bit in the register. For example, 0x1 would indicate a write to VALUE[0].

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

VFP_D_REGS_READ

VFP/NEON D 64 bit register read. Fields:

ID unsigned int

The register number.

VALUE unsigned int

The current value of the register.

VFP_Q_REGS

VFP/NEON Q 128 bit register write. Fields:

ALIASING enum

Type of register aliasing used.

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a write to the corresponding byte in the register. For example, 0x1 would indicate a write to VALUE[7:0].

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

VFP_Q_REGS_READ

VFP/NEON Q 128 bit register read. Fields:

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a read to the corresponding byte in the register. For example, 0x1 would indicate a read to VALUE[7:0].

VALUE unsigned int

The current value of the register.

VFP_SYS_REGS

Writes to the VFP/NEON units system registers. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ID enum

Which VFP system register is written.

MASK unsigned int

The mask indicating updated cumulative bits.

OLD_VALUE unsigned int

The register's old value overwritten.

VALUE unsigned int

The new value written to the VFP system register.

VFP_S_REGS

VFP/NEON S 32 bit register write. Fields:

ALIASING enum

Type of register aliasing used.

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a write to the corresponding bit in the register. For example, 0x1 would indicate a write to VALUE[0].

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

VFP_S_REGS_READ

VFP/NEON S 32 bit register read. Fields:

ID unsigned int

The register number.

VALUE unsigned int

The current value of the register.

VIRTUAL_TAG_ALLOCATED

Virtual tag information when allocated. Fields:

TAG_VADDR unsigned int

Virtual tag virtual address.

VADDR unsigned int

Data Virtual address.

VIRTUAL_TAG unsigned int

Virtual tag value.

WAYPOINT

Signals end of basic block execution. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ISSET enum

Origin instruction set.

IS_COND bool

Indicates if this is a conditional waypoint.

PC unsigned int

Origin address (or 0 if unavailable).

TAKEN bool

Indicates if this waypoint was taken.

TARGET unsigned int

Destination address.

TARGET_ISET enum

Destination instruction set.

WFE_END

WFE ended. Fields:

INST_COUNT unsigned int

Ticks count when leaving WFE.

WFE_EVENT_REGISTER

WFE event register status: set/clear, reason. Fields:

INST_COUNT unsigned int

Ticks count.

REASON enum

Reason for set/clear. Only REASON==0 (Cleared by WFE) clears the bit, all other reasons set it.

WFE_IGNORED

WFE ignored. Fields:

EVENT bool

This WFE was ignored because the event register was set.

INST_COUNT unsigned int

Ticks count when ignoring WFE.

TRAPPED bool

This WFE was trapped.

WFE_START

WFE entered. Fields:

INST_COUNT unsigned int

Ticks count when entering WFE.

WFI_END

WFI ended. Fields:

INST_COUNT unsigned int

Ticks count when leaving WFI.

WFI_IGNORED

WFI ignored. Fields:

DISABLED bool

This WFI was ignored because WFI is disabled.

INST_COUNT unsigned int

Ticks count when ignoring WFI.

REASON enum

specifies reason why WFI trace was ignored.

TRAPPED bool

This WFI was trapped.

WFI_START

WFI entered. Fields:

INST_COUNT unsigned int

Ticks count when entering WFI.

WFI_WAKEUP

WFI wakeup. Fields:

INST_COUNT unsigned int

Ticks count when WFI wakeup occurred.

REASON enum

Reason for wakeup.

2.67 ARM_Neoverse-V1

This section describes the trace sources.

AA64_ASE_SVE_REGS

Modification of AArch64 Advanced SIMD or SVE Vector Registers (via SIMD or Scalar operations), SVE Predicate Registers and ZTO. Fields:

ID enum

The register identifier.

MASK unsigned int

Bitmask of the register to signal the modified bits in the VALUE field.

SM bool

Whether the PE is in Streaming Mode.

VALUE unsigned int

Value of AArch64 vector register.

ASYNC_MEMORY_FAULT

Context ID Register write. Fields:

FAULT unsigned int

Fault status present in ESR.ISS encoding for Serror in ARCH64 and present in DFSR encoding for external abort format in ARCH32.

PADDR unsigned int

Physical Address (or 0 if unavailable).

VADDR signed int

Virtual Address (or 0 if unavailable).

ATOMIC_END_ACCESS

Bus trace access for atomics. Fields:

ACCESS_FAIL bool

Memory access failed.

ADDR unsigned int

The virtual address of the access.

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

COMPARE_VALUE unsigned int

Compare value for CAS.

LOAD_VALUE unsigned int

Loaded values.

NSDESC unsigned int

The security state of the access.

OPERAND_VALUE unsigned int

Operation's operand.

OPERATION enum

Operation type.

PADDR unsigned int

The physical address of the access.

PRIV bool

Is this a privileged access?.

ATOMIC_START_ACCESS

Bus trace access for atomics. Fields:

ADDR unsigned int

The virtual address of the access.

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

COMPARE_VALUE unsigned int

Compare value for CAS.

NSDESC unsigned int

The security state of the access.

OPERAND_VALUE unsigned int

Operation's operand.

OPERATION enum

Operation type.

PADDR unsigned int

The physical address of the access.

PRIV bool

Is this a privileged access?.

ArchMsg.Error.exit_code

DISPLAY exit code: %{EXITCODE} component kind: %{KIND} reason: %{REASON}. Fields:

EXITCODE signed int

Exit code to be returned.

KIND string

Component kind that invoked the exit code trace.

REASON string

Reason why the exit code trace was invoked (optional).

ArchMsg.Warning.IgnoredWarmResetRequest

Event fires when software attempts to request warm reset through write to deprecated DBGPRCR.CWRR.

ArchMsg.Warning.RAS_ErrselrUnpredictableState

DISPLAY ERRSELR is in unpredictable state due to value greater than number_of_error_records written to it.

ArchMsg.Warning.RAS_UnpredictableStateWrite

DISPLAY %{REG_NAME} write ignored due to unpredictable ERRSELR state. Fields:

REG_NAME string

Name of register being written.

ArchMsg.Warning.ReportedRevisionRegisterRead

DISPLAY Reported revision and patch do not match that of the cpu. Fields:

ORIGINAL_VALUE unsigned int

The value the register would have based on the modelled revision.

REGISTER string

The name of the register that has ben read.

REPORTED_VALUE unsigned int

The value the register has been overridden to.

ArchMsg.Warning.Unpredictably Indexed PM Event Register

TAGS **UNPREDICTABLE** DISPLAY Event %{IsDirect:(Selector value|Register)} %{SEL} is >= %{N}. Fields:

IsDirect unsigned int

Direct Access rather than indirect.

N unsigned int

Number of Accessible Registers.

SEL unsigned int

Selector.

ArchMsg.Warning.cache_contents_unknown

A cache has been activated without being invalidated and may contain **UNKNOWN** content. DISPLAY %{SIDE:(D|I)}-Cache was enabled in %{REGIME:(SECURE|NON_SECURE|HYP)} regime but was not invalidated since power-on: cache lines could contain **UNKNOWN** content %{SIDE:(Note: D-side caches must invalidate SECURE lines even if the cache is only used by NON_SECURE code in order to ensure that unexpected dirty lines tagged SECURE are not naturally evicted)}. Fields:

REGIME unsigned int

Current translation regime.

SIDE unsigned int

0 d-side cache or unified cache, 1 i-side.

ArchMsg.Warning.dcimvac_matches_watchpoint

TAGS **UNPREDICTABLE** DISPLAY Watchpoints matching an AArch32 DCIMVA are implementation defined.

ArchMsg.Warning.decode_fieldmismatch

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with different fields %{FIELD1} and %{FIELD2}. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_frachitsnegative

TAGS **UNPREDICTABLE** DISPLAY VCVT Instruction is unpredictable with a negative number of fraction bits.

ArchMsg.Warning.decode_initblock

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable in an IT block.

ArchMsg.Warning.decode_initblocknotlast

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable in an IT block when not the last.

ArchMsg.Warning.decode_invalid_condition

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable when the condition is not AL.

ArchMsg.Warning.decode_invalidfieldcombination

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with the values in %{FIELD1} and %{FIELD2}. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_invalidvalue

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with the value in %{FIELD}. Fields:

FIELD enum

Field name.

ArchMsg.Warning.decode_mem_hint_unallocated

TAGS **UNPREDICTABLE** DISPLAY There is no memory hint allocated to this bit pattern.

ArchMsg.Warning.decode_registermatch

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with %{FIELD1} and %{FIELD2} as the same register. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_registermismatch

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with %{FIELD1} and %{FIELD2} as different registers. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_registeroutofrange

TAGS **UNPREDICTABLE** DISPLAY Floating-point instruction is unpredictable indexing beyond the end of the register bank.

ArchMsg.Warning.decode_sbzsbo

TAGS **UNPREDICTABLE** DISPLAY Reserved bits in the instruction are not canonical.

ArchMsg.Warning.decode_transactiontoobig

TAGS **UNPREDICTABLE** DISPLAY Attempt to perform a memory transaction of more than 128 bytes.

ArchMsg.Warning.decode_unpred_other

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable.

ArchMsg.Warning.decode_unpreduseofpc

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable when %{FIELD} is PC. Fields:

FIELD enum

Field name.

ArchMsg.Warning.decode_unpreduseofr13

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable when %{FIELD} is R13. Fields:

FIELD enum

Field name.

ArchMsg.Warning.decode_unpreduseofr13inreglist

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with R13 in register-list.

ArchMsg.Warning.decode_useofd16orhigher

TAGS **UNPREDICTABLE** DISPLAY Attempt to use register D16 or higher where it is not allowed.

ArchMsg.Warning.decode_writebackandbaseinlist

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with write back when the base register is in the transfer list.

ArchMsg.Warning.decode_zeroregistersinlist

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable when no registers are to be transferred.

ArchMsg.Warning.recursive_branch

An instruction is performing a branch that targets the same instruction. If this was intended then a WFI might be more effective DISPLAY recursive branch detected at PC=%{PC}. PRIMARY KEY PC. Fields:

PC unsigned int

Address of instruction causing the branch.

ArchMsg.Warning.recursive_exception

An instruction has generated an exception that targets the same instruction. DISPLAY recursive %{TYPE} exception detected at PC=%{PC} (CPSR=%{CPSR}). PRIMARY KEY PC. Fields:

CPSR unsigned int

Processor state.

PC unsigned int

Address of instruction causing the exception.

TYPE enum

Type of exception.

ArchMsg.Warning.reentrant_vector_catch

TAGS **UNPREDICTABLE** DISPLAY Vector catch debug event on Prefetch or Data abort vector %**{ADDR}**. Fields:

ADDR unsigned int

vector address.

ArchMsg.Warning.reserved_it_state

TAGS **UNPREDICTABLE** DISPLAY Execution with reserved IT state %**{ITSTATE}**. Fields:

ITSTATE unsigned int

reserved IT state.

ArchMsg.Warning.tlb_contents_unknown

Translation has been enabled without invalidating the TLB. DISPLAY Translation was enabled in the %**{PAS:(Secure|Non-secure|Root|Realm)}** EL%**{EL:d}** translation regime (vmid=%**{VMID:x}**) without invalidating %**{INVALIDITY:(|d-side |i-side |)}**entries since power-on: %**{INVALIDITY:(|D||I)}**TLB could contain **UNKNOWN** content. Fields:

EL unsigned int

Exception level of translation regime.

INVALIDITY unsigned int

Which TLB may be invalid (if TLBs are separate).

NON_SECURE bool

Whether the translation regime is non-secure.

PAS unsigned int

Physical address space of translation regime.

VMID unsigned int

Current VMID, if applicable.

ArchMsg.Warning.unknown_DLR_DSPSR

Exiting debug state with unknown DLR or DSPSR.

ArchMsg.Warning.unknown_ELR_SPSR

Returning from debug exception with unknown ELR or SPSR.

ArchMsg.Warning.unpred_FAR_on_LRCPC3_non_atomic_mem_access

TAGS **UNPREDICTABLE** Unpredictable FAR/FSTATUS on LRCPC3 pair instructions resulting in two separate memory accessesDISPLAY Unpredictable FAR on LRCPC3 non atomic

memory access at %{ADDR1} with FSTATUS %{FSTATUS1} or %{ADDR2} with FSTATUS %{FSTATUS2}. Fields:

ADDR1 unsigned int

Possible FAR address 1.

ADDR2 unsigned int

Possible FAR address 2.

FSTATUS1 unsigned int

Possible fault status value 1.

FSTATUS2 unsigned int

Possible fault status value 2.

ArchMsg.Warning.unpred_async_tag_check_fail_on_sve_speculative_load_do_update_ffr

Unpredictable set to FALSE of the FFR element associated to SVE speculative load with an async tag check fail TAGS **UNPREDICTABLE**.

ArchMsg.Warning.unpredictable_a32_breakpoint

TAGS **UNPREDICTABLE** DISPLAY Unpredictable breakpoint on a A32 instruction at %{ADDR1} Offset %{OFFSET} BAS %{BAS} for a %{IS_ADDRESS_MISMATCH_BREAKPOINT:(Mismatch|Match)} Breakpoint number %{N:d} Breakpoint address %{ADDR2}. Fields:

ADDR1 unsigned int

Address.

ADDR2 unsigned int

Breakpoint address.

BAS unsigned int

BAS field.

IS_ADDRESS_MISMATCH_BREAKPOINT bool

Is a Mismatch breakpoint, not a Match breakpoint.

N unsigned int

Breakpoint number.

OFFSET unsigned int

Always has a value of zero.

ArchMsg.Warning.unpredictable_banked_register_access

TAGS **UNPREDICTABLE** DISPLAY Access to banked register (R=%{R:d} SYSm=%{SYSM:2x}) is unpredictable. Fields:

R unsigned int

R bit.

SYSM unsigned int

SYSm.

ArchMsg.Warning.unpredictable_ccfail_undef

TAGS **UNPREDICTABLE** DISPLAY An instruction %{OPCODE} at VA=%{VADDR} failed its condition codes check but would otherwise have caused a non-data-dependent trap or undefined exception. Behaviour is implementation-defined choice of exception or nop. Fields:

OPCODE unsigned int

opcode of the instruction.

VADDR unsigned int

Virtual address of the instruction being executed.

ArchMsg.Warning.unpredictable_ici_bits

TAGS **UNPREDICTABLE** DISPLAY ICI bits do not apply or are architecturally unpredictable for instruction.

ArchMsg.Warning.unpredictable_t32_breakpoint

TAGS **UNPREDICTABLE** DISPLAY Unpredictable breakpoint on a T32 instruction at %{ADDR1} Offset %{OFFSET} BAS %{BAS} for a %{IS_ADDRESS_MISMATCH_BREAKPOINT:(Mismatch|Match)} Breakpoint number %{N:d} Breakpoint address %{ADDR2}. Fields:

ADDR1 unsigned int

Address.

ADDR2 unsigned int

Breakpoint address.

BAS unsigned int

BAS field.

IS_ADDRESS_MISMATCH_BREAKPOINT bool

Is a Mismatch breakpoint, not a Match breakpoint.

N unsigned int

Breakpoint number.

OFFSET unsigned int

Indicates which halfword of a 32-bit Thumb instruction the breakpoint is set on. 0 for the first halfword or if the instruction is 16 bits wide, or 1 for the second halfword.

ArchMsg.Warning.unpredictable_t32_vector_catch

TAGS **UNPREDICTABLE** DISPLAY Unpredictable Vector catch exception on a T32 instruction at %{ADDR} Offset %{OFFSET}. Fields:

ADDR unsigned int

Address.

OFFSET unsigned int

Indicates which halfword of a 32-bit Thumb instruction the vector catch exception is set on. 0 for the first halfword or if the instruction is 16 bits wide, or 1 for the second halfword.

ArchMsg.Warning.unpredictable_watchpoint_far

TAGS **UNPREDICTABLE** DISPLAY The value for FAR/EDWAR for hitting this watchpoint may be anywhere in the range %{LOWER_BOUND} to %{UPPER_BOUND}. Fields:

LOWER_BOUND unsigned int

the lowest address accessed by the instruction that triggered the watchpoint.

UPPER_BOUND unsigned int

the highest watchpointed address accessed by that instruction.

ArchMsg.Warning.unpredictable_watchpoint_far_isscmp

DISPLAY The value for FAR/EDWAR for hitting this watchpoint may be anywhere in the range %{LOWER_BOUND} to %{UPPER_BOUND}. Fields:

LOWER_BOUND unsigned int

the lowest address accessed by the instruction that triggered the watchpoint.

UPPER_BOUND unsigned int

the highest watchpointed address accessed by that instruction.

ArchMsg.Warning.warning_AdvSIMDExpandImmUnexpectedZero

TAGS **UNPREDICTABLE** DISPLAY AdvSIMDExpandImm may treat this immediate value as **UNPREDICTABLE**.

ArchMsg.Warning.warning_ConditionalSMC

TAGS **UNPREDICTABLE** DISPLAY SMC instruction has **UNPREDICTABLE** effects when conditional and when combined with traps.

ArchMsg.Warning.warning_TLBID_out_of_range

TLBID out of range in TLBI according to the configuration of NIS, NVIS, NOS and NVOS. Fields:

NIS unsigned int

NIS.

NOS unsigned int

NOS.

NVIS unsigned int

NVIS.

NVOS unsigned int

NVOS.

TLBID unsigned int

The TLBID value used in the TLBI instruction.

ArchMsg.Warning.warning_TTL_or_TG_mismatches_current_translation_granule

TTL or TG mismatches current translation granule in TLBI VA/RVA/IPA/RIPA. Fields:

CURRENT_TG unsigned int

Current TG as indicated by TCR_ELx or VTCR_EL2.

TLBI_TG unsigned int

TG (or TTL[3:2]) used in TLBI instruction.

ArchMsg.Warning.warning_access_crosses_page_boundary

TAGS **UNPREDICTABLE** DISPLAY Access crosses page boundary.

ArchMsg.Warning.warning_access_crosses_page_boundary_has_fault

TAGS **UNPREDICTABLE** DISPLAY Access that crosses page boundary has fault(s).

ArchMsg.Warning.warning_access_crosses_page_boundary_has_fault_on_both_pages

TAGS **UNPREDICTABLE** DISPLAY Access that crosses page boundary has a fault on each page.

Fields:

ADDR1 unsigned int

FAR address on first page.

ADDR2 unsigned int

FAR address on second page.

FSTATUS1 unsigned int

Fault status on first page.

FSTATUS2 unsigned int

Fault status on second page.

ArchMsg.Warning.warning_access_crosses_page_boundary_spanning_different_memory

TAGS **UNPREDICTABLE** DISPLAY Access at address %{ADDR} crosses page from MEM_TYPE %{MEMTYPE_PAGE1} to %{MEMTYPE_PAGE2}. Fields:

ADDR unsigned int

address of access crossing page.

MEMTYPE_PAGE1 signed int

First Page's memory type (0-Normal, 1-Non-Normal).

MEMTYPE_PAGE2 signed int

Second Page's memory type (0-Normal, 1-Non-Normal).

ArchMsg.Warning.warning_access_wraps_around_memory

TAGS **UNPREDICTABLE** DISPLAY Access wraps around memory in %{WIDTH:d}bit mode.

Fields:

WIDTH unsigned int

Address width.

ArchMsg.Warning.warning_bcr_linking_status

TAGS **UNPREDICTABLE** DISPLAY DBGBCR.{BT2,BT} combination gives unpredictable behaviour.

ArchMsg.Warning.warning_bcr_mask_reserved

TAGS **UNPREDICTABLE** DISPLAY DBGBCR_ELn.MASK is set to a reserved value.

ArchMsg.Warning.warning_bvr_masked_bit_not_zero

TAGS **UNPREDICTABLE** DISPLAY DBGBVRn_EL1.VA contains a bit masked by DBGBCRn_EL1.MASK that is not zero.

ArchMsg.Warning.warning_ccsidr_unimplemented_level

TAGS **UNPREDICTABLE** DISPLAY CCSIDR read while CSSELR %{CSSELR:x} points to an unimplemented cache level. Fields:

CSSELR unsigned int

current effective value of CSSELR.

ArchMsg.Warning.warning_change_to_ns_when_tge_set

TAGS **UNPREDICTABLE** DISPLAY Attempting to change NS to 1 when HCR.TGE = 1.

ArchMsg.Warning.warning_contiguous_bit_check_abort

TAGS **UNPREDICTABLE** DISPLAY An abort occurred whilst attempting to check TLB entry is contiguous at %{ENTRY_ADDR}. Fields:

ENTRY_ADDR unsigned int

address of conflicting TLB entry.

ArchMsg.Warning.warning_contiguous_bit_error

TAGS **UNPREDICTABLE** DISPLAY TLB entry at %{ENTRY_ADDR} was not contiguous with entry at %{CONTIG_ADDR} Contents are %{ENTRY_DATA} but expected %{CONTIG_DATA}. Fields:

CONTIG_ADDR unsigned int

address of the TLB first read, with which this entry is expected to be contiguous.

CONTIG_DATA unsigned int

expected contents based on the entry at CONTIG_ADDR.

ENTRY_ADDR unsigned int

address of conflicting TLB entry.

ENTRY_DATA unsigned int

contents of conflicting TLB entry.

ArchMsg.Warning.warning_contiguous_bit_error_D128

TAGS **UNPREDICTABLE** DISPLAY TLB entry at %{ENTRY_ADDR} was not contiguous with entry at %{CONTIG_ADDR} Contents are %{ENTRY_DATA_HI} %{ENTRY_DATA_LO} but expected %{CONTIG_DATA}. Fields:

CONTIG_ADDR unsigned int

address of the TLB first read, with which this entry is expected to be contiguous.

CONTIG_DATA unsigned int

expected contents based on the entry at CONTIG_ADDR.

ENTRY_ADDR unsigned int

address of conflicting TLB entry.

ENTRY_DATA_HI unsigned int

Higher bit contents of conflicting TLB entry.

ENTRY_DATA_LO unsigned int

Lower bit contents of conflicting TLB entry.

ArchMsg.Warning.warning_cp10_cp11_mismatch

TAGS **UNPREDICTABLE** DISPLAY Unpredictable configuration of CP10 and CP11 controls at EL **%{EL:d}**. Fields:

EL unsigned int

EL owning the control.

ArchMsg.Warning.warning_cp10_cp11_reserved_value

TAGS **UNPREDICTABLE** DISPLAY Unpredictable configuration of CPACR CP10 (**%{CP10}**) or CP11 (**%{CP11}**). Fields:

CP10 unsigned int

CP10 access.

CP11 unsigned int

CP11 access.

ArchMsg.Warning.warning_csselr_level_out_of_range

TAGS **UNPREDICTABLE** DISPLAY CSSELR.Level out of range, written value **%{LEVEL}** with only **%{IMPLEMENTED:d}** levels implemented. Fields:

IMPLEMENTED unsigned int

number of cache levels implemented.

LEVEL unsigned int

written value of CSSELR.Level.

TYPE enum

The type of cache selected by CSSELR.InD and TnD.

ArchMsg.Warning.warning_debug_flow_control_bits_not_obeyed

TAGS **UNPREDICTABLE** DISPLAY Flow control bits for **%{REGISTER}** were not obeyed, causing **%{ERROR}**. Fields:

ERROR enum

type of error resulting.

REGISTER enum

register under consideration.

ArchMsg.Warning.warning_debug_register_access_during_reset

TAGS **UNPREDICTABLE** DISPLAY Unpredictable **%{IS_WRITE:(read|write)}** access to debug register offset **%{OFFSET}** during reset. Fields:

IS_WRITE unsigned int

Write Not Read.

OFFSET unsigned int

Register Offset.

ArchMsg.Warning.warning_decode_cps_inconsistent_fields

TAGS **UNPREDICTABLE** DISPLAY The mode setting fields in the CPS instruction are inconsistent.

ArchMsg.Warning.warning_decode_invalid_state

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable in current exception-level/security state.

ArchMsg.Warning.warning_default_cacheable_mmu_on

TAGS **UNPREDICTABLE** DISPLAY Default cacheable is enabled (HCR.DC = 1) while MMU is enabled.

ArchMsg.Warning.warning_default_cacheable_vmmu_off

TAGS **UNPREDICTABLE** DISPLAY Default cacheable is enabled (HCR.DC = 1) while HCR.VM = 0.

ArchMsg.Warning.warning_deprecated_wvr_bit_2_set

TAGS **UNPREDICTABLE** DISPLAY DBGWVR%{N:d}_EL1=%{WVR:h} has bit 2 set, deprecated in ARMv8. Fields:

N unsigned int

index.

WVR unsigned int

DBGWVR.

ArchMsg.Warning.warning_eret_while_software_step_active_pending

Missing ISB between setting MDSCR_EL1.SS and ERET TAGS **UNPREDICTABLE**.

ArchMsg.Warning.warning_exclusive_to_non_normal

TAGS **UNPREDICTABLE** DISPLAY Exclusive Access to Non-Normal Memory %{ADDR}. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_exclusive_to_non_writeback

TAGS **UNPREDICTABLE** DISPLAY Exclusive Access to Non-Writeback-Cacheable Memory %{ADDR}. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_execute_from_device_memory

TAGS **UNPREDICTABLE** DISPLAY An attempt was made to execute from Device memory.

ArchMsg.Warning.warning_illegal_cpsr_mode

TAGS **UNPREDICTABLE** DISPLAY Writing an illegal or unimplement mode (%{NEW_MODE:h}) to CPSR was unpredictable. Fields:

NEW_MODE unsigned int

New value of CPSR.M.

ArchMsg.Warning.warning_illegal_srs_mode

TAGS **UNPREDICTABLE** DISPLAY Illegal or **UNPREDICTABLE** mode (%{MODE:h}) used for SRS instruction. Fields:

MODE unsigned int

mode for Banked SP.

ArchMsg.Warning.warning_implementation_defined_read_debug_register_in_SCS

Read Access to Debug registers from the processor is IMPLEMENTATION DEFINED TAGS IMP_DEF. Fields:

PPB_OFFSET unsigned int

debug register offset.

ArchMsg.Warning.warning_implementation_defined_sequential_security_transitions_supported

The behavior of sequential instruction fetches that cross from non-secure to secure memory and contain an SG instruction is **CONSTRAINED UNPREDICTABLE** TAGS **UNPREDICTABLE**.

ArchMsg.Warning.warning_implementation_defined_stack_limit_check_supported

It is **IMPLEMENTATION DEFINED** whether stack pointer limit checking is performed for this instruction TAGS IMP_DEF. Fields:

ADDRESS unsigned int

Address of instruction.

OPCODE unsigned int

instruction opcode.

ArchMsg.Warning.warning_implementation_defined_taggednonsharedmemorytype

TAGS IMP_DEF DISPLAY Memory attribute has Tagged, non shared memory type; it is **IMPLEMENTATION DEFINED** whether the memory location is treated as Tagged or non-Tagged.

ArchMsg.Warning.warning_implementation_defined_write_debug_register_in_SCS

Write Access to Debug registers from the processor is IMPLEMENTATION DEFINED TAGS IMP_DEF. Fields:

DATA unsigned int

data attempted to be written.

PPB_OFFSET unsigned int

debug register offset.

ArchMsg.Warning.warning_invalid_tcr_granule

TAGS **UNPREDICTABLE** DISPLAY TCR.TG%{TG_ID:(0|1)} bits have been set to a granule size %{REQUEST:d}K not implemented - using %{SUBSTITUTE:d}K. Fields:

REQUEST signed int

page size requested (or 0 for reserved).

SUBSTITUTE signed int

best guess available page size.

TG_ID bool

bits TG0 or TG1.

ArchMsg.Warning.warning_load_multiple_user_registers_from_user_mode

TAGS UNPREDICTABLE DISPLAY An LDM(user registers) instruction executed from user mode.

ArchMsg.Warning.warning_mte_in_mem_attr_not_supported

TAGS UNPREDICTABLE DISPLAY Memory attribute %{ATTR} has Tagged memory type but Armv8.5-MTE is not implemented. Fields:

ATTR unsigned int

Memory attribute.

ArchMsg.Warning.warning_reserved_breakpoint_state_match

TAGS UNPREDICTABLE DISPLAY %{IS_BREAKPOINT:(Watchpoint|Breakpoint)} programmed with a reserved combination of HMC, SSC and %{IS_BREAKPOINT:(PAC|PMC)}. Fields:

IS_BREAKPOINT unsigned int

Is a breakpoint, not a watchpoint.

ArchMsg.Warning.warning_shareability

TAGS UNPREDICTABLE DISPLAY Unpredictable: combination of the 1st and 2nd stages of translation is Normal Inner Non-Cacheable, Outer Non-Cacheable.

ArchMsg.Warning.warning_software_step_set_while_enabled

TAGS UNPREDICTABLE DISPLAY MDSCR_EL1.SS set to 1 while software step debug exceptions are enabled.

ArchMsg.Warning.warning_thumb_instruction_wraps_around_memory

TAGS UNPREDICTABLE DISPLAY Thumb instruction wraps around memory.

ArchMsg.Warning.warning_ttbr_sbz_bits_are_not_zero

TAGS UNPREDICTABLE DISPLAY %{TTBR}[%{MSB:d};0] should be zero, but are not. Fields:

MSB unsigned int

x-1.

TTBR enum

which TTBR register.

ArchMsg.Warning.warning_unaligned_address_dbgdtrrx_write

TAGS UNPREDICTABLE DISPLAY Unaligned address %{ADDR} in X0/R0 when DBGDTRRX written in memory access mode. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unaligned_address_dbgdtrtx_read

TAGS **UNPREDICTABLE** DISPLAY Unaligned address %{ADDR} in X0/R0 when DBGDTRTX read in memory access mode. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unaligned_to_device

TAGS **UNPREDICTABLE** DISPLAY Unaligned Access to Device Memory %{ADDR}. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unaligned_to_strongly_ordered

TAGS **UNPREDICTABLE** DISPLAY Unaligned Access to Strongly Ordered Memory %{ADDR}. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unknown_sau_rnr

TAGS **UNPREDICTABLE** DISPLAY SAU_RNR was set to an unsupported value.

ArchMsg.Warning.warning_unpredictable_AIRCR_PRIS_and_BFHFNMINs

TAGS **UNPREDICTABLE** DISPLAY The effect of setting both AIRCR.BFHFNMINs and AIRCR.PRIS to 1 is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_AIRCR_VECTCLRACTIVE_when_not_in_debug

TAGS **UNPREDICTABLE** DISPLAY The effect of writing a 1 to AIRCR.VECTCLRACTIVE if the processor is not halted in Debug state is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_AIRCR_VECTRESET_and_SYSRESETREQ

TAGS **UNPREDICTABLE** DISPLAY When the processor is halted in Debug state, if a write to the register writes a 1 to both VECTRESET and SYSRESETREQ, the behavior is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_AIRCR_VECTRESET_when_not_in_debug

TAGS **UNPREDICTABLE** DISPLAY The effect of writing a 1 to AIRCR.VECTRESET if the processor is not halted in Debug state is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_AIRCR_incorrect_VKEY

TAGS **UNPREDICTABLE** DISPLAY The value 0x05FA must be written to AIRCR.VKEY, otherwise the register write is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_EXC_RETURN_Reserved_Bit

EXC_RETURN[23:7] are reserved with the special condition that all bits should be written as one. Values other than all 1s are **UNPREDICTABLE**. TAGS **UNPREDICTABLE**. Fields:

EXC_RETURN unsigned int

Special PC value.

SBO_MASK unsigned int

Which bits Should Be One to not UNPRED.

ArchMsg.Warning.warning_unpredictable_PMEVTYPErn_ELO_VS_reserved_value

TAGS UNPREDICTABLE DISPLAY Unpredictable reserved value of PMEVTYPER<n>_ELO.VS.

ArchMsg.Warning.warning_unpredictable_change_to_DEMCR_MON_STEP_at_insufficient_priority

TAGS UNPREDICTABLE DISPLAY The effect of changing DEMCR.MON_STEP at an execution priority that is lower than the priority of the DebugMonitor exception is **CONSTRAINED UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_change_to_priority_of_active_exception

TAGS UNPREDICTABLE DISPLAY Changing the priority of an active exception is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_clear_lspact

TAGS UNPREDICTABLE DISPLAY **UNPREDICTABLE** state on context switch. Fields:

DRVEXC_TAKEN unsigned int

Derived exception taken, not pended.

REACHED_LAST_STORE unsigned int

Only the last faulted.

ArchMsg.Warning.warning_unpredictable_dbg_exit_unaligned_dlr

TAGS UNPREDICTABLE DISPLAY Debug exit to AArch32 with DLR[0] set to 1. PC[0] can be set to 0 or to DLR[0].

ArchMsg.Warning.warning_unpredictable_exception_catch

TAGS UNPREDICTABLE DISPLAY Generation of an exception catch event is unpredictable.

ArchMsg.Warning.warning_unpredictable_exception_return_inconsistent_state

TAGS UNPREDICTABLE DISPLAY state on exception return is unpredictable.

ArchMsg.Warning.warning_unpredictable_exception_return_instruction

TAGS UNPREDICTABLE DISPLAY exception return instruction is unpredictable.

ArchMsg.Warning.warning_unpredictable_in_debug_state

TAGS UNPREDICTABLE DISPLAY Instruction is **UNPREDICTABLE** when executed in debug state.

ArchMsg.Warning.warning_unpredictable_mair_encoding

TAGS UNPREDICTABLE DISPLAY Memory attribute %{ATTR} has invalid MAIR encoding; this may lead to unpredictable results. Fields:

ATTR unsigned int

Memory attribute.

ArchMsg.Warning.warning_unpredictable_pmu_counter_access

TAGS UNPREDICTABLE DISPLAY Access to PMU counters from non-secure ELO or EL1 is **UNPREDICTABLE** with MDCR_EL2.HPMN set to 0.

ArchMsg.Warning.warning_unpredictable_prioritization_breakpoint_match_vector_catch

TAGS UNPREDICTABLE DISPLAY Two events with the same priority occurred at the same instruction: Address Matching Vector Catch debug event, and Breakpoint debug event. It is constrained unpredictable which is taken.

ArchMsg.Warning.warning_unpredictable_stack_selection

TAGS **UNPREDICTABLE** DISPLAY **UNPREDICTABLE** state on context switch.

ArchMsg.Warning.warning_unpredictable_tsize_out_of_range

TAGS **UNPREDICTABLE** DISPLAY **%{TCR}.T%{TTBR:d}SZ (%{TSIZE:d} bits)** is out of range. Must be between 25 and **%{TSIZE_MAX:d}**. Fields:

TCR enum

Which TCR.

TSIZE signed int

Value of TSize.

TSIZE_MAX signed int

The maximum allowed value.

TTBR unsigned int

Which TTBR region.

ArchMsg.Warning.warning_unpredictable_unaligned_pc_as_base_register

TAGS **UNPREDICTABLE** DISPLAY The PC value must be word-aligned, otherwise the behavior of the instruction is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_unaligned_pop_stack

TAGS **UNPREDICTABLE** DISPLAY **CONSTRAINED UNPREDICTABLE** unaligned stack frame **%{FRAME_ADDR:x}** in exception-return. Fields:

FRAME_ADDR unsigned int

unaligned stack frame address.

ArchMsg.Warning.warning_unpredictable_unaligned_to_non_writeback

TAGS **UNPREDICTABLE** DISPLAY Unaligned Access to Non-Writeback-Cacheable Memory.

ArchMsg.Warning.warning_unpredictable_vmsa_memattrib

TAGS **UNPREDICTABLE** DISPLAY Unpredictable vmsa memory attribute with texcb value programmed to **%{TEXCB}** when tex remap is **%{TEX_REMAP}**. Fields:

TEXCB unsigned int

texcb value.

TEX_REMAP bool

use tex remap.

ArchMsg.Warning.warning_unpredictable_vtcr_t0sz_sl0

TAGS **UNPREDICTABLE** DISPLAY The combination of **VTCR.TOSZ=%{TOSZ:d}** and **VTCR.SL0=%{SL0:d}** is unpredictable. Fields:

SL0 signed int

VTCR.SL0, starting level for stage 2 translation table walks.

TOSZ signed int

VTCR.TOSZ, required input address range.

ArchMsg.Warning.warning_unpredictable_write_DHCSR

TAGS UNPREDICTABLE DISPLAY UNPREDICTABLE set of bit-changes in DHCSR. Fields:

NEW unsigned int

data attempted to be written.

OLD unsigned int

previous value.

ArchMsg.Warning.warning_unpredictable_write_SHCSR

TAGS UNPREDICTABLE DISPLAY UNPREDICTABLE setting a pending bit to 1 for an exception with priority >= execution priority.

ArchMsg.Warning.warning_unsupported_access_to_memory_mapped_register

TAGS UNPREDICTABLE DISPLAY Access to memory mapped register is unsupported.

ArchMsg.Warning.warning_user_jmcr_access

TAGS UNPREDICTABLE DISPLAY User %{WRITE:(write|read)} access to JMCR is unpredictable. Fields:

WRITE unsigned int

access is write.

ArchMsg.Warning.warning_wcr_mask_and_bas

TAGS UNPREDICTABLE DISPLAY DBGWCR_EL1.MASK is non-zero and BAS is not 0xff.

ArchMsg.Warning.warning_wcr_mask_reserved

TAGS UNPREDICTABLE DISPLAY DBGWCR_ELn.MASK is set to a reserved value.

ArchMsg.Warning.warning_wcr_non_configuous_bas

TAGS UNPREDICTABLE DISPLAY DBGWCR_EL1.BAS has non-contiguous set of ones.

ArchMsg.Warning.warning_wvr_masked_bit_not_zero

TAGS UNPREDICTABLE DISPLAY DBGWVRn_EL1.VA contains a bit masked by DBGWCRn_EL1.MASK that is not zero.

ArchMsg.Why.why_branch_target_exception

A branch target exception has occurred. DISPLAY branch target exception: target pc=%{PC}, target opcode=%{OPCODE}. PRIMARY KEY PC. Fields:

BTYPE unsigned int

pstate.BTYPE at time branch taken.

OPCODE unsigned int

opcode at branch target.

PC unsigned int

Virtual Address of branch target.

ArchMsg.Why.why_illegal_state

An illegal mode change has occurred. DISPLAY illegal mode change: %{MESSAGE}. Fields:

MESSAGE string

The message.

ArchMsg.warning_invalid_interval_counter_reload_value

Invalid interval counter reload value in PMSIRR_EL1.INTERVAL: software should set this to a value greater than or equal to the minimum indicated by PMSIDR_EL1.Interval.

ArchMsg.warning_invalid_spe_buffer_write_pointer

Invalid value programmed in PMBPTR_EL1.PTR or PMBLIMITR_EL1.LIMIT: Current write pointer must be at least one sample record below the write limit pointer and PMBPTR_EL1.PTR[63:56] must be equal to PMBLIMITR_EL1.LIMIT[63:56].

BRANCH_MISPREDICT

Simulating branch mispredict. Fields:

PC unsigned int

Origin address (or 0 if unavailable).

BRA_DIR

Direct branches, to immediate address. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISET enum

The instruction set of the branch instruction.

IS_COND bool

Indicates if this is a conditional branch.

IS_HINTED bool

Whether the conditional instruction is hinted.

IS_ISB bool

Whether the instruction is ISB.

IS_LINK bool

Indicates if this is a branch with link.

PC unsigned int

The address of the branch instruction.

TARGET_ISET enum

The instruction set after the branch.

TARGET_PC unsigned int

The address the instruction branches to.

BRA_INDIR

Indirect branches, perhaps to a register. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISET enum

The instruction set of the branch instruction.

IS_COND bool

Indicates if this is a conditional branch.

IS_HINTED bool

Whether the conditional instruction is hinted.

IS_ISB bool

Whether the instruction is ISB.

IS_LINK bool

Indicates if this is a branch with link.

PC unsigned int

The address of the branch instruction.

TARGET_ISET enum

The instruction set after the branch.

TARGET_PC unsigned int

The address the instruction branches to.

BRA_INDIR_CCFAIL

Indirect branches fail, perhaps to a register.

BRA_RET_CCFAIL

Branch return fail, perhaps to a register.

CACHE_MAINTENANCE_OP

Cache Maintenance Operation. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

DATA unsigned int

Specified MVA or Set/way.

FUNCTION enum

Clean / Invalidate.

SCOPE enum

Affected region.

SIDE enum

Inst / Data.

CCFAIL

Conditional instruction condition check fail. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CCFAIL_UNC

Unconditional instruction condition check fail. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CCPASS

Conditional instruction condition check pass. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CCPASS_UNC

Unconditional instruction condition check pass. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CODE_CACHE_MAINT

Code cache maintenance. Fields:

END_INCL unsigned int

Inclusive end address for VA or PA requests.

NS bool

Non-secure world for PA.

START unsigned int

Start address for VA or PA requests.

TYPE enum

Request type.

COMPILE_BLOCK_END

Last instruction of basic block translated. Fields:

VADDR unsigned int

Address of next instruction after this basic block.

COMPILE_BLOCK_START

First instruction of basic block translated. Fields:

VADDR unsigned int

Address of first instruction in basic block.

COMPILE_INST

ARM instruction compiled. Fields:

DISASS string

Disassembly of the instruction.

ISSET enum

The instruction set of this instruction.

ITSTATE unsigned int

The ITSTATE current for the instruction.

OPCODE unsigned int

The opcode of the instruction.

PC unsigned int

The address of the instruction.

SIZE unsigned int

The size of the instruction in bytes.

CONTEXTIDR

Context ID Register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

NS bool

Secure or nonsecure banked register is accessed.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The new value written.

CONTEXT_SYNC

Called for every context synchronization event. Fields:

ARCH_CSE bool

Architectural Context Synchronization Event.

PC unsigned int

The address of the synchronization instruction.

SYSREG_SYNC bool

System register synchronization event.

CORE_ENDIAN

Core BE8 Big-Endian state changed. Fields:

BE8 bool

Core BE8 Big-Endian state.

CORE_INFO

Static processor attributes. Only triggered by a call to DumpState(). Fields:

ARCH_PROFILE enum

The architecture profile of the core.

CLUSTER_ID unsigned int

The cluster ID of this processor.

CORE_NUM unsigned int

The number of this core in an MP processor.

DETAILED_CORE_FEATURES enum

Detailed core features as a bitfield. Enum values are bitmasks to extract the info from the field.

FPU_VERSION enum

The VFP version implemented by the core.

MEM_ARCH enum

The memory architecture of the core.

MIDR_EL1 unsigned int

MIDR_EL1 value.

NUM_CORES unsigned int

The number of cores in this MP processor.

QUANTUM_SIZE unsigned int

The default quantum size of the core (in ticks).

SECURITY_FEATURES bool

Does the core have security features?.

CORE_LOADS

Processor load accesses. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ACQREL enum

Is this an acquire/release.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DATA unsigned int

The data read or written.

ELEMENT_SIZE unsigned int

Width of each element.

ENDIANNESS enum

Byte ordering of this access.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

LOCK enum

Normal, exclusive or locked access.

MECID unsigned int

Memory Encryption Context ID, if applicable.

MECID2 unsigned int

Memory Encryption Context ID of the second page, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PAGE_SIZE unsigned int

Page size as $\log_2(\text{size in bytes})$.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK, 2=Exclusive Failed, 3=Unsupported.

SIZE unsigned int

Width of the access in bytes. Only required if DATA is not traced.

TRANS bool

Is this a translated access.

VADDR unsigned int

The virtual address of the access.

CORE_REGS

Changes of the core registers R0 to R14. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number, 0 to 14.

MODE enum

Bank of the register accessed.

OLD_VALUE unsigned int

The old value overwritten.

PHYS_ID enum

The physical register accessed.

VALUE unsigned int

The new value written to the register.

CORE_REGS64

Changes of the core registers X0..X30, SP_ELn. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ID enum

The register number, 0-30 for X0-X30, >=32 for SP_ELn.

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

CORE_REGS64_READ

General purpose AArch64 register has been read X0..X30, SP_ELn. Fields:

ID enum

The register number, 0-30 for X0-X30, >=32 for SP_ELn.

VALUE unsigned int

The value read from the register.

CORE_STALLED_TRANSITION

Detect when a core is prevented from progressing PChannel states. Fields:

CURRENT_STATE enum

Current power mode of core.

REASON string

The reason a transition was stalled, and what must be done to unblock it.

TARGET_STATE enum

Desired power mode of core after transition.

CORE_STORES

Processor store accesses. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ACQREL enum

Is this an acquire/release.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DATA unsigned int

The data read or written.

ELEMENT_SIZE unsigned int

Width of each element.

ENDIANNESS enum

Byte ordering of this access.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

LOCK enum

Normal, exclusive or locked access.

MECID unsigned int

Memory Encryption Context ID, if applicable.

MECID2 unsigned int

Memory Encryption Context ID of the second page, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PAGE_SIZE unsigned int

Page size as $\log_2(\text{size in bytes})$.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK, 2=Exclusive Failed, 3=Unsupported.

SIZE unsigned int

Width of the access in bytes. Only required if DATA is not traced.

TRANS bool

Is this a translated access.

VADDR unsigned int

The virtual address of the access.

CP14_READ

System Coprocessor register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP14 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc1 unsigned int

opcode 1.

opc2 unsigned int

opcode 2.

CP14_WRITE

System Coprocessor register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP14 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc1 unsigned int

opcode 1.

opc2 unsigned int

opcode 2.

CP15_READ

System Coprocessor register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP15 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc1 unsigned int

opcode 1.

opc2 unsigned int

opcode 2.

CP15_READ64

System Coprocessor register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP15 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc unsigned int

opcode 1.

CP15_WRITE

System Control Coprocessor register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP15 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc1 unsigned int

opcode 1.

opc2 unsigned int

opcode 2.

CP15_WRITE64

System Coprocessor register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP15 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc unsigned int

opcode 1.

CPSR

CPSR change. (In AArch64 PSTATE represented in SPSR format). Fields:

CORE_NUM unsigned int

Core number in a multi processor.

OLD_VALUE unsigned int

The old CPSR value.

UNKNOWN unsigned int

Bits within the register that have unknown value.

VALUE unsigned int

The new CPSR value.

CRYPTO_SPEC

Every crypto instruction speculatively executed.

CorePowerStateChange

Core power-up state has changed. DISPLAY Core powered %{PU:(up|down)}. Fields:

PU bool

Whether the core powered up.

DATA_CACHE_SET_ALLOC_TAGS

Set allocation tags from DC GVA instruction. Fields:

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

NSDESC unsigned int

The security state of the access.

PADDR unsigned int

The physical address of the access.

PAS enum

The physical address space of the page.

RESPONSE enum

0=Aborted, 1=OK.

SIZE unsigned int

The size of memory in bytes.

TAG unsigned int

The value of the tag being set.

TAGGING_TYPE enum

Type of Memory tagging (Physical or Virtual).

TAG_VADDR unsigned int

The virtual address of the Virtual Tag if Virtual tagging is used; otherwise 0.

VADDR unsigned int

Virtual address used for setting allocation tags.

DATA_CACHE_ZERO

Zero and invalidate cache line from DC ZVA instruction. Fields:

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

NSDESC unsigned int

The security state of the access.

PADDR unsigned int

The physical address of the access.

PADDR_TAG unsigned int

If tag checked, the physical address tag for the access.

PAS enum

The physical address space of the page.

RESPONSE enum

0=Aborted, 1=OK.

SIZE unsigned int

The size of zero'd memory in bytes.

TAG_CHECKED bool

Is this access tag checked.

VADDR unsigned int

Virtual address of the zero'd cache line.

DEBUG_EVENT

Hardware debug support event. Fields:

EVENT enum

Description of event.

VALUE unsigned int

data value.

DEBUG_TARGET_ISET

Hardware debug target ISA. Fields:

TARGET_ISET enum

The target instruction set of the debug state.

DMI_ALLOCATE

DMI allocated. Fields:

ACCESS_DIRECTION enum

READ / WRITE.

PADDR unsigned int

The physical address of the DMI.

PTR unsigned int

DMI range pointer.

SIDE enum

DSIDE / ISIDE.

SIZE unsigned int

The size of DMI.

VADDR unsigned int

The virtual address of the DMI..

DMI_HIT

DMI hit, access used DMI (only fire for small DMI regions i.e. less than a full page). Fields:

ACCESS_DIRECTION enum

READ / WRITE.

PADDR unsigned int

The physical address of the DMI.

SIDE enum

DSIDE / ISIDE.

SIZE unsigned int

The size of the DMI.

VADDR unsigned int

The virtual address of the DMI..

DMI_REVOKE

DMI revoked. Fields:

ACCESS_DIRECTION enum

READ / WRITE.

PADDR unsigned int

The physical address of the DMI.

SIDE enum

DSIDE / ISIDE.

SIZE unsigned int

The size of DMI.

VADDR unsigned int

The virtual address of the DMI..

END_COMPILE

Compilation end. Fields:

END_OF_PAGE_COUNT unsigned int

Number of basic blocks exited due to page boundary since START_COMPILE.

FETCHFAIL_COUNT unsigned int

Number of basic blocks exited due to ifetch failure START_COMPILE.

INST_COUNT unsigned int

Number of instructions compiled since START_COMPILE.

NONSEQ_COUNT unsigned int

Number of basic blocks exited due to non-sequential instructions since START_COMPILE.

PAGE_STRADDLE_COUNT unsigned int

Number of basic blocks exited due to unaligned instructions crossing page since START_COMPILE.

ESB

Error Synchronization Barrier hit. Fields:

ASYNC bool

ESB is async.

IMPLICIT bool

ESB is implicit.

EXCEPTION

Exception taken. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ESR unsigned int

The value assigned to ESR when exception is taken.

ESR_EC unsigned int

The Value assigned to ESR.EC when exception is taken.

IS_PHYSICAL bool

Physical or Virtual exception.

IS_TRAP bool

Exception coming from a trap.

LR unsigned int

The value assigned to the link register.

PC unsigned int

The location where the exception occurred.

PREFERRED_RETURN unsigned int

The preferred return address for the exception.

TAKEN_LOCALLY bool

Whether exception is taken locally(same exception level).

TARGET_ISET enum

The instruction set of the exception handler code.

TARGET_PC unsigned int

The address the exception branches to.

VECTOR enum

The exception vector.

VECTOR_OFFSET unsigned int

The offset of an exception vector from the base address.

EXCEPTION_END

Every exception completed.

EXCEPTION_INFO

Trace human-readable reason of the current exception for data/instruction aborts, floating point exceptions and unknown exceptions. Fields:

DESCRIPTION string

Reason for current exception.

ESR_EC unsigned int

The value of ESR.EC when exception is taken.

EXCEPTION_RAISE

Every exception raised.

EXCEPTION_RETURN

Branches on leaving exception. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISET enum

The instruction set of the branch instruction.

IS_COND bool

Indicates if this is a conditional branch.

IS_HINTED bool

Whether the conditional instruction is hinted.

IS_ISB bool

Whether the instruction is ISB.

IS_LINK bool

Indicates if this is a branch with link.

PC unsigned int

The address of the branch instruction.

TARGET_ISET enum

The instruction set after the branch.

TARGET_PC unsigned int

The address the instruction branches to.

EXCEPTION_RETURN_PREBRANCH

Exception return event that must occur before PC and processor state have updated.

EXCEPTION_START

Every exception started.

FIQ_TAKEN

FIQ taken exception.

FP_EXCEPTIONS_SIGNALLED

Trace which FP exceptions were signalled as a part of the FP operations performed by an instruction. Fields:

FP_EXCEPTIONS_SIGNALLED_MASK unsigned int

Indicates which FP exceptions were signalled by an instruction. This uses the same encoding as the cumulative floating-point exception bitfields in FPSR.

FP_STATE

Floating point state. Fields:

FPSR unsigned int

FPSR register value.

MASK unsigned int

The mask indicating updated cumulative bits.

FREQ_CHANGED

Core frequency changed. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The instruction count of this core.

NEW_FREQ signed int

The new frequency of this core (expressed in Hz).

OLD_FREQ signed int

The old frequency of this core (expressed in Hz).

HLT

HLT instruction occurred. Fields:

IMM16 unsigned int

The 16-bit immediate value encoded in HLT instruction.

INST

Every instruction executed. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DEBUG_STATE bool

The instruction is executed in debug state.

DISASS string

Disassembly of instruction.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISSET enum

The current instruction set.

ITSTATE unsigned int

The current ITSTATE.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

MODE enum

The mode the core is in.

NS unsigned int

The core's non-secure bit.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

OPCODE unsigned int

The opcode of the instruction.

PADDR unsigned int

The physical address of the instruction.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the instruction.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the page.

PC unsigned int

The address of the instruction.

SECURITY_STATE enum

The core's security state.

SIZE unsigned int

The size of the instruction in bytes.

INST_END

Every instruction completed.

INST_START

Every instruction started. Fields:

DEBUG_STATE bool

The instruction is executed in debug state.

ISET enum

The current instruction set.

MODE enum

The mode the core is in.

NS enum

The current Secure State.

PC unsigned int

The address of the conditional instruction.

SECURITY_STATE enum

The current Security State.

INST_STRADDLE

Instruction straddles boundary. Fields:

DEBUG_STATE bool

The instruction is executed in debug state.

ISET enum

The current instruction set.

MODE enum

The mode the core is in.

NS enum

The current Secure State.

PC unsigned int

The address of the conditional instruction.

SECURITY_STATE enum

The current Security State.

IRQ_TAKEN

IRQ taken exception.

ISB

ISB instruction occurred. Fields:

PC unsigned int

The address of the ISB instruction.

LOAD_MULTIPLE_REGS_LIST

List of destination registers (not source) of a multiple registers load instruction. Fields:

REG string

Register to add into the list.

LOCAL_MONITOR

Local monitor activity. Fields:

PADDR unsigned int

Local Monitor Address.

State enum

State of the monitor (Open/Exclusive).

MAPPING_CREATE

A memory mapping has been created. This is an implementation detail of the model and so may change without notice. Fields:

ASID unsigned int

The ASID used during translation, or 0 if no ASID was used.

IS_VALID bool

Whether the mapping is valid or not.

PADDR unsigned int

Base physical address of region, or 0 if not known.

SIDE enum

I-side or D-side.

SIZE unsigned int

Size of the page, or 0 if not known.

VADDR unsigned int

Base virtual address of region.

VMID unsigned int

The VMID used during translation, or 0 if no VMID was used.

MAPPING_DESTROY

A memory mapping has been destroyed. This is an implementation detail of the model and so may change without notice. Fields:

ASID unsigned int

The ASID used during translation.

CAUSE enum

The cause of invalidation.

PADDR unsigned int

Base physical address of page.

SIDE enum

I-side or D-side.

SIZE unsigned int

Size of the page.

VADDR unsigned int

Base virtual address of page.

VMID unsigned int

The VMID used during translation.

MEMMAP_DEBUG_READ

Memory mapped reads to the debug registers. Fields:

ADDR unsigned int

address.

CORE_NUM unsigned int

Core number in a multi processor.

EXT bool

Whether access is from an external device (such as the DAP).

NS enum

Secure state of the access.

REG_NAME string

Name of the debug register accessed.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

MEMMAP_DEBUG_WRITE

Memory mapped writes to the debug registers. Fields:

ADDR **unsigned int**

address.

CORE_NUM **unsigned int**

Core number in a multi processor.

EXT **bool**

Whether access is from an external device (such as the DAP).

NS **enum**

Secure state of the access.

REG_NAME **string**

Name of the debug register accessed.

UNDEF **bool**

The register accessed is undefined.

VALUE **unsigned int**

The new value written.

MMU_D128_TTB_READ

This event is triggered by reads caused by a D128 translation table walk. Fields:

ABORTED **bool**

Was the walk successful.

DATA_HIGH **unsigned int**

The data upper bits read.

DATA_LOW **unsigned int**

The data lower bits read.

IPA **unsigned int**

For stage 1, the IPA of the read.

LEVEL **unsigned int**

Translation table level.

MECID **unsigned int**

Memory Encryption Context(MEC) ID, if applicable.

MPAM_PARTID **unsigned int**

MPAM Partition ID.

MPAM_PMG **unsigned int**

MPAM Performance Monitoring Group.

MPAM_SP **unsigned int**

MPAM Partition ID Space.

OUTADDR unsigned int

The output physical address.

PADDR unsigned int

The physical address of the read.

PASmem enum

Physical Address Space of the accessed memory.

PASreq enum

Physical Address Space of the requested lookup.

REGIME_EL enum

Entry matches in this translation regime.

REQUESTER enum

The requester of this table walk.

SIDE enum

Inst / Data.

STAGE unsigned int

Translation stage.

MMU_D128_TTB_WRITE

This event is triggered by writes caused by a D128 translation table walk. Fields:

ABORTED bool

Was the walk successful.

DATA_HIGH unsigned int

The data upper bits written.

DATA_LOW unsigned int

The data lower bits written.

IPA unsigned int

For stage 1, the IPA of the write.

IS_HACDBS_ACCESS bool

If the access was due to HACDBS_CLEAN.

LEVEL unsigned int

Translation table level.

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

OUTADDR unsigned int

The output physical address.

PADDR unsigned int

The physical address of the write.

PASmem enum

Physical Address Space of the accessed memory.

PASreq enum

Physical Address Space of the requested lookup.

REGIME_EL enum

Entry matches in this translation regime.

SIDE enum

Inst / Data.

STAGE unsigned int

Translation stage.

MMU_TRANS

Address translation information. Fields:

ASID unsigned int

Address space identifier.

CORE_NUM unsigned int

Core number in a multi processor.

Hyp bool

Entry matches in Hyp state only.

INNERCACHE_RA bool

Is the inner cache allocate on read.

INNERCACHE_TRANSIENT bool

Is the inner write-through transient.

INNERCACHE_TYPE enum

Inner Caching scheme (NC/MB/WA).

INNERCACHE_WA bool

Is the inner cache allocate on write.

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MEMTYPE enum

Memory type.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC enum

Is secure side supposed to access secure or nonsecure memory.

OUTERCACHE_RA bool

Is the outer cache allocate on read.

OUTERCACHE_TRANSIENT bool

Is the outer write-through transient.

OUTERCACHE_TYPE enum

Outer Caching scheme (NC/MB/WA).

OUTERCACHE_WA bool

Is the outer cache allocate on write.

PADDR unsigned int

Physical address of the access.

PAGESIZE unsigned int

Page size as log2(size).

PAS enum

PAS supposed to be accessed is secure or nonsecure memory or Root/Realm if RME is implemented.

SH enum

Shareability.

SIDE enum

Inst / Data.

STAGE1_PERM unsigned int

Stage 1 permissions, [5:3] Privileged access (XWR) [2:0] User access (XWR).

STAGE2_PERM unsigned int

Stage 2 permission mask [5:3] Privileged access (XWR) [2:0] User access (XWR).

VADDR unsigned int

Virtual address of the access.

VMID unsigned int

Virtual machine identifier.

XS bool

XS Attribute.

nG enum

Flag indicating whether ASID will be matched.

MMU_TTB_READ

This event is triggered by reads caused by a translation table walk. Fields:

ABORTED bool

Was the walk successful.

DATA unsigned int

The data read.

DESC_KIND enum

The encoding scheme of the descriptors used for this walk.

IPA unsigned int

For stage 1, the IPA of the read.

LEVEL unsigned int

Translation table level.

LPAE bool

Is this for an LPAE translation (*DEPRECATED*, use DESC_KIND).

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSmem bool

Non secure memory access.

NSreq bool

Non secure request.

OUTADDR unsigned int

The output physical address.

PADDR unsigned int

The physical address of the read.

PASmem enum

Physical Address Space of the accessed memory.

PASreq enum

Physical Address Space of the requested lookup.

REGIME_EL enum

Entry matches in this translation regime.

REQUESTER enum

The requester of this table walk.

SIDE enum

Inst / Data.

STAGE unsigned int

Translation stage.

MMU_TTB_WRITE

This event is triggered by writes caused by a translation table walk. Fields:

ABORTED bool

Was the walk successful.

DATA unsigned int

The data written.

DESC_KIND enum

The encoding scheme of the descriptors used for this walk.

IPA unsigned int

For stage 1, the IPA of the write.

IS_HACDBS_ACCESS bool

If the access was due to HACDBS_CLEAN.

LEVEL unsigned int

Translation table level.

LPAE bool

Is this for an LPAE translation (*DEPRECATED*, use DESC_KIND).

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSmem bool

Non secure memory access.

NSreq bool

Non secure request.

OUTADDR unsigned int

The output physical address.

PADDR unsigned int

The physical address of the write.

PASmem enum

Physical Address Space of the accessed memory.

PASreq enum

Physical Address Space of the requested lookup.

REGIME_EL enum

Entry matches in this translation regime.

SIDE enum

Inst / Data.

STAGE unsigned int

Translation stage.

MODE_CHANGE

Mode change. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

MODE enum

The new mode.

NON_SECURE enum

The core's new non-secure bit.

OLD_MODE enum

The old mode.

SECURITY_STATE enum

The new security state.

NOT_TAG_CHECKED_INST

MTE Tag not checked for current instruction.

NV2_CORE_128_LOADS

Processor load accesses for nested virtualization of v9.4 128-bit registers. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ACQREL enum

Is this an acquire/release.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes

([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DATA unsigned int

The data read or written.

ELEMENT_SIZE unsigned int

Width of each element.

ENDIANNESS enum

Byte ordering of this access.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

LOCK enum

Normal, exclusive or locked access.

MECID unsigned int

Memory Encryption Context ID, if applicable.

MECID2 unsigned int

Memory Encryption Context ID of the second page, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PAGE_SIZE unsigned int

Page size as $\log_2(\text{size in bytes})$.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

REGNUM_OPERAND unsigned int

Register number corresponding to the instruction operand.

RESPONSE enum

0=Aborted, 1=OK, 2=Exclusive Failed, 3=Unsupported.

SIZE unsigned int

Width of the access in bytes. Only required if DATA is not traced.

TRANS bool

Is this a translated access.

VADDR unsigned int

The virtual address of the access.

NV2_CORE_128_STORES

Processor store accesses for nested virtualization of v9.4 128-bit registers. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ACQREL enum

Is this an acquire/release.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DATA unsigned int

The data read or written.

ELEMENT_SIZE unsigned int

Width of each element.

ENDIANNESS enum

Byte ordering of this access.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

LOCK enum

Normal, exclusive or locked access.

MECID unsigned int

Memory Encryption Context ID, if applicable.

MECID2 unsigned int

Memory Encryption Context ID of the second page, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PAGE_SIZE unsigned int

Page size as $\log_2(\text{size in bytes})$.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

REGNUM_OPERAND unsigned int

Register number corresponding to the instruction operand.

RESPONSE enum

0=Aborted, 1=OK, 2=Exclusive Failed, 3=Unsupported.

SIZE unsigned int

Width of the access in bytes. Only required if DATA is not traced.

TRANS bool

Is this a translated access.

VADDR unsigned int

The virtual address of the access.

NV2_CORE_LOADS

Processor load accesses for v8.4 nested virtualization. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ACQREL enum

Is this an acquire/release.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DATA unsigned int

The data read or written.

ELEMENT_SIZE unsigned int

Width of each element.

ENDIANNESS enum

Byte ordering of this access.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

LOCK enum

Normal, exclusive or locked access.

MECID unsigned int

Memory Encryption Context ID, if applicable.

MECID2 unsigned int

Memory Encryption Context ID of the second page, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PAGE_SIZE unsigned int

Page size as $\log_2(\text{size in bytes})$.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

REGNUM_OPERAND unsigned int

Register number corresponding to the instruction operand.

RESPONSE enum

0=Aborted, 1=OK, 2=Exclusive Failed, 3=Unsupported.

SIZE unsigned int

Width of the access in bytes. Only required if DATA is not traced.

TRANS bool

Is this a translated access.

VADDR unsigned int

The virtual address of the access.

NV2_CORE_STORES

Processor store accesses for v8.4 nested virtualization. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ACQREL enum

Is this an acquire/release.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DATA unsigned int

The data read or written.

ELEMENT_SIZE unsigned int

Width of each element.

ENDIANNESS enum

Byte ordering of this access.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

LOCK enum

Normal, exclusive or locked access.

MECID unsigned int

Memory Encryption Context ID, if applicable.

MECID2 unsigned int

Memory Encryption Context ID of the second page, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PAGE_SIZE unsigned int

Page size as $\log_2(\text{size in bytes})$.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

REGNUM_OPERAND unsigned int

Register number corresponding to the instruction operand.

RESPONSE enum

0=Aborted, 1=OK, 2=Exclusive Failed, 3=Unsupported.

SIZE unsigned int

Width of the access in bytes. Only required if DATA is not traced.

TRANS bool

Is this a translated access.

VADDR unsigned int

The virtual address of the access.

PERIODIC

Called for every quantum. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The instruction count of this CPU.

PC unsigned int

The address of the next instruction to be executed on this CPU.

PMU_COUNTER_OVERFLOW

PMU counter overflow. Fields:

COUNTER_GROUP enum

Counter group.

COUNTER_RANGE enum

Counter range.

EVENT_ID unsigned int

Event Identifier, field is valid only for Event counter group.

INDEX unsigned int

Counter index (as selected by PMSELR), field is valid only for Event/Cycle counter groups.

INTERRUPT bool

Is interrupt enabled on overflow for this counter.

POINTER_AUTH

Called when execution hits a pointer authentication check. Fields:

AUTH_STATE bool

Authentication result.

KEY_USED enum

Pointer authentication key.

POTINDEX_UPDATE

Value of POT index for current pc changed. Fields:

NEW_POTINDEX unsigned int

New POTIndex.

OLD_POTINDEX unsigned int

Old POTIndex.

PREFETCH_MEMORY64

Prefetch from PRFM or PRFUM instructions. Fields:

PRFOP enum

Prefetch hint.

VADDR unsigned int

Virtual address of the location that should be prefetched.

PRELOAD64_REQUEST

Preload requests from PRFM or PRFUM instructions. Fields:

PRFOP enum

Prefetch hint.

VADDR unsigned int

Virtual address of the location that should be prefetched.

PRELOAD_DATA

Data preload from PLD instruction. Fields:

VADDR unsigned int

Virtual address of the data that should be preloaded.

PRELOAD_INST

Instruction preload from PLI instruction. Fields:

VADDR unsigned int

Virtual address of the instruction that should be preloaded.

PRE_CORE_LOAD

Trace just before a core load. Fields:

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

PRE_CORE_STORE

Trace just before a core store. Fields:

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

PRE_TTB_READ

This event is triggered before reads caused by a translation table walk. Fields:

ADDR unsigned int

The physical address of the read.

PSTATE

PSTATE fields update. Fields:

ALLINT unsigned int

All interrupt mask bit.

BTYP unsigned int

Branch target identification bit.

CORE_NUM unsigned int

Core number in a multi processor.

DAIF unsigned int

Exception mask bits.

DIT unsigned int

Data Independent Timing bit.

EL unsigned int

Current Exception level field.

EXLOCK unsigned int

Exception return state lock bit.

IL unsigned int

Illegal Execution state bit.

NZCV unsigned int

Condition flags.

PACM unsigned int

PACM bit.

PAN unsigned int

Privileged Access Never state bit.

PM unsigned int

PMU exception mask bit.

PPEND unsigned int

PMU exception pending bit.

SM unsigned int

StreamingMode status bit.

SP unsigned int

Stack pointer register selection bit.

SS unsigned int

Software Step bit.

SSBS unsigned int

Speculative Store Bypass Safe bit.

TCO unsigned int

Tag Check Override bit.

TOUCHED unsigned int

Mask of the fields that have been touched since this trace has last been emitted. Note that touched does not necessarily mean the value changed. The index of the field is the bit index in the mask plus the index of this field plus one.

UAO unsigned int

User Access Override bit.

UINJ unsigned int

Undefined Instruction exceptions bit.

ZA unsigned int

ZA storage and ZT0 register enable bit.

nRW unsigned int

Current Execution state bit.

PSTATE_PACTIVE_OPMODE_TRANSITION

Trace pactive and Opmode mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Operating mode after the transition.

OLD_STATE enum

Operating mode before the transition.

PSTATE_PACTIVE_OPMODE_TRANSITION_DENIED

Trace the denied pactive and Opmode mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Operating mode after the transition.

OLD_STATE enum

Operating mode before the transition.

REASON string

Reason for the transition being denied.

PSTATE_PACTIVE_POWER_TRANSITION

Trace pactive and power mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Power mode after the transition.

OLD_STATE enum

Power mode before the transition.

PSTATE_PACTIVE_POWER_TRANSITION_DENIED

Trace the denied pactive and power mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Power mode after the transition.

OLD_STATE enum

Power mode before the transition.

REASON string

Reason for the transition being denied.

PSTATE_PACTIVE_THREAD_OPMODE_TRANSITION

Trace pactive and Opmode mode transitions for threads on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Operating mode after the transition.

OLD_STATE enum

Operating mode before the transition.

QUANTUM_START

Called at the start of every quantum.

RANGE_PRELOAD_REQUEST

Preload requests from RPRFM. Fields:

COUNT unsigned int

Number of blocks of data to access.

LENGTH unsigned int

Number of contiguous bytes to be accessed in each block.

OP unsigned int

Range prefetch operation .

REUSE unsigned int

Maximum number of bytes to be accessed before next RPRFM .

STRIDE signed int

Stride to advance after accessing LENGTH bytes.

VADDR unsigned int

Base virtual address of range prefetch request.

RUN_STATE

Run state transition. Fields:

INST_COUNT unsigned int

Ticks count at point of transition.

NEW enum

New run state.

OLD enum

Old run state.

SBP_SAMPLE_COLLISION

DISPLAY Statistical profiling samples collided .

SBP_SAMPLE_DISCARDED

DISPLAY Operation discarded by statistical profiling unit when configured in discard mode.
Fields:

BUFFER_POINTER unsigned int

Value of buffer pointer.

DATA unsigned int

Data discarded.

SIZE unsigned int

Size of sample, in bytes.

SBP_SAMPLE_END

DISPLAY Statistical profiling of an operation ended.

SBP_SAMPLE_POP

DISPLAY Operation which might be sampled by statistical profiling.

SBP_SAMPLE_RECORDED

DISPLAY Operation sampled by statistical profiling has been written to the buffer at %{BUFFER_ADDRESS} with size %{SIZE} and data %{DATA}. Fields:

BUFFER_ADDRESS unsigned int

Virtual Address of start of buffer write.

DATA unsigned int

Data written to the buffer.

SIZE unsigned int

Size of sample written to buffer, in bytes.

SBP_SAMPLE_START

DISPLAY Statistical profiling of an operation started.

SBP_SAMPLE_TAKEN

DISPLAY Operation sampled by statistical profiling after filtering.

SEMIHOSTING_CALL

Call of a semihost function occurred. Fields:

PC unsigned int

The program counter after the semihosting call.

SEMIHOSTING_PRECALL

About to call semihost. Fields:

EL2 bool

Use EL2 translation regime.

NS bool

Is Non-Secure.

REG_WIDTH unsigned int

The current register width in bytes.

SIGNAL

External signal state change. Fields:

SIGNAL enum

Signal that changed.

STATE bool

Signal asserted state.

SOFTWARE_STEP

Return the Debug Software Step state. Fields:

STATE enum

Software Step state.

SPSR

SPSR change. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

MODE enum

Which of the banked SPSR registers is written.

OLD_VALUE unsigned int

The old SPSR value.

VALUE unsigned int

The new SPSR value.

START_COMPILE

Compilation started. Fields:

VADDR unsigned int

Instruction where compilation begins.

STG_MULTIPLE_MICRO_OPS_ADDRESS_RANGE

Address range for MTE stores with multiple uops. Fields:

ADDR_END unsigned int

Final address.

ADDR_START unsigned int

Starting address.

STORE_MULTIPLE_REGS_ADDRESS_RANGE

Address range for stores with multiple registers. Fields:

ADDR_END unsigned int

Final address.

ADDR_START unsigned int

Starting address.

SUBOPTIMAL_LDST_RETIRED

suboptimal load/store. Fields:

ACCESS_SIZE unsigned int

Log2 of the access width in bytes used for alignment checking (1: half-word, 2: word ...).

OPTIMAL_ALIGN_SELECTOR unsigned int

Selector to decide the optimal alignment size (1: access width, 2: 4-bytes, 3: 8-bytes, ..., 12: 4-kbytes).

VADDR unsigned int

The virtual address of the access.

SVE_INST_SPEC

SVE operations speculatively executed.

SVE_LD_RETIRED

SVE load instructions architecturally executed.

SVE_LOADS

SVE load accesses. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ACQREL enum

Is this an acquire/release.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DATA unsigned int

The data read or written.

ELEMENT_SIZE unsigned int

Width of each element.

ENDIANNESS enum

Byte ordering of this access.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

LOCK enum

Normal, exclusive or locked access.

MECID unsigned int

Memory Encryption Context ID, if applicable.

MECID2 unsigned int

Memory Encryption Context ID of the second page, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PAGE_SIZE unsigned int

Page size as $\log_2(\text{size in bytes})$.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK, 2=Exclusive Failed, 3=Unsupported.

SIZE unsigned int

Width of the access in bytes. Only required if DATA is not traced.

TRANS bool

Is this a translated access.

VADDR unsigned int

The virtual address of the access.

SVE_MEM_PREFETCH

Scalable Vector Extension memory prefetch. Fields:

CACHE_LEVEL enum

The preferred cache level.

HINT enum

What the memory is expected to be used for.

SIZE unsigned int

The prefetch size in bytes.

TEMPORAL bool

Whether the data is temporal.

VADDR unsigned int

The virtual address of the prefetch.

SVE_REG_READ

Scalable Vector Extension register read. Fields:

ID enum

The register identifier.

SM bool

Whether the PE is in Streaming Mode.

VALUE unsigned int

The read data.

SVE_REG_UPDATE

Scalable Vector Extension register update. Fields:

ID enum

The register identifier.

SM bool

Whether the PE is in Streaming Mode.

VALUE unsigned int

The written data.

SVE_REG_WRITE

Scalable Vector Extension register write. Fields:

ID enum

The register identifier.

SM bool

Whether the PE is in Streaming Mode.

VALUE unsigned int

The written data.

SVE_STORES

SVE store accesses. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ACQREL enum

Is this an acquire/release.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DATA unsigned int

The data read or written.

ELEMENT_SIZE unsigned int

Width of each element.

ENDIANNESS enum

Byte ordering of this access.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

LOCK enum

Normal, exclusive or locked access.

MECID unsigned int

Memory Encryption Context ID, if applicable.

MECID2 unsigned int

Memory Encryption Context ID of the second page, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PAGE_SIZE unsigned int

Page size as $\log_2(\text{size in bytes})$.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK, 2=Exclusive Failed, 3=Unsupported.

SIZE unsigned int

Width of the access in bytes. Only required if DATA is not traced.

TRANS bool

Is this a translated access.

VADDR unsigned int

The virtual address of the access.

SVE_ST_RETIRED

SVE store instructions architecturally executed.

SVE_UNALIGNED_LDST_RETIRED

SVE load or store instructions architecturally executed that would cause alignment fault if A bit is set.

SVE_Z_PRED_MEM_ACCESS_INFO

Information about SVE/SME predicated accesses to Z registers (non Gather/Scatter). Fields:

MEM_ELEM_SIZE unsigned int

Width of elements accessed in memory (in bytes).

RnW bool

Whether this is a Read not a Write.

VA unsigned int

Virtual base address computed for accesses.

SYNC

Called for every synchronization, that is every quantum break. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The instruction count of this CPU.

LOCAL_QUANTUM unsigned int

The local quantum of this CPU.

LOCAL_TIME unsigned int

The local time of this CPU.

SYNC_REASON enum

Reason for sync.

SYSCALL

System call instruction executed. Fields:

IMM unsigned int

Immediate value of the system call instruction.

TYPE enum

System call type.

VADDR unsigned int

Instruction that caused the system call.

SYSREG_READ128

System Coprocessor 128 bit register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

REGNUM enum

Internal register number.

REGNUM_PHYS enum

Internal register number physically accessed (e.g. in presense of VHE redirection).

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_READ64

System Coprocessor register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REGNUM enum

Internal register number.

REGNUM_OPERAND enum

Register number corresponding to the instruction operand.

REGNUM_PHYS enum

Internal register number physically accessed (e.g. in presense of VHE redirection).

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_UPDATE128

Triggers when the system updates a register. Fields:

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG enum

Register number.

UNKNOWN unsigned int

Bits of the register which became unknown.

VALUE unsigned int

Value written to the register.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_UPDATE32

Triggers when the system updates a register. Fields:

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG enum

Register number.

UNKNOWN unsigned int

Bits of the register which became unknown.

VALUE unsigned int

Value written to the register.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_UPDATE64

Triggers when the system updates a register. Fields:

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG enum

Register number.

UNKNOWN unsigned int

Bits of the register which became unknown.

VALUE unsigned int

Value written to the register.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_WRITE128

System Coprocessor 128 bit register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

REGNUM enum

Internal register number.

REGNUM_PHYS enum

Internal register number physically accessed (e.g. in presense of VHE redirection).

UNDEF bool

The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_WRITE64

System Coprocessor register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REGNUM enum

Internal register number.

REGNUM_OPERAND enum

Register number corresponding to the instruction operand.

REGNUM_PHYS enum

Internal register number physically accessed (e.g. in presense of VHE redirection).

UNDEF bool

The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

UNALIGNED_LDST_RETIRE

Processor unaligned load/store. Fields:

ACCESS_SIZE unsigned int

Log2 of the access width in bytes used for alignment checking (1=>half-word, 2->word ...).

NUMBER_OF_BEATS unsigned int

Number of accesses (beats) in this burst (Total size in bytes is ACCESS_SIZE * NUMBER_OF_BEATS).

VADDR unsigned int

The virtual address of the access.

VFP_D_REGS

VFP/NEON D 64 bit register write. Fields:

ALIASING enum

Type of register aliasing used.

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a write to the corresponding bit in the register. For example, 0x1 would indicate a write to VALUE[0].

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

VFP_D_REGS_READ

VFP/NEON D 64 bit register read. Fields:

ID unsigned int

The register number.

VALUE unsigned int

The current value of the register.

VFP_Q_REGS

VFP/NEON Q 128 bit register write. Fields:

ALIASING enum

Type of register aliasing used.

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a write to the corresponding byte in the register. For example, 0x1 would indicate a write to VALUE[7:0].

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

VFP_Q_REGS_READ

VFP/NEON Q 128 bit register read. Fields:

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a read to the corresponding byte in the register. For example, 0x1 would indicate a read to VALUE[7:0].

VALUE unsigned int

The current value of the register.

VFP_SYS_REGS

Writes to the VFP/NEON units system registers. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ID enum

Which VFP system register is written.

MASK unsigned int

The mask indicating updated cumulative bits.

OLD_VALUE unsigned int

The register's old value overwritten.

VALUE unsigned int

The new value written to the VFP system register.

VFP_S_REGS

VFP/NEON S 32 bit register write. Fields:

ALIASING enum

Type of register aliasing used.

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a write to the corresponding bit in the register. For example, 0x1 would indicate a write to VALUE[0].

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

VFP_S_REGS_READ

VFP/NEON S 32 bit register read. Fields:

ID unsigned int

The register number.

VALUE unsigned int

The current value of the register.

VIRTUAL_TAG_ALLOCATED

Virtual tag information when allocated. Fields:

TAG_VADDR unsigned int

Virtual tag virtual address.

VADDR unsigned int

Data Virtual address.

VIRTUAL_TAG unsigned int

Virtual tag value.

WAYPOINT

Signals end of basic block execution. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ISET enum

Origin instruction set.

IS_COND bool

Indicates if this is a conditional waypoint.

PC unsigned int

Origin address (or 0 if unavailable).

TAKEN bool

Indicates if this waypoint was taken.

TARGET unsigned int

Destination address.

TARGET_ISET enum

Destination instruction set.

WFE_END

WFE ended. Fields:

INST_COUNT unsigned int

Ticks count when leaving WFE.

WFE_EVENT_REGISTER

WFE event register status: set/clear, reason. Fields:

INST_COUNT unsigned int

Ticks count.

REASON enum

Reason for set/clear. Only REASON==0 (Cleared by WFE) clears the bit, all other reasons set it.

WFE_IGNORED

WFE ignored. Fields:

EVENT bool

This WFE was ignored because the event register was set.

INST_COUNT unsigned int

Ticks count when ignoring WFE.

TRAPPED bool

This WFE was trapped.

WFE_START

WFE entered. Fields:

INST_COUNT unsigned int

Ticks count when entering WFE.

WFI_END

WFI ended. Fields:

INST_COUNT unsigned int

Ticks count when leaving WFI.

WFI_IGNORED

WFI ignored. Fields:

DISABLED bool

This WFI was ignored because WFI is disabled.

INST_COUNT unsigned int

Ticks count when ignoring WFI.

REASON enum

specifies reason why WFI trace was ignored.

TRAPPED bool

This WFI was trapped.

WFI_START

WFI entered. Fields:

INST_COUNT unsigned int

Ticks count when entering WFI.

WFI_WAKEUP

WFI wakeup. Fields:

INST_COUNT unsigned int

Ticks count when WFI wakeup occurred.

REASON enum

Reason for wakeup.

2.68 ARM_Neoverse-V2

This section describes the trace sources.

AA64_ASE_SVE_REGS

Modification of AArch64 Advanced SIMD or SVE Vector Registers (via SIMD or Scalar operations), SVE Predicate Registers and ZT0. Fields:

ID enum

The register identifier.

MASK unsigned int

Bitmask of the register to signal the modified bits in the VALUE field.

SM bool

Whether the PE is in Streaming Mode.

VALUE unsigned int

Value of AArch64 vector register.

ASYNC_MEMORY_FAULT

Context ID Register write. Fields:

FAULT unsigned int

Fault status present in ESR.ISS encoding for Serror in ARCH64 and present in DFSR encoding for external abort format in ARCH32.

PADDR unsigned int

Physical Address (or 0 if unavailable).

VADDR signed int

Virtual Address (or 0 if unavailable).

ATOMIC_END_ACCESS

Bus trace access for atomics. Fields:

ACCESS_FAIL bool

Memory access failed.

ADDR unsigned int

The virtual address of the access.

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory

attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

COMPARE_VALUE unsigned int

Compare value for CAS.

LOAD_VALUE unsigned int

Loaded values.

NSDESC unsigned int

The security state of the access.

OPERAND_VALUE unsigned int

Operation's operand.

OPERATION enum

Operation type.

PADDR unsigned int

The physical address of the access.

PRIV bool

Is this a privileged access?.

ATOMIC_START_ACCESS

Bus trace access for atomics. Fields:

ADDR unsigned int

The virtual address of the access.

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

COMPARE_VALUE unsigned int

Compare value for CAS.

NSDESC unsigned int

The security state of the access.

OPERAND_VALUE unsigned int

Operation's operand.

OPERATION enum

Operation type.

PADDR unsigned int

The physical address of the access.

PRIV bool

Is this a privileged access?.

ArchMsg.Error.exit_code

DISPLAY exit code: %{EXITCODE} component kind: %{KIND} reason: %{REASON}. Fields:

EXITCODE signed int

Exit code to be returned.

KIND string

Component kind that invoked the exit code trace.

REASON string

Reason why the exit code trace was invoked (optional).

ArchMsg.Warning.IgnoredWarmResetRequest

Event fires when software attempts to request warm reset through write to deprecated DBGPRCR.CWRR.

ArchMsg.Warning.RAS_ErrselrUnpredictableState

DISPLAY ERRSELR is in unpredictable state due to value greater than number_of_error_records written to it.

ArchMsg.Warning.RAS_UnpredictableStateWrite

DISPLAY %{REG_NAME} write ignored due to unpredictable ERRSELR state. Fields:

REG_NAME string

Name of register being written.

ArchMsg.Warning.ReportedRevisionRegisterRead

DISPLAY Reported revision and patch do not match that of the cpu. Fields:

ORIGINAL_VALUE unsigned int

The value the register would have based on the modelled revision.

REGISTER string

The name of the register that has ben read.

REPORTED_VALUE unsigned int

The value the register has been overridden to.

ArchMsg.Warning.Unpredictably Indexed PM Event Register

TAGS UNPREDICTABLE DISPLAY Event %{IsDirect:(Selector value|Register)} %{SEL} is >= %{N}. Fields:

IsDirect unsigned int

Direct Access rather than indirect.

N unsigned int

Number of Accessible Registers.

SEL unsigned int

Selector.

ArchMsg.Warning.cache_contents_unknown

A cache has been activated without being invalidated and may contain **UNKNOWN** content. DISPLAY %{SIDE:(D|I)}-Cache was enabled in %{REGIME:(SECURE|NON_SECURE|HYP)} regime but was not invalidated since power-on: cache lines could contain **UNKNOWN** content %{SIDE:(Note: D-side caches must invalidate SECURE lines even if the cache is only used by NON_SECURE code in order to ensure that unexpected dirty lines tagged SECURE are not naturally evicted)}. Fields:

REGIME unsigned int

Current translation regime.

SIDE unsigned int

0 d-side cache or unified cache, 1 i-side.

ArchMsg.Warning.dcimvac_matches_watchpoint

TAGS **UNPREDICTABLE** DISPLAY Watchpoints matching an AArch32 DCIMVA are implementation defined.

ArchMsg.Warning.decode_fieldmismatch

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with different fields %{FIELD1} and %{FIELD2}. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_fracbitsnegative

TAGS **UNPREDICTABLE** DISPLAY VCVT Instruction is unpredictable with a negative number of fraction bits.

ArchMsg.Warning.decode_initblock

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable in an IT block.

ArchMsg.Warning.decode_initblocknotlast

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable in an IT block when not the last.

ArchMsg.Warning.decode_invalid_condition

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable when the condition is not AL.

ArchMsg.Warning.decode_invalidfieldcombination

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with the values in %{FIELD1} and %{FIELD2}. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_invalidvalue

TAGS UNPREDICTABLE DISPLAY Instruction is unpredictable with the value in %{FIELD}. Fields:

FIELD enum

Field name.

ArchMsg.Warning.decode_mem_hint_unallocated

TAGS UNPREDICTABLE DISPLAY There is no memory hint allocated to this bit pattern.

ArchMsg.Warning.decode_registermatch

TAGS UNPREDICTABLE DISPLAY Instruction is unpredictable with %{FIELD1} and %{FIELD2} as the same register. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_registermismatch

TAGS UNPREDICTABLE DISPLAY Instruction is unpredictable with %{FIELD1} and %{FIELD2} as different registers. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_registeroutofrange

TAGS UNPREDICTABLE DISPLAY Floating-point instruction is unpredictable indexing beyond the end of the register bank.

ArchMsg.Warning.decode_sbzsbo

TAGS UNPREDICTABLE DISPLAY Reserved bits in the instruction are not canonical.

ArchMsg.Warning.decode_transactiontoobig

TAGS UNPREDICTABLE DISPLAY Attempt to perform a memory transaction of more than 128 bytes.

ArchMsg.Warning.decode_unpred_other

TAGS UNPREDICTABLE DISPLAY Instruction is unpredictable.

ArchMsg.Warning.decode_unpreduseofpc

TAGS UNPREDICTABLE DISPLAY Instruction is unpredictable when %{FIELD} is PC. Fields:

FIELD enum

Field name.

ArchMsg.Warning.decode_unpreduseofr13

TAGS UNPREDICTABLE DISPLAY Instruction is unpredictable when %{FIELD} is R13. Fields:

FIELD enum

Field name.

ArchMsg.Warning.decode_unpreduseofr13inreglist

TAGS UNPREDICTABLE DISPLAY Instruction is unpredictable with R13 in register-list.

ArchMsg.Warning.decode_useofd16orhigher

TAGS UNPREDICTABLE DISPLAY Attempt to use register D16 or higher where it is not allowed.

ArchMsg.Warning.decode_writebackandbaseinlist

TAGS UNPREDICTABLE DISPLAY Instruction is unpredictable with write back when the base register is in the transfer list.

ArchMsg.Warning.decode_zeroregistersinlist

TAGS UNPREDICTABLE DISPLAY Instruction is unpredictable when no registers are to be transferred.

ArchMsg.Warning.misaligned_ptrs

DISPLAY TRBPTR_EL1(%{TRBPTR_EL1}) or TRBTRG_EL1(%{TRBTRG_EL1}) are not aligned as per TRBIDR_EL1.Align(%{TRBIDR_EL1}). Fields:

TRBIDR_EL1 unsigned int

TRBIDR_EL1.Align value.

TRBPTR_EL1 unsigned int

Current write pointer.

TRBTRG_EL1 unsigned int

Trigger pointer.

ArchMsg.Warning.out_of_range_ptrs

DISPLAY TRBPTR_EL1(%{TRBPTR_EL1}) is out of range as per TRBBASER_EL1(%{TRBBASER_EL1}) or TRBLIMITR_EL1.LIMIT(%{TRBLIMITR_EL1}). Fields:

TRBBASER_EL1 unsigned int

Base pointer.

TRBLIMITR_EL1 unsigned int

Limit pointer.

TRBPTR_EL1 unsigned int

Current write pointer.

ArchMsg.Warning.recursive_branch

An instruction is performing a branch that targets the same instruction. If this was intended then a WFI might be more effective DISPLAY recursive branch detected at PC=%{PC}. PRIMARY KEY PC. Fields:

PC unsigned int

Address of instruction causing the branch.

ArchMsg.Warning.recursive_exception

An instruction has generated an exception that targets the same instruction. DISPLAY recursive %{TYPE} exception detected at PC=%{PC} (CPSR=%{CPSR}). PRIMARY KEY PC. Fields:

CPSR unsigned int

Processor state.

PC unsigned int

Address of instruction causing the exception.

TYPE enum

Type of exception.

ArchMsg.Warning.reentrant_vector_catch

TAGS **UNPREDICTABLE** DISPLAY Vector catch debug event on Prefetch or Data abort vector %{ADDR}. Fields:

ADDR unsigned int

vector address.

ArchMsg.Warning.reserved_it_state

TAGS **UNPREDICTABLE** DISPLAY Execution with reserved IT state %{ITSTATE}. Fields:

ITSTATE unsigned int

reserved IT state.

ArchMsg.Warning.tlb_contents_unknown

Translation has been enabled without invalidating the TLB. DISPLAY Translation was enabled in the %{PAS:(Secure|Non-secure|Root|Realm)} EL%{EL:d} translation regime (vmid=%{VMID:x}) without invalidating %{INVALIDITY:(|d-side |i-side |)}entries since power-on: %{INVALIDITY:(|D||)}TLB could contain **UNKNOWN** content. Fields:

EL unsigned int

Exception level of translation regime.

INVALIDITY unsigned int

Which TLB may be invalid (if TLBs are separate).

NON_SECURE bool

Whether the translation regime is non-secure.

PAS unsigned int

Physical address space of translation regime.

VMID unsigned int

Current VMID, if applicable.

ArchMsg.Warning.unknown_DLR_DSPSR

Exiting debug state with unknown DLR or DSPSR.

ArchMsg.Warning.unknown_ELR_SPSR

Returning from debug exception with unknown ELR or SPSR.

ArchMsg.Warning.unpred_FAR_on_LRCPC3_non_atomic_mem_access

TAGS **UNPREDICTABLE** Unpredictable FAR/FSTATUS on LRCPC3 pair instructions resulting in two separate memory accesses
DISPLAY Unpredictable FAR on LRCPC3 non atomic memory access at %{ADDR1} with FSTATUS %{FSTATUS1} or %{ADDR2} with FSTATUS %{FSTATUS2}. Fields:

ADDR1 unsigned int

Possible FAR address 1.

ADDR2 unsigned int

Possible FAR address 2.

FSTATUS1 unsigned int

Possible fault status value 1.

FSTATUS2 unsigned int

Possible fault status value 2.

ArchMsg.Warning.unpred_async_tag_check_fail_on_sve_speculative_load_do_update_ffr

Unpredictable set to FALSE of the FFR element associated to SVE speculative load with an async tag check fail TAGS **UNPREDICTABLE**.

ArchMsg.Warning.unpredictable_a32_breakpoint

TAGS **UNPREDICTABLE** DISPLAY Unpredictable breakpoint on a A32 instruction at %{ADDR1} Offset %{OFFSET} BAS %{BAS} for a %{IS_ADDRESS_MISMATCH_BREAKPOINT:(Mismatch|Match)} Breakpoint number %{N:d} Breakpoint address %{ADDR2}. Fields:

ADDR1 unsigned int

Address.

ADDR2 unsigned int

Breakpoint address.

BAS unsigned int

BAS field.

IS_ADDRESS_MISMATCH_BREAKPOINT bool

Is a Mismatch breakpoint, not a Match breakpoint.

N unsigned int

Breakpoint number.

OFFSET unsigned int

Always has a value of zero.

ArchMsg.Warning.unpredictable_banked_register_access

TAGS **UNPREDICTABLE** DISPLAY Access to banked register (R=%{R:d} SYSm=%{SYSM:2x}) is unpredictable. Fields:

R unsigned int

R bit.

SYSM unsigned int

SYSm.

ArchMsg.Warning.unpredictable_ccfail_undef

TAGS **UNPREDICTABLE** DISPLAY An instruction %{OPCODE} at VA=%{VADDR} failed its condition codes check but would otherwise have caused a non-data-dependent trap or undefined exception. Behaviour is implementation-defined choice of exception or nop. Fields:

OPCODE unsigned int

opcode of the instruction.

VADDR unsigned int

Virtual address of the instruction being executed.

ArchMsg.Warning.unpredictable_ici_bits

TAGS **UNPREDICTABLE** DISPLAY ICI bits do not apply or are architecturally unpredictable for instruction.

ArchMsg.Warning.unpredictable_t32_breakpoint

TAGS **UNPREDICTABLE** DISPLAY Unpredictable breakpoint on a T32 instruction at %{ADDR1} Offset %{OFFSET} BAS %{BAS} for a %{IS_ADDRESS_MISMATCH_BREAKPOINT:(Mismatch|Match)} Breakpoint number %{N:d} Breakpoint address %{ADDR2}. Fields:

ADDR1 unsigned int

Address.

ADDR2 unsigned int

Breakpoint address.

BAS unsigned int

BAS field.

IS_ADDRESS_MISMATCH_BREAKPOINT bool

Is a Mismatch breakpoint, not a Match breakpoint.

N unsigned int

Breakpoint number.

OFFSET unsigned int

Indicates which halfword of a 32-bit Thumb instruction the breakpoint is set on. 0 for the first halfword or if the instruction is 16 bits wide, or 1 for the second halfword.

ArchMsg.Warning.unpredictable_t32_vector_catch

TAGS **UNPREDICTABLE** DISPLAY Unpredictable Vector catch exception on a T32 instruction at %{ADDR} Offset %{OFFSET}. Fields:

ADDR unsigned int

Address.

OFFSET unsigned int

Indicates which halfword of a 32-bit Thumb instruction the vector catch exception is set on. 0 for the first halfword or if the instruction is 16 bits wide, or 1 for the second halfword.

ArchMsg.Warning.unpredictable_watchpoint_far

TAGS **UNPREDICTABLE** DISPLAY The value for FAR/EDWAR for hitting this watchpoint may be anywhere in the range %{LOWER_BOUND} to %{UPPER_BOUND}. Fields:

LOWER_BOUND unsigned int

the lowest address accessed by the instruction that triggered the watchpoint.

UPPER_BOUND unsigned int

the highest watchpointed address accessed by that instruction.

ArchMsg.Warning.unpredictable_watchpoint_far_isscmp

DISPLAY The value for FAR/EDWAR for hitting this watchpoint may be anywhere in the range %{LOWER_BOUND} to %{UPPER_BOUND}. Fields:

LOWER_BOUND unsigned int

the lowest address accessed by the instruction that triggered the watchpoint.

UPPER_BOUND unsigned int

the highest watchpointed address accessed by that instruction.

ArchMsg.Warning.warning_AdvSIMDExpandImmUnexpectedZero

TAGS **UNPREDICTABLE** DISPLAY AdvSIMDExpandImm may treat this immediate value as **UNPREDICTABLE**.

ArchMsg.Warning.warning_ConditionalSMC

TAGS **UNPREDICTABLE** DISPLAY SMC instruction has **UNPREDICTABLE** effects when conditional and when combined with traps.

ArchMsg.Warning.warning_TLBID_out_of_range

TLBID out of range in TLBI according to the configuration of NIS, NVIS, NOS and NVOS. Fields:

NIS unsigned int

NIS.

NOS unsigned int

NOS.

NVIS unsigned int

NVIS.

NVOS unsigned int

NVOS.

TLBID unsigned int

The TLBID value used in the TLBI instruction.

ArchMsg.Warning.warning_TTL_or_TG_mismatches_current_translation_granule

TTL or TG mismatches current translation granule in TLBI VA/RVA/IPA/RIPA. Fields:

CURRENT_TG unsigned int

Current TG as indicated by TCR_ELx or VTCR_EL2.

TLBI_TG unsigned int

TG (or TTL[3:2]) used in TLBI instruction.

ArchMsg.Warning.warning_access_crosses_page_boundary

TAGS **UNPREDICTABLE** DISPLAY Access crosses page boundary.

ArchMsg.Warning.warning_access_crosses_page_boundary_has_fault

TAGS **UNPREDICTABLE** DISPLAY Access that crosses page boundary has fault(s).

ArchMsg.Warning.warning_access_crosses_page_boundary_has_fault_on_both_pages

TAGS **UNPREDICTABLE** DISPLAY Access that crosses page boundary has a fault on each page.
Fields:

ADDR1 unsigned int

FAR address on first page.

ADDR2 unsigned int

FAR address on second page.

FSTATUS1 unsigned int

Fault status on first page.

FSTATUS2 unsigned int

Fault status on second page.

ArchMsg.Warning.warning_access_crosses_page_boundary_spanning_different_memory

TAGS **UNPREDICTABLE** DISPLAY Access at address %{ADDR} crosses page from MEM_TYPE %{MEMTYPE_PAGE1} to %{MEMTYPE_PAGE2}. Fields:

ADDR unsigned int

address of access crossing page.

MEMTYPE_PAGE1 signed int

First Page's memory type (0-Normal, 1-Non-Normal).

MEMTYPE_PAGE2 signed int

Second Page's memory type (0-Normal, 1-Non-Normal).

ArchMsg.Warning.warning_access_wraps_around_memory

TAGS **UNPREDICTABLE** DISPLAY Access wraps around memory in %{WIDTH:d}bit mode.
Fields:

WIDTH unsigned int

Address width.

ArchMsg.Warning.warning_bcr_linking_status

TAGS **UNPREDICTABLE** DISPLAY DBGBCR.{BT2,BT} combination gives unpredictable behaviour.

ArchMsg.Warning.warning_bcr_mask_reserved

TAGS **UNPREDICTABLE** DISPLAY DBGBCR_ELn.MASK is set to a reserved value.

ArchMsg.Warning.warning_bvr_masked_bit_not_zero

TAGS **UNPREDICTABLE** DISPLAY DBGBVRn_EL1.VA contains a bit masked by DBGBCRn_EL1.MASK that is not zero.

ArchMsg.Warning.warning_ccsidr_unimplemented_level

TAGS **UNPREDICTABLE** DISPLAY CCSIDR read while CSSELR %{CSSELR:x} points to an unimplemented cache level. Fields:

CSSELR unsigned int

current effective value of CSSELR.

ArchMsg.Warning.warning_change_to_ns_when_tge_set

TAGS **UNPREDICTABLE** DISPLAY Attempting to change NS to 1 when HCR.TGE = 1.

ArchMsg.Warning.warning_contiguous_bit_check_abort

TAGS **UNPREDICTABLE** DISPLAY An abort occurred whilst attempting to check TLB entry is contiguous at %{ENTRY_ADDR}. Fields:

ENTRY_ADDR unsigned int

address of conflicting TLB entry.

ArchMsg.Warning.warning_contiguous_bit_error

TAGS **UNPREDICTABLE** DISPLAY TLB entry at %{ENTRY_ADDR} was not contiguous with entry at %{CONTIG_ADDR} Contents are %{ENTRY_DATA} but expected %{CONTIG_DATA}. Fields:

CONTIG_ADDR unsigned int

address of the TLB first read, with which this entry is expected to be contiguous.

CONTIG_DATA unsigned int

expected contents based on the entry at CONTIG_ADDR.

ENTRY_ADDR unsigned int

address of conflicting TLB entry.

ENTRY_DATA unsigned int

contents of conflicting TLB entry.

ArchMsg.Warning.warning_contiguous_bit_error_D128

TAGS **UNPREDICTABLE** DISPLAY TLB entry at %{ENTRY_ADDR} was not contiguous with entry at %{CONTIG_ADDR} Contents are %{ENTRY_DATA_HI} %{ENTRY_DATA_LO} but expected %{CONTIG_DATA}. Fields:

CONTIG_ADDR unsigned int

address of the TLB first read, with which this entry is expected to be contiguous.

CONTIG_DATA unsigned int

expected contents based on the entry at CONTIG_ADDR.

ENTRY_ADDR unsigned int

address of conflicting TLB entry.

ENTRY_DATA_HI unsigned int

Higher bit contents of conflicting TLB entry.

ENTRY_DATA_LO unsigned int

Lower bit contents of conflicting TLB entry.

ArchMsg.Warning.warning_cp10_cp11_mismatch

TAGS **UNPREDICTABLE** DISPLAY Unpredictable configuration of CP10 and CP11 controls at EL
%{EL:d}. Fields:

EL unsigned int

EL owning the control.

ArchMsg.Warning.warning_cp10_cp11_reserved_value

TAGS **UNPREDICTABLE** DISPLAY Unpredictable configuration of CPACR CP10 (%{CP10}) or
CP11 (%{CP11}). Fields:

CP10 unsigned int

CP10 access.

CP11 unsigned int

CP11 access.

ArchMsg.Warning.warning_csselr_level_out_of_range

TAGS **UNPREDICTABLE** DISPLAY CSSELR.Level out of range, written value %{LEVEL} with only
%{IMPLEMENTED:d} levels implemented. Fields:

IMPLEMENTED unsigned int

number of cache levels implemented.

LEVEL unsigned int

written value of CSSELR.Level.

TYPE enum

The type of cache selected by CSSELR.InD and TnD.

ArchMsg.Warning.warning_debug_flow_control_bits_not_obeyed

TAGS **UNPREDICTABLE** DISPLAY Flow control bits for %{REGISTER} were not obeyed, causing
%{ERROR}. Fields:

ERROR enum

type of error resulting.

REGISTER enum

register under consideration.

ArchMsg.Warning.warning_debug_register_access_during_reset

TAGS **UNPREDICTABLE** DISPLAY Unpredictable %{IS_WRITE:(read|write)} access to debug register offset %{OFFSET} during reset. Fields:

IS_WRITE unsigned int

Write Not Read.

OFFSET unsigned int

Register Offset.

ArchMsg.Warning.warning_decode_cps_inconsistent_fields

TAGS **UNPREDICTABLE** DISPLAY The mode setting fields in the CPS instruction are inconsistent.

ArchMsg.Warning.warning_decode_invalid_state

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable in current exception-level/security state.

ArchMsg.Warning.warning_default_cacheable_mmu_on

TAGS **UNPREDICTABLE** DISPLAY Default cacheable is enabled (HCR.DC = 1) while MMU is enabled.

ArchMsg.Warning.warning_default_cacheable_vmmu_off

TAGS **UNPREDICTABLE** DISPLAY Default cacheable is enabled (HCR.DC = 1) while HCR.VM = 0.

ArchMsg.Warning.warning_deprecated_wvr_bit_2_set

TAGS **UNPREDICTABLE** DISPLAY DBGWVR%{N:d}_EL1=%{WVR:h} has bit 2 set, deprecated in ARMv8. Fields:

N unsigned int

index.

WVR unsigned int

DBGWVR.

ArchMsg.Warning.warning_eret_while_software_step_active_pending

Missing ISB between setting MDSCR_EL1.SS and ERET TAGS **UNPREDICTABLE**.

ArchMsg.Warning.warning_exclusive_to_non_normal

TAGS **UNPREDICTABLE** DISPLAY Exclusive Access to Non-Normal Memory %{ADDR}. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_exclusive_to_non_writeback

TAGS **UNPREDICTABLE** DISPLAY Exclusive Access to Non-Writeback-Cacheable Memory %{ADDR}. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_execute_from_device_memory

TAGS **UNPREDICTABLE** DISPLAY An attempt was made to execute from Device memory.

ArchMsg.Warning.warning_illegal_cpsr_mode

TAGS **UNPREDICTABLE** DISPLAY Writing an illegal or unimplement mode ({NEW_MODE:h}) to CPSR was unpredictable. Fields:

NEW_MODE unsigned int

New value of CPSR.M.

ArchMsg.Warning.warning_illegal_srs_mode

TAGS **UNPREDICTABLE** DISPLAY Illegal or **UNPREDICTABLE** mode ({MODE:h}) used for SRS instruction. Fields:

MODE unsigned int

mode for Banked SP.

ArchMsg.Warning.warning_implementation_defined_read_debug_register_in_SCS

Read Access to Debug registers from the processor is **IMPLEMENTATION DEFINED** TAGS **IMP_DEF**. Fields:

PPB_OFFSET unsigned int

debug register offset.

ArchMsg.Warning.warning_implementation_defined_sequential_security_transitions_supported

The behavior of sequential instruction fetches that cross from non-secure to secure memory and contain an SG instruction is **CONSTRAINED UNPREDICTABLE** TAGS **UNPREDICTABLE**.

ArchMsg.Warning.warning_implementation_defined_stack_limit_check_supported

It is **IMPLEMENTATION DEFINED** whether stack pointer limit checking is performed for this instruction TAGS **IMP_DEF**. Fields:

ADDRESS unsigned int

Address of instruction.

OPCODE unsigned int

instruction opcode.

ArchMsg.Warning.warning_implementation_defined_taggednonsharedmemorytype

TAGS **IMP_DEF** DISPLAY Memory attribute has Tagged, non shared memory type; it is **IMPLEMENTATION DEFINED** whether the memory location is treated as Tagged or non-Tagged.

ArchMsg.Warning.warning_implementation_defined_write_debug_register_in_SCS

Write Access to Debug registers from the processor is **IMPLEMENTATION DEFINED** TAGS **IMP_DEF**. Fields:

DATA unsigned int

data attempted to be written.

PPB_OFFSET unsigned int

debug register offset.

ArchMsg.Warning.warning_invalid_tcr_granule

TAGS UNPREDICTABLE DISPLAY TCR.TG{%TG_ID:(0|1)} bits have been set to a granule size {%REQUEST:d}K not implemented - using {%SUBSTITUTE:d}K. Fields:

REQUEST signed int

page size requested (or 0 for reserved).

SUBSTITUTE signed int

best guess available page size.

TG_ID bool

bits TG0 or TG1.

ArchMsg.Warning.warning_load_multiple_user_registers_from_user_mode

TAGS UNPREDICTABLE DISPLAY An LDM(user registers) instruction executed from user mode.

ArchMsg.Warning.warning_mte_in_mem_attr_not_supported

TAGS UNPREDICTABLE DISPLAY Memory attribute {%ATTR} has Tagged memory type but Armv8.5-MTE is not implemented. Fields:

ATTR unsigned int

Memory attribute.

ArchMsg.Warning.warning_reserved_breakpoint_state_match

TAGS UNPREDICTABLE DISPLAY {%IS_BREAKPOINT:(Watchpoint|Breakpoint)} programmed with a reserved combination of HMC, SSC and {%IS_BREAKPOINT:(PAC|PMC)}. Fields:

IS_BREAKPOINT unsigned int

Is a breakpoint, not a watchpoint.

ArchMsg.Warning.warning_shareability

TAGS UNPREDICTABLE DISPLAY Unpredictable: combination of the 1st and 2nd stages of translation is Normal Inner Non-Cacheable, Outer Non-Cacheable.

ArchMsg.Warning.warning_software_step_set_while_enabled

TAGS UNPREDICTABLE DISPLAY MDSCR_EL1.SS set to 1 while software step debug exceptions are enabled.

ArchMsg.Warning.warning_thumb_instruction_wraps_around_memory

TAGS UNPREDICTABLE DISPLAY Thumb instruction wraps around memory.

ArchMsg.Warning.warning_ttbr_sbz_bits_are_not_zero

TAGS UNPREDICTABLE DISPLAY {%TTBR}[{%MSB:d}:0] should be zero, but are not. Fields:

MSB unsigned int

x-1.

TTBR enum

which TTBR register.

ArchMsg.Warning.warning_unaligned_address_dbgdtrrx_write

TAGS **UNPREDICTABLE** DISPLAY Unaligned address %{ADDR} in X0/R0 when DBGDTRRX written in memory access mode. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unaligned_address_dbgdtrtx_read

TAGS **UNPREDICTABLE** DISPLAY Unaligned address %{ADDR} in X0/R0 when DBGDTRTX read in memory access mode. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unaligned_to_device

TAGS **UNPREDICTABLE** DISPLAY Unaligned Access to Device Memory %{ADDR}. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unaligned_to_strongly_ordered

TAGS **UNPREDICTABLE** DISPLAY Unaligned Access to Strongly Ordered Memory %{ADDR}. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unknown_sau_rnr

TAGS **UNPREDICTABLE** DISPLAY SAU_RNR was set to an unsupported value.

ArchMsg.Warning.warning_unpredictable_AIRCR_PRIS_and_BFHFNMINs

TAGS **UNPREDICTABLE** DISPLAY The effect of setting both AIRCR.BFHFNMINs and AIRCR.PRIS to 1 is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_AIRCR_VECTCLRACTIVE_when_not_in_debug

TAGS **UNPREDICTABLE** DISPLAY The effect of writing a 1 to AIRCR.VECTCLRACTIVE if the processor is not halted in Debug state is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_AIRCR_VECTRESET_and_SYSRESETREQ

TAGS **UNPREDICTABLE** DISPLAY When the processor is halted in Debug state, if a write to the register writes a 1 to both VECTRESET and SYSRESETREQ, the behavior is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_AIRCR_VECTRESET_when_not_in_debug

TAGS **UNPREDICTABLE** DISPLAY The effect of writing a 1 to AIRCR.VECTRESET if the processor is not halted in Debug state is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_AIRCR_incorrect_VKEY

TAGS **UNPREDICTABLE** DISPLAY The value 0x05FA must be written to AIRCR.VKEY, otherwise the register write is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_EXC_RETURN_Reserved_Bit

EXC_RETURN[23:7] are reserved with the special condition that all bits should be written as one. Values other than all 1s are **UNPREDICTABLE**. TAGS **UNPREDICTABLE**. Fields:

EXC_RETURN unsigned int

Special PC value.

SBO_MASK unsigned int

Which bits Should Be One to not UNPRED.

ArchMsg.Warning.warning_unpredictable_PMEVTYPEPn_ELO_VS_reserved_value

TAGS **UNPREDICTABLE** DISPLAY Unpredictable reserved value of PMEVTYPEn<n>_ELO.VS.

ArchMsg.Warning.warning_unpredictable_change_to_DEMCR_MON_STEP_at_insufficient_priority

TAGS **UNPREDICTABLE** DISPLAY The effect of changing DEMCR.MON_STEP at an execution priority that is lower than the priority of the DebugMonitor exception is **CONSTRAINED UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_change_to_priority_of_active_exception

TAGS **UNPREDICTABLE** DISPLAY Changing the priority of an active exception is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_clear_lspact

TAGS **UNPREDICTABLE** DISPLAY **UNPREDICTABLE** state on context switch. Fields:

DRVEXC_TAKEN unsigned int

Derived exception taken, not pended.

REACHED_LAST_STORE unsigned int

Only the last faulted.

ArchMsg.Warning.warning_unpredictable_dbg_exit_unaligned_dlr

TAGS **UNPREDICTABLE** DISPLAY Debug exit to AArch32 with DLR[0] set to 1. PC[0] can be set to 0 or to DLR[0].

ArchMsg.Warning.warning_unpredictable_exception_catch

TAGS **UNPREDICTABLE** DISPLAY Generation of an exception catch event is unpredictable.

ArchMsg.Warning.warning_unpredictable_exception_return_inconsistent_state

TAGS **UNPREDICTABLE** DISPLAY state on exception return is unpredictable.

ArchMsg.Warning.warning_unpredictable_exception_return_instruction

TAGS **UNPREDICTABLE** DISPLAY exception return instruction is unpredictable.

ArchMsg.Warning.warning_unpredictable_in_debug_state

TAGS **UNPREDICTABLE** DISPLAY Instruction is **UNPREDICTABLE** when executed in debug state.

ArchMsg.Warning.warning_unpredictable_mair_encoding

TAGS **UNPREDICTABLE** DISPLAY Memory attribute %{ATTR} has invalid MAIR encoding; this may lead to unpredictable results. Fields:

ATTR unsigned int

Memory attribute.

ArchMsg.Warning.warning_unpredictable_pmu_counter_access

TAGS **UNPREDICTABLE** DISPLAY Access to PMU counters from non-secure EL0 or EL1 is **UNPREDICTABLE** with MDCR_EL2.HPMN set to 0.

ArchMsg.Warning.warning_unpredictable_prioritization_breakpoint_match_vector_catch

TAGS **UNPREDICTABLE** DISPLAY Two events with the same priority occurred at the same instruction: Address Matching Vector Catch debug event, and Breakpoint debug event. It is constrained unpredictable which is taken.

ArchMsg.Warning.warning_unpredictable_stack_selection

TAGS **UNPREDICTABLE** DISPLAY **UNPREDICTABLE** state on context switch.

ArchMsg.Warning.warning_unpredictable_tsize_out_of_range

TAGS **UNPREDICTABLE** DISPLAY %{TCR}.T%{TTBR:d}SZ (%{TSIZE:d} bits) is out of range. Must be between 25 and %{TSIZE_MAX:d}. Fields:

TCR enum

Which TCR.

TSIZE signed int

Value of TSize.

TSIZE_MAX signed int

The maximum allowed value.

TTBR unsigned int

Which TTBR region.

ArchMsg.Warning.warning_unpredictable_unaligned_pc_as_base_register

TAGS **UNPREDICTABLE** DISPLAY The PC value must be word-aligned, otherwise the behavior of the instruction is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_unaligned_pop_stack

TAGS **UNPREDICTABLE** DISPLAY **CONSTRAINED UNPREDICTABLE** unaligned stack frame %{FRAME_ADDR:x} in exception-return. Fields:

FRAME_ADDR unsigned int

unaligned stack frame address.

ArchMsg.Warning.warning_unpredictable_unaligned_to_non_writeback

TAGS **UNPREDICTABLE** DISPLAY Unaligned Access to Non-Writeback-Cacheable Memory.

ArchMsg.Warning.warning_unpredictable_vmsa_memattrib

TAGS **UNPREDICTABLE** DISPLAY Unpredictable vmsa memory attribute with texcb value programmed to %{TEXCB} when tex remap is %{TEX_REMAP}. Fields:

TEXCB unsigned int

texcb value.

TEX_REMAP bool

use tex remap.

ArchMsg.Warning.warning_unpredictable_vtcr_t0sz_sl0

TAGS **UNPREDICTABLE** DISPLAY The combination of VTCR.TOSZ=%{TOSZ:d} and VTCR.SLO=%{SLO:d} is unpredictable. Fields:

SLO signed int

VTCR.SLO, starting level for stage 2 translation table walks.

TOSZ signed int

VTCR.TOSZ, required input address range.

ArchMsg.Warning.warning_unpredictable_write_DHCSR

TAGS **UNPREDICTABLE** DISPLAY **UNPREDICTABLE** set of bit-changes in DHCSR. Fields:

NEW unsigned int

data attempted to be written.

OLD unsigned int

previous value.

ArchMsg.Warning.warning_unpredictable_write_SHCSR

TAGS **UNPREDICTABLE** DISPLAY **UNPREDICTABLE** setting a pending bit to 1 for an exception with priority >= execution priority.

ArchMsg.Warning.warning_unsupported_access_to_memory_mapped_register

TAGS **UNPREDICTABLE** DISPLAY Access to memory mapped register is unsupported.

ArchMsg.Warning.warning_user_jmcr_access

TAGS **UNPREDICTABLE** DISPLAY User %{WRITE:(write|read)} access to JMCR is unpredictable. Fields:

WRITE unsigned int

access is write.

ArchMsg.Warning.warning_wcr_mask_and_bas

TAGS **UNPREDICTABLE** DISPLAY DBGWCR_EL1.MASK is non-zero and BAS is not 0xff.

ArchMsg.Warning.warning_wcr_mask_reserved

TAGS **UNPREDICTABLE** DISPLAY DBGWCR_ELn.MASK is set to a reserved value.

ArchMsg.Warning.warning_wcr_non_configuous_bas

TAGS **UNPREDICTABLE** DISPLAY DBGWCR_EL1.BAS has non-contiguous set of ones.

ArchMsg.Warning.warning_wvr_masked_bit_not_zero

TAGS **UNPREDICTABLE** DISPLAY DBGWVRn_EL1.VA contains a bit masked by DBGWCRn_EL1.MASK that is not zero.

ArchMsg.Why.why_branch_target_exception

A branch target exception has occurred. DISPLAY branch target exception: target pc=%{PC}, target opcode=%{OPCODE}. PRIMARY KEY PC. Fields:

BTYPE unsigned int

pstate.BTYPE at time branch taken.

OPCODE unsigned int

opcode at branch target.

PC unsigned int

Virtual Address of branch target.

ArchMsg.Why.why_illegal_state

An illegal mode change has occurred. DISPLAY illegal mode change: %{MESSAGE}. Fields:

MESSAGE string

The message.

ArchMsg.warning_invalid_interval_counter_reload_value

Invalid interval counter reload value in PMSIRR_EL1.INTERVAL: software should set this to a value greater than or equal to the minimum indicated by PMSIDR_EL1.Interval.

ArchMsg.warning_invalid_spe_buffer_write_pointer

Invalid value programmed in PMBPTR_EL1.PTR or PMBLIMITR_EL1.LIMIT: Current write pointer must be at least one sample record below the write limit pointer and PMBPTR_EL1.PTR[63:56] must be equal to PMBLIMITR_EL1.LIMIT[63:56].

BRANCH_MISPREDICT

Simulating branch mispredict. Fields:

PC unsigned int

Origin address (or 0 if unavailable).

BRA_DIR

Direct branches, to immediate address. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISSET enum

The instruction set of the branch instruction.

IS_COND bool

Indicates if this is a conditional branch.

IS_HINTED bool

Whether the conditional instruction is hinted.

IS_ISB bool

Whether the instruction is ISB.

IS_LINK bool

Indicates if this is a branch with link.

PC unsigned int

The address of the branch instruction.

TARGET_ISET enum

The instruction set after the branch.

TARGET_PC unsigned int

The address the instruction branches to.

BRA_INDIR

Indirect branches, perhaps to a register. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISET enum

The instruction set of the branch instruction.

IS_COND bool

Indicates if this is a conditional branch.

IS_HINTED bool

Whether the conditional instruction is hinted.

IS_ISB bool

Whether the instruction is ISB.

IS_LINK bool

Indicates if this is a branch with link.

PC unsigned int

The address of the branch instruction.

TARGET_ISET enum

The instruction set after the branch.

TARGET_PC unsigned int

The address the instruction branches to.

BRA_INDIR_CCFAIL

Indirect branches fail, perhaps to a register.

BRA_RET_CCFAIL

Branch return fail, perhaps to a register.

CACHE_MAINTENANCE_OP

Cache Maintenance Operation. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

DATA unsigned int

Specified MVA or Set/way.

FUNCTION enum

Clean / Invalidate.

SCOPE enum

Affected region.

SIDE enum

Inst / Data.

CCFAIL

Conditional instruction condition check fail. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CCFAIL_UNC

Unconditional instruction condition check fail. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CCPASS

Conditional instruction condition check pass. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CCPASS_UNC

Unconditional instruction condition check pass. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CODE_CACHE_MAINT

Code cache maintenance. Fields:

END_INCL unsigned int

Inclusive end address for VA or PA requests.

NS bool

Non-secure world for PA.

START unsigned int

Start address for VA or PA requests.

TYPE enum

Request type.

COMPILE_BLOCK_END

Last instruction of basic block translated. Fields:

VADDR unsigned int

Address of next instruction after this basic block.

COMPILE_BLOCK_START

First instruction of basic block translated. Fields:

VADDR unsigned int

Address of first instruction in basic block.

COMPILE_INST

ARM instruction compiled. Fields:

DISASS string

Disassembly of the instruction.

ISSET enum

The instruction set of this instruction.

ITSTATE unsigned int

The ITSTATE current for the instruction.

OPCODE unsigned int

The opcode of the instruction.

PC unsigned int

The address of the instruction.

SIZE unsigned int

The size of the instruction in bytes.

CONTEXTIDR

Context ID Register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

NS bool

Secure or nonsecure banked register is accessed.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The new value written.

CONTEXT_SYNC

Called for every context synchronization event. Fields:

ARCH_CSE bool

Architectural Context Synchronization Event.

PC unsigned int

The address of the synchronization instruction.

SYSREG_SYNC bool

System register synchronization event.

CORE_ENDIAN

Core BE8 Big-Endian state changed. Fields:

BE8 bool

Core BE8 Big-Endian state.

CORE_INFO

Static processor attributes. Only triggered by a call to DumpState(). Fields:

ARCH_PROFILE enum

The architecture profile of the core.

CLUSTER_ID unsigned int

The cluster ID of this processor.

CORE_NUM unsigned int

The number of this core in an MP processor.

DETAILED_CORE_FEATURES enum

Detailed core features as a bitfield. Enum values are bitmasks to extract the info from the field.

FPU_VERSION enum

The VFP version implemented by the core.

MEM_ARCH enum

The memory architecture of the core.

MIDR_EL1 unsigned int

MIDR_EL1 value.

NUM_CORES unsigned int

The number of cores in this MP processor.

QUANTUM_SIZE unsigned int

The default quantum size of the core (in ticks).

SECURITY_FEATURES bool

Does the core have security features?.

CORE_LOADS

Processor load accesses. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ACQREL enum

Is this an acquire/release.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DATA unsigned int

The data read or written.

ELEMENT_SIZE unsigned int

Width of each element.

ENDIANNESS enum

Byte ordering of this access.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

LOCK enum

Normal, exclusive or locked access.

MECID unsigned int

Memory Encryption Context ID, if applicable.

MECID2 unsigned int

Memory Encryption Context ID of the second page, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PADDR_TAG unsigned int

If tag checked, the physical address tag for the access.

PAGE_SIZE unsigned int

Page size as $\log_2(\text{size in bytes})$.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK, 2=Exclusive Failed, 3=Unsupported.

SIZE unsigned int

Width of the access in bytes. Only required if DATA is not traced.

TAG_CHECKED bool

Is this access tag checked.

TAG_CHECKED2 bool

Is the second page access tag checked.

TRANS bool

Is this a translated access.

VADDR unsigned int

The virtual address of the access.

CORE_REGS

Changes of the core registers R0 to R14. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number, 0 to 14.

MODE enum

Bank of the register accessed.

OLD_VALUE unsigned int

The old value overwritten.

PHYS_ID enum

The physical register accessed.

VALUE unsigned int

The new value written to the register.

CORE_REGS64

Changes of the core registers X0..X30, SP_ELn. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ID enum

The register number, 0-30 for X0-X30, >=32 for SP_ELn.

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

CORE_REGS64_READ

General purpose AArch64 register has been read X0..X30, SP_ELn. Fields:

ID enum

The register number, 0-30 for X0-X30, >=32 for SP_ELn.

VALUE unsigned int

The value read from the register.

CORE_STALLED_TRANSITION

Detect when a core is prevented from progressing PChannel states. Fields:

CURRENT_STATE enum

Current power mode of core.

REASON string

The reason a transition was stalled, and what must be done to unblock it.

TARGET_STATE enum

Desired power mode of core after transition.

CORE_STORES

Processor store accesses. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ACQREL enum

Is this an acquire/release.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DATA unsigned int

The data read or written.

ELEMENT_SIZE unsigned int

Width of each element.

ENDIANNESS enum

Byte ordering of this access.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

LOCK enum

Normal, exclusive or locked access.

MECID unsigned int

Memory Encryption Context ID, if applicable.

MECID2 unsigned int

Memory Encryption Context ID of the second page, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PADDR_TAG unsigned int

If tag checked, the physical address tag for the access.

PAGE_SIZE unsigned int

Page size as $\log_2(\text{size in bytes})$.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK, 2=Exclusive Failed, 3=Unsupported.

SIZE unsigned int

Width of the access in bytes. Only required if DATA is not traced.

TAG_CHECKED bool

Is this access tag checked.

TAG_CHECKED2 bool

Is the second page access tag checked.

TRANS bool

Is this a translated access.

VADDR unsigned int

The virtual address of the access.

CP14_READ

System Coprocessor register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP14 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc1 unsigned int

opcode 1.

opc2 unsigned int

opcode 2.

CP14_WRITE

System Coprocessor register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP14 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc1 unsigned int

opcode 1.

opc2 unsigned int
opcode 2.

CP15_READ

System Coprocessor register read. Fields:

CORE_NUM unsigned int
Core number in a multi processor.

CRm unsigned int
CRm.

CRn unsigned int
CRn.

NS bool
Secure or nonsecure banked register is accessed.

REG_NAME string
Name of the CP15 register accessed.

SECURITY_STATE enum
The current security state.

UNDEF bool
The register accessed is undefined.

VALUE unsigned int
The value read.

opc1 unsigned int
opcode 1.

opc2 unsigned int
opcode 2.

CP15_READ64

System Coprocessor register read. Fields:

CORE_NUM unsigned int
Core number in a multi processor.

CRm unsigned int
CRm.

NS bool
Secure or nonsecure banked register is accessed.

REG_NAME string
Name of the CP15 register accessed.

SECURITY_STATE enum
The current security state.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc unsigned int

opcode 1.

CP15_WRITE

System Control Coprocessor register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP15 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc1 unsigned int

opcode 1.

opc2 unsigned int

opcode 2.

CP15_WRITE64

System Coprocessor register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP15 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc unsigned int

opcode 1.

CPSR

CPSR change. (In AArch64 PSTATE represented in SPSR format). Fields:

CORE_NUM unsigned int

Core number in a multi processor.

OLD_VALUE unsigned int

The old CPSR value.

UNKNOWN unsigned int

Bits within the register that have unknown value.

VALUE unsigned int

The new CPSR value.

CRYPTO_SPEC

Every crypto instruction speculatively executed.

CorePowerStateChange

Core power-up state has changed. DISPLAY Core powered %{PU:(up|down)}. Fields:

PU bool

Whether the core powered up.

DATA_CACHE_SET_ALLOC_TAGS

Set allocation tags from DC GVA instruction. Fields:

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate

when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

NSDESC unsigned int

The security state of the access.

PADDR unsigned int

The physical address of the access.

PAS enum

The physical address space of the page.

RESPONSE enum

0=Aborted, 1=OK.

SIZE unsigned int

The size of memory in bytes.

TAG unsigned int

The value of the tag being set.

TAGGING_TYPE enum

Type of Memory tagging (Physical or Virtual).

TAG_VADDR unsigned int

The virtual address of the Virtual Tag if Virtual tagging is used; otherwise 0.

VADDR unsigned int

Virtual address used for setting allocation tags.

DATA_CACHE_ZERO

Zero and invalidate cache line from DC ZVA instruction. Fields:

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

NSDESC unsigned int

The security state of the access.

PADDR unsigned int

The physical address of the access.

PADDR_TAG unsigned int

If tag checked, the physical address tag for the access.

PAS enum

The physical address space of the page.

RESPONSE enum

0=Aborted, 1=OK.

SIZE unsigned int

The size of zero'd memory in bytes.

TAG_CHECKED bool

Is this access tag checked.

VADDR unsigned int

Virtual address of the zero'd cache line.

DEBUG_EVENT

Hardware debug support event. Fields:

EVENT enum

Description of event.

VALUE unsigned int

data value.

DEBUG_TARGET_ISET

Hardware debug target ISA. Fields:

TARGET_ISET enum

The target instruction set of the debug state.

DMI_ALLOCATE

DMI allocated. Fields:

ACCESS_DIRECTION enum

READ / WRITE.

PADDR unsigned int

The physical address of the DMI.

PTR unsigned int

DMI range pointer.

SIDE enum

DSIDE / ISIDE.

SIZE unsigned int

The size of DMI.

VADDR unsigned int

The virtual address of the DMI..

DMI_HIT

DMI hit, access used DMI (only fire for small DMI regions i.e. less than a full page). Fields:

ACCESS_DIRECTION enum

READ / WRITE.

PADDR unsigned int

The physical address of the DMI.

SIDE enum

DSIDE / ISIDE.

SIZE unsigned int

The size of the DMI.

VADDR unsigned int

The virtual address of the DMI..

DMI_REVOKE

DMI revoked. Fields:

ACCESS_DIRECTION enum

READ / WRITE.

PADDR unsigned int

The physical address of the DMI.

SIDE enum

DSIDE / ISIDE.

SIZE unsigned int

The size of DMI.

VADDR unsigned int

The virtual address of the DMI..

END_COMPILE

Compilation end. Fields:

END_OF_PAGE_COUNT unsigned int

Number of basic blocks exited due to page boundary since START_COMPILE.

FETCHFAIL_COUNT unsigned int

Number of basic blocks exited due to ifetch failure START_COMPILE.

INST_COUNT unsigned int

Number of instructions compiled since START_COMPILE.

NONSEQ_COUNT unsigned int

Number of basic blocks exited due to non-sequential instructions since START_COMPILE.

PAGE_STRADDLE_COUNT unsigned int

Number of basic blocks exited due to unaligned instructions crossing page since START_COMPILE.

ESB

Error Synchronization Barrier hit. Fields:

ASYNC bool

ESB is async.

IMPLICIT bool

ESB is implicit.

EXCEPTION

Exception taken. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ESR unsigned int

The value assigned to ESR when exception is taken.

ESR_EC unsigned int

The Value assigned to ESR.EC when exception is taken.

IS_PHYSICAL bool

Physical or Virtual exception.

IS_TRAP bool

Exception coming from a trap.

LR unsigned int

The value assigned to the link register.

PC unsigned int

The location where the exception occurred.

PREFERRED_RETURN unsigned int

The preferred return address for the exception.

TAKEN_LOCALLY bool

Whether exception is taken locally(same exception level).

TARGET_ISET enum

The instruction set of the exception handler code.

TARGET_PC unsigned int

The address the exception branches to.

VECTOR enum

The exception vector.

VECTOR_OFFSET unsigned int

The offset of an exception vector from the base address.

EXCEPTION_END

Every exception completed.

EXCEPTION_INFO

Trace human-readable reason of the current exception for data/instruction aborts, floating point exceptions and unknown exceptions. Fields:

DESCRIPTION string

Reason for current exception.

ESR_EC unsigned int

The value of ESR.EC when exception is taken.

EXCEPTION_RAISE

Every exception raised.

EXCEPTION_RETURN

Branches on leaving exception. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISET enum

The instruction set of the branch instruction.

IS_COND bool

Indicates if this is a conditional branch.

IS_HINTED bool

Whether the conditional instruction is hinted.

IS_ISB bool

Whether the instruction is ISB.

IS_LINK bool

Indicates if this is a branch with link.

PC unsigned int

The address of the branch instruction.

TARGET_ISET enum

The instruction set after the branch.

TARGET_PC unsigned int

The address the instruction branches to.

EXCEPTION_RETURN_PREBRANCH

Exception return event that must occur before PC and processor state have updated.

EXCEPTION_START

Every exception started.

FIQ_TAKEN

FIQ taken exception.

FP_EXCEPTIONS_SIGNALLED

Trace which FP exceptions were signalled as a part of the FP operations performed by an instruction. Fields:

FP_EXCEPTIONS_SIGNALLED_MASK unsigned int

Indicates which FP exceptions were signalled by an instruction. This uses the same encoding as the cumulative floating-point exception bitfields in FPSR.

FP_STATE

Floating point state. Fields:

FPSR unsigned int

FPSR register value.

MASK unsigned int

The mask indicating updated cumulative bits.

FREQ_CHANGED

Core frequency changed. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The instruction count of this core.

NEW_FREQ signed int

The new frequency of this core (expressed in Hz).

OLD_FREQ signed int

The old frequency of this core (expressed in Hz).

HLT

HLT instruction occurred. Fields:

IMM16 unsigned int

The 16-bit immediate value encoded in HLT instruction.

INST

Every instruction executed. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DEBUG_STATE bool

The instruction is executed in debug state.

DISASS string

Disassembly of instruction.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISSET **enum**

The current instruction set.

ITSTATE **unsigned int**

The current ITSTATE.

LOCAL_TIME **unsigned int**

The core's local time, relative to the current quantum.

MODE **enum**

The mode the core is in.

NS **unsigned int**

The core's non-secure bit.

NSDESC **unsigned int**

The physical address non-secure bit.

NSDESC2 **unsigned int**

The second page physical address non-secure bit.

OPCODE **unsigned int**

The opcode of the instruction.

PADDR **unsigned int**

The physical address of the instruction.

PADDR2 **unsigned int**

If different from PADDR, the physical address of the second page of the instruction.

PAS **enum**

The physical address space of the page.

PAS2 **enum**

The physical address space of the page.

PC **unsigned int**

The address of the instruction.

SECURITY_STATE **enum**

The core's security state.

SIZE **unsigned int**

The size of the instruction in bytes.

INST_END

Every instruction completed.

INST_START

Every instruction started. Fields:

DEBUG_STATE **bool**

The instruction is executed in debug state.

ISSET enum

The current instruction set.

MODE enum

The mode the core is in.

NS enum

The current Secure State.

PC unsigned int

The address of the conditional instruction.

SECURITY_STATE enum

The current Security State.

INST_STRADDLE

Instruction straddles boundary. Fields:

DEBUG_STATE bool

The instruction is executed in debug state.

ISSET enum

The current instruction set.

MODE enum

The mode the core is in.

NS enum

The current Secure State.

PC unsigned int

The address of the conditional instruction.

SECURITY_STATE enum

The current Security State.

IRQ_TAKEN

IRQ taken exception.

ISB

ISB instruction occurred. Fields:

PC unsigned int

The address of the ISB instruction.

LOAD_MULTIPLE_REGS_LIST

List of destination registers (not source) of a multiple registers load instruction. Fields:

REG string

Register to add into the list.

LOCAL_MONITOR

Local monitor activity. Fields:

PADDR unsigned int

Local Monitor Address.

State enum

State of the monitor (Open/Exclusive).

MAPPING_CREATE

A memory mapping has been created. This is an implementation detail of the model and so may change without notice. Fields:

ASID unsigned int

The ASID used during translation, or 0 if no ASID was used.

IS_VALID bool

Whether the mapping is valid or not.

PADDR unsigned int

Base physical address of region, or 0 if not known.

SIDE enum

I-side or D-side.

SIZE unsigned int

Size of the page, or 0 if not known.

VADDR unsigned int

Base virtual address of region.

VMID unsigned int

The VMID used during translation, or 0 if no VMID was used.

MAPPING_DESTROY

A memory mapping has been destroyed. This is an implementation detail of the model and so may change without notice. Fields:

ASID unsigned int

The ASID used during translation.

CAUSE enum

The cause of invalidation.

PADDR unsigned int

Base physical address of page.

SIDE enum

I-side or D-side.

SIZE unsigned int

Size of the page.

VADDR unsigned int

Base virtual address of page.

VMID unsigned int

The VMID used during translation.

MEMMAP_DEBUG_READ

Memory mapped reads to the debug registers. Fields:

ADDR unsigned int

address.

CORE_NUM unsigned int

Core number in a multi processor.

EXT bool

Whether access is from an external device (such as the DAP).

NS enum

Secure state of the access.

REG_NAME string

Name of the debug register accessed.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

MEMMAP_DEBUG_WRITE

Memory mapped writes to the debug registers. Fields:

ADDR unsigned int

address.

CORE_NUM unsigned int

Core number in a multi processor.

EXT bool

Whether access is from an external device (such as the DAP).

NS enum

Secure state of the access.

REG_NAME string

Name of the debug register accessed.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The new value written.

MEMTAG_LOADS

Allocation tag memory read; when attributes and system configuration allow tag accesses.

Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

NUM_GRANULES unsigned int

Number of tags to be Loaded/stored.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK.

TAGGING_TYPE enum

Type of Memory tagging (Physical or Virtual).

TAGS unsigned int

The tags read or written. Tags for subsequent granules would be found at subsequent bytes in this field (One byte for each tag).

TAG_VADDR unsigned int

The virtual address of the Virtual Tag if Virtual tagging is used; otherwise 0.

VADDR unsigned int

The virtual address of the access.

MEMTAG_LOAD_INST

Allocation tag load instruction; regardless of tags being actually accessed. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

NUM_GRANULES unsigned int

Number of tags to be Loaded/stored.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK.

TAGGING_TYPE enum

Type of Memory tagging (Physical or Virtual).

TAGS unsigned int

The tags read or written. Tags for subsequent granules would be found at subsequent bytes in this field (One byte for each tag).

TAG_VADDR unsigned int

The virtual address of the Virtual Tag if Virtual tagging is used; otherwise 0.

VADDR unsigned int

The virtual address of the access.

MEMTAG_STORES

Allocation tag memory write; when attributes and system configuration allow tag accesses.

Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

NUM_GRANULES unsigned int

Number of tags to be Loaded/stored.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK.

TAGGING_TYPE enum

Type of Memory tagging (Physical or Virtual).

TAGS unsigned int

The tags read or written. Tags for subsequent granules would be found at subsequent bytes in this field (One byte for each tag).

TAG_VADDR unsigned int

The virtual address of the Virtual Tag if Virtual tagging is used; otherwise 0.

VADDR unsigned int

The virtual address of the access.

MEMTAG_STORE_INST

Allocation tag store instruction; regardless of tags being actually accessed. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

NUM_GRANULES unsigned int

Number of tags to be Loaded/stored.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK.

TAGGING_TYPE enum

Type of Memory tagging (Physical or Virtual).

TAGS unsigned int

The tags read or written. Tags for subsequent granules would be found at subsequent bytes in this field (One byte for each tag).

TAG_VADDR unsigned int

The virtual address of the Virtual Tag if Virtual tagging is used; otherwise 0.

VADDR unsigned int

The virtual address of the access.

MMU_D128_TTB_READ

This event is triggered by reads caused by a D128 translation table walk. Fields:

ABORTED bool

Was the walk successful.

DATA_HIGH unsigned int

The data upper bits read.

DATA_LOW unsigned int

The data lower bits read.

IPA unsigned int

For stage 1, the IPA of the read.

LEVEL unsigned int

Translation table level.

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

OUTADDR unsigned int

The output physical address.

PADDR unsigned int

The physical address of the read.

PASmem enum

Physical Address Space of the accessed memory.

PASreq enum

Physical Address Space of the requested lookup.

REGIME_EL enum

Entry matches in this translation regime.

REQUESTER enum

The requester of this table walk.

SIDE enum

Inst / Data.

STAGE unsigned int

Translation stage.

MMU_D128_TTB_WRITE

This event is triggered by writes caused by a D128 translation table walk. Fields:

ABORTED bool

Was the walk successful.

DATA_HIGH unsigned int

The data upper bits written.

DATA_LOW unsigned int

The data lower bits written.

IPA unsigned int

For stage 1, the IPA of the write.

IS_HACDBS_ACCESS bool

If the access was due to HACDBS_CLEAN.

LEVEL unsigned int

Translation table level.

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

OUTADDR unsigned int

The output physical address.

PADDR unsigned int

The physical address of the write.

PASmem enum

Physical Address Space of the accessed memory.

PASreq enum

Physical Address Space of the requested lookup.

REGIME_EL enum

Entry matches in this translation regime.

SIDE enum

Inst / Data.

STAGE unsigned int

Translation stage.

MMU_TRANS

Address translation information. Fields:

ASID unsigned int

Address space identifier.

CORE_NUM unsigned int

Core number in a multi processor.

Hyp bool

Entry matches in Hyp state only.

INNERCACHE_RA bool

Is the inner cache allocate on read.

INNERCACHE_TRANSIENT bool

Is the inner write-through transient.

INNERCACHE_TYPE enum

Inner Caching scheme (NC/MB/WA).

INNERCACHE_WA bool

Is the inner cache allocate on write.

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MEMTYPE enum

Memory type.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC enum

Is secure side supposed to access secure or nonsecure memory.

OUTERCACHE_RA bool

Is the outer cache allocate on read.

OUTERCACHE_TRANSIENT bool

Is the outer write-through transient.

OUTERCACHE_TYPE enum

Outer Caching scheme (NC/MB/WA).

OUTERCACHE_WA bool

Is the outer cache allocate on write.

PADDR unsigned int

Physical address of the access.

PAGESIZE unsigned int

Page size as $\log_2(\text{size})$.

PAS enum

PAS supposed to be accessed is secure or nonsecure memory or Root/Realm if RME is implemented.

SH enum

Shareability.

SIDE enum

Inst / Data.

STAGE1_PERM unsigned int

Stage 1 permissions, [5:3] Privileged access (XWR) [2:0] User access (XWR).

STAGE2_PERM unsigned int

Stage 2 permission mask [5:3] Privileged access (XWR) [2:0] User access (XWR).

VADDR unsigned int

Virtual address of the access.

VMID unsigned int

Virtual machine identifier.

XS bool

XS Attribute.

nG enum

Flag indicating whether ASID will be matched.

MMU_TTB_READ

This event is triggered by reads caused by a translation table walk. Fields:

ABORTED bool

Was the walk successful.

DATA unsigned int

The data read.

DESC_KIND enum

The encoding scheme of the descriptors used for this walk.

IPA unsigned int

For stage 1, the IPA of the read.

LEVEL unsigned int

Translation table level.

LPAE bool

Is this for an LPAE translation (*DEPRECATED*, use DESC_KIND).

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSmem bool

Non secure memory access.

NSreq bool

Non secure request.

OUTADDR unsigned int

The output physical address.

PADDR unsigned int

The physical address of the read.

PASmem enum

Physical Address Space of the accessed memory.

PASreq enum

Physical Address Space of the requested lookup.

REGIME_EL enum

Entry matches in this translation regime.

REQUESTER enum

The requester of this table walk.

SIDE enum

Inst / Data.

STAGE unsigned int

Translation stage.

MMU_TTB_WRITE

This event is triggered by writes caused by a translation table walk. Fields:

ABORTED bool

Was the walk successful.

DATA unsigned int

The data written.

DESC_KIND enum

The encoding scheme of the descriptors used for this walk.

IPA unsigned int

For stage 1, the IPA of the write.

IS_HACDBS_ACCESS bool

If the access was due to HACDBS_CLEAN.

LEVEL unsigned int

Translation table level.

LPAE bool

Is this for an LPAE translation (*DEPRECATED*, use DESC_KIND).

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSmem bool

Non secure memory access.

NSreq bool

Non secure request.

OUTADDR unsigned int

The output physical address.

PADDR unsigned int

The physical address of the write.

PASmem enum

Physical Address Space of the accessed memory.

PASreq enum

Physical Address Space of the requested lookup.

REGIME_EL enum

Entry matches in this translation regime.

SIDE enum

Inst / Data.

STAGE unsigned int

Translation stage.

MODE_CHANGE

Mode change. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

MODE enum

The new mode.

NON_SECURE enum

The core's new non-secure bit.

OLD_MODE enum

The old mode.

SECURITY_STATE enum

The new security state.

NOT_TAG_CHECKED_INST

MTE Tag not checked for current instruction.

NV2_CORE_128_LOADS

Processor load accesses for nested virtualization of v9.4 128-bit registers. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ACQREL enum

Is this an acquire/release.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DATA unsigned int

The data read or written.

ELEMENT_SIZE unsigned int

Width of each element.

ENDIANNESS enum

Byte ordering of this access.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

LOCK enum

Normal, exclusive or locked access.

MECID unsigned int

Memory Encryption Context ID, if applicable.

MECID2 unsigned int

Memory Encryption Context ID of the second page, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PADDR_TAG unsigned int

If tag checked, the physical address tag for the access.

PAGE_SIZE unsigned int

Page size as $\log_2(\text{size in bytes})$.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

REGNUM_OPERAND unsigned int

Register number corresponding to the instruction operand.

RESPONSE enum

0=Aborted, 1=OK, 2=Exclusive Failed, 3=Unsupported.

SIZE unsigned int

Width of the access in bytes. Only required if DATA is not traced.

TAG_CHECKED bool

Is this access tag checked.

TAG_CHECKED2 bool

Is the second page access tag checked.

TRANS bool

Is this a translated access.

VADDR unsigned int

The virtual address of the access.

NV2_CORE_128_STORES

Processor store accesses for nested virtualization of v9.4 128-bit registers. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ACQREL enum

Is this an acquire/release.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DATA unsigned int

The data read or written.

ELEMENT_SIZE unsigned int

Width of each element.

ENDIANNESS enum

Byte ordering of this access.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

LOCK enum

Normal, exclusive or locked access.

MECID unsigned int

Memory Encryption Context ID, if applicable.

MECID2 unsigned int

Memory Encryption Context ID of the second page, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PADDR_TAG unsigned int

If tag checked, the physical address tag for the access.

PAGE_SIZE unsigned int

Page size as $\log_2(\text{size in bytes})$.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

REGNUM_OPERAND unsigned int

Register number corresponding to the instruction operand.

RESPONSE enum

0=Aborted, 1=OK, 2=Exclusive Failed, 3=Unsupported.

SIZE unsigned int

Width of the access in bytes. Only required if DATA is not traced.

TAG_CHECKED bool

Is this access tag checked.

TAG_CHECKED2 bool

Is the second page access tag checked.

TRANS bool

Is this a translated access.

VADDR unsigned int

The virtual address of the access.

NV2_CORE_LOADS

Processor load accesses for v8.4 nested virtualization. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ACQREL enum

Is this an acquire/release.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DATA unsigned int

The data read or written.

ELEMENT_SIZE unsigned int

Width of each element.

ENDIANNESS enum

Byte ordering of this access.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

LOCK enum

Normal, exclusive or locked access.

MECID unsigned int

Memory Encryption Context ID, if applicable.

MECID2 unsigned int

Memory Encryption Context ID of the second page, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PADDR_TAG unsigned int

If tag checked, the physical address tag for the access.

PAGE_SIZE unsigned int

Page size as $\log_2(\text{size in bytes})$.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

REGNUM_OPERAND unsigned int

Register number corresponding to the instruction operand.

RESPONSE enum

0=Aborted, 1=OK, 2=Exclusive Failed, 3=Unsupported.

SIZE unsigned int

Width of the access in bytes. Only required if DATA is not traced.

TAG_CHECKED bool

Is this access tag checked.

TAG_CHECKED2 bool

Is the second page access tag checked.

TRANS bool

Is this a translated access.

VADDR unsigned int

The virtual address of the access.

NV2_CORE_STORES

Processor store accesses for v8.4 nested virtualization. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ACQREL enum

Is this an acquire/release.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DATA unsigned int

The data read or written.

ELEMENT_SIZE unsigned int

Width of each element.

ENDIANNESS enum

Byte ordering of this access.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

LOCK enum

Normal, exclusive or locked access.

MECID unsigned int

Memory Encryption Context ID, if applicable.

MECID2 unsigned int

Memory Encryption Context ID of the second page, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PADDR_TAG unsigned int

If tag checked, the physical address tag for the access.

PAGE_SIZE unsigned int

Page size as $\log_2(\text{size in bytes})$.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

REGNUM_OPERAND unsigned int

Register number corresponding to the instruction operand.

RESPONSE enum

0=Aborted, 1=OK, 2=Exclusive Failed, 3=Unsupported.

SIZE unsigned int

Width of the access in bytes. Only required if DATA is not traced.

TAG_CHECKED bool

Is this access tag checked.

TAG_CHECKED2 bool

Is the second page access tag checked.

TRANS bool

Is this a translated access.

VADDR unsigned int

The virtual address of the access.

PERIODIC

Called for every quantum. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The instruction count of this CPU.

PC unsigned int

The address of the next instruction to be executed on this CPU.

PMU_COUNTER_OVERFLOW

PMU counter overflow. Fields:

COUNTER_GROUP enum

Counter group.

COUNTER_RANGE enum

Counter range.

EVENT_ID unsigned int

Event Identifier, field is valid only for Event counter group.

INDEX unsigned int

Counter index (as selected by PMSELR), field is valid only for Event/Cycle counter groups.

INTERRUPT bool

Is interrupt enabled on overflow for this counter.

POINTER_AUTH

Called when execution hits a pointer authentication check. Fields:

AUTH_STATE bool

Authentication result.

KEY_USED enum

Pointer authentication key.

POTINDEX_UPDATE

Value of POT index for current pc changed. Fields:

NEW_POTINDEX unsigned int

New POTIndex.

OLD_POTINDEX unsigned int

Old POTIndex.

PREFETCH_MEMORY64

Prefetch from PRFM or PRFUM instructions. Fields:

PRFOP enum

Prefetch hint.

VADDR unsigned int

Virtual address of the location that should be prefetched.

PRELOAD64_REQUEST

Preload requests from PRFM or PRFUM instructions. Fields:

PRFOP enum

Prefetch hint.

VADDR unsigned int

Virtual address of the location that should be prefetched.

PRELOAD_DATA

Data preload from PLD instruction. Fields:

VADDR unsigned int

Virtual address of the data that should be preloaded.

PRELOAD_INST

Instruction preload from PLI instruction. Fields:

VADDR unsigned int

Virtual address of the instruction that should be preloaded.

PRE_CORE_LOAD

Trace just before a core load. Fields:

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

PRE_CORE_STORE

Trace just before a core store. Fields:

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

PRE_TTB_READ

This event is triggered before reads caused by a translation table walk. Fields:

ADDR unsigned int

The physical address of the read.

PSTATE

PSTATE fields update. Fields:

ALLINT unsigned int

All interrupt mask bit.

BTYPE unsigned int

Branch target identification bit.

CORE_NUM unsigned int

Core number in a multi processor.

DAIF unsigned int

Exception mask bits.

DIT unsigned int

Data Independent Timing bit.

EL unsigned int

Current Exception level field.

EXLOCK unsigned int

Exception return state lock bit.

IL unsigned int

Illegal Execution state bit.

NZCV unsigned int

Condition flags.

PACM unsigned int

PACM bit.

PAN unsigned int

Privileged Access Never state bit.

PM unsigned int

PMU exception mask bit.

PPEND unsigned int

PMU exception pending bit.

SM unsigned int

StreamingMode status bit.

SP unsigned int

Stack pointer register selection bit.

SS unsigned int

Software Step bit.

SSBS unsigned int

Speculative Store Bypass Safe bit.

TCO unsigned int

Tag Check Override bit.

TOUCHED unsigned int

Mask of the fields that have been touched since this trace has last been emitted. Note that touched does not necessarily mean the value changed. The index of the field is the bit index in the mask plus the index of this field plus one.

UAO unsigned int

User Access Override bit.

UINJ unsigned int

Undefined Instruction exceptions bit.

ZA unsigned int

ZA storage and ZT0 register enable bit.

nRW unsigned int

Current Execution state bit.

PSTATE_PACTIVE_OPMODE_TRANSITION

Trace active and Opmode mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Operating mode after the transition.

OLD_STATE enum

Operating mode before the transition.

PSTATE_PACTIVE_OPMODE_TRANSITION_DENIED

Trace the denied active and Opmode mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Operating mode after the transition.

OLD_STATE enum

Operating mode before the transition.

REASON string

Reason for the transition being denied.

PSTATE_PACTIVE_POWER_TRANSITION

Trace active and power mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Power mode after the transition.

OLD_STATE enum

Power mode before the transition.

PSTATE_PACTIVE_POWER_TRANSITION_DENIED

Trace the denied active and power mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Power mode after the transition.

OLD_STATE enum

Power mode before the transition.

REASON string

Reason for the transition being denied.

PSTATE_PACTIVE_THREAD_OPMODE_TRANSITION

Trace pactive and Opmode mode transitions for threads on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Operating mode after the transition.

OLD_STATE enum

Operating mode before the transition.

QUANTUM_START

Called at the start of every quantum.

RANGE_PRELOAD_REQUEST

Preload requests from RPRFM. Fields:

COUNT unsigned int

Number of blocks of data to access.

LENGTH unsigned int

Number of contiguous bytes to be accessed in each block.

OP unsigned int

Range prefetch operation .

REUSE unsigned int

Maximum number of bytes to be accessed before next RPRFM .

STRIDE signed int

Stride to advance after accessing LENGTH bytes.

VADDR unsigned int

Base virtual address of range prefetch request.

RUN_STATE

Run state transition. Fields:

INST_COUNT unsigned int

Ticks count at point of transition.

NEW enum

New run state.

OLD enum

Old run state.

SBP_SAMPLE_COLLISION

DISPLAY Statistical profiling samples collided .

SBP_SAMPLE_DISCARDED

DISPLAY Operation discarded by statistical profiling unit when configured in discard mode.
Fields:

BUFFER_POINTER unsigned int

Value of buffer pointer.

DATA unsigned int

Data discarded.

SIZE unsigned int

Size of sample, in bytes.

SBP_SAMPLE_END

DISPLAY Statistical profiling of an operation ended.

SBP_SAMPLE_POP

DISPLAY Operation which might be sampled by statistical profiling.

SBP_SAMPLE_RECORDED

DISPLAY Operation sampled by statistical profiling has been written to the buffer at %{BUFFER_ADDRESS} with size %{SIZE} and data %{DATA}. Fields:

BUFFER_ADDRESS unsigned int

Virtual Address of start of buffer write.

DATA unsigned int

Data written to the buffer.

SIZE unsigned int

Size of sample written to buffer, in bytes.

SBP_SAMPLE_START

DISPLAY Statistical profiling of an operation started.

SBP_SAMPLE_TAKEN

DISPLAY Operation sampled by statistical profiling after filtering.

SEMIHOSTING_CALL

Call of a semihost function occurred. Fields:

PC unsigned int

The program counter after the semihosting call.

SEMIHOSTING_PRECALL

About to call semihost. Fields:

EL2 bool

Use EL2 translation regime.

NS bool

Is Non-Secure.

REG_WIDTH unsigned int

The current register width in bytes.

SIGNAL

External signal state change. Fields:

SIGNAL *enum*

Signal that changed.

STATE *bool*

Signal asserted state.

SOFTWARE_STEP

Return the Debug Software Step state. Fields:

STATE *enum*

Software Step state.

SPSR

SPSR change. Fields:

CORE_NUM *unsigned int*

Core number in a multi processor.

MODE *enum*

Which of the banked SPSR registers is written.

OLD_VALUE *unsigned int*

The old SPSR value.

VALUE *unsigned int*

The new SPSR value.

START_COMPILE

Compilation started. Fields:

VADDR *unsigned int*

Instruction where compilation begins.

STG_MULTIPLE_MICRO_OPS_ADDRESS_RANGE

Address range for MTE stores with multiple uops. Fields:

ADDR_END *unsigned int*

Final address.

ADDR_START *unsigned int*

Starting address.

STORE_MULTIPLE_REGS_ADDRESS_RANGE

Address range for stores with multiple registers. Fields:

ADDR_END *unsigned int*

Final address.

ADDR_START *unsigned int*

Starting address.

SUBOPTIMAL_LDST_RETIRED

suboptimal load/store. Fields:

ACCESS_SIZE unsigned int

Log2 of the access width in bytes used for alignment checking (1: half-word, 2: word ...).

OPTIMAL_ALIGN_SELECTOR unsigned int

Selector to decide the optimal alignment size (1: access width, 2: 4-bytes, 3: 8-bytes, ..., 12: 4-kbytes).

VADDR unsigned int

The virtual address of the access.

SVE_INST_SPEC

SVE operations speculatively executed.

SVE_LD_RETIRED

SVE load instructions architecturally executed.

SVE_LOADS

SVE load accesses. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ACQREL enum

Is this an acquire/release.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DATA unsigned int

The data read or written.

ELEMENT_SIZE unsigned int

Width of each element.

ENDIANNESS enum

Byte ordering of this access.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

LOCK enum

Normal, exclusive or locked access.

MECID unsigned int

Memory Encryption Context ID, if applicable.

MECID2 unsigned int

Memory Encryption Context ID of the second page, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PADDR_TAG unsigned int

If tag checked, the physical address tag for the access.

PAGE_SIZE unsigned int

Page size as $\log_2(\text{size in bytes})$.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK, 2=Exclusive Failed, 3=Unsupported.

SIZE unsigned int

Width of the access in bytes. Only required if DATA is not traced.

TAG_CHECKED bool

Is this access tag checked.

TAG_CHECKED2 bool

Is the second page access tag checked.

TRANS bool

Is this a translated access.

VADDR unsigned int

The virtual address of the access.

SVE_MEM_PREFETCH

Scalable Vector Extension memory prefetch. Fields:

CACHE_LEVEL enum

The preferred cache level.

HINT enum

What the memory is expected to be used for.

SIZE unsigned int

The prefetch size in bytes.

TEMPORAL bool

Whether the data is temporal.

VADDR unsigned int

The virtual address of the prefetch.

SVE_REG_READ

Scalable Vector Extension register read. Fields:

ID enum

The register identifier.

SM bool

Whether the PE is in Streaming Mode.

VALUE unsigned int

The read data.

SVE_REG_UPDATE

Scalable Vector Extension register update. Fields:

ID enum

The register identifier.

SM bool

Whether the PE is in Streaming Mode.

VALUE unsigned int

The written data.

SVE_REG_WRITE

Scalable Vector Extension register write. Fields:

ID enum

The register identifier.

SM bool

Whether the PE is in Streaming Mode.

VALUE unsigned int

The written data.

SVE_STORES

SVE store accesses. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ACQREL enum

Is this an acquire/release.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DATA unsigned int

The data read or written.

ELEMENT_SIZE unsigned int

Width of each element.

ENDIANNESS enum

Byte ordering of this access.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

LOCK enum

Normal, exclusive or locked access.

MECID unsigned int

Memory Encryption Context ID, if applicable.

MECID2 unsigned int

Memory Encryption Context ID of the second page, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PADDR_TAG unsigned int

If tag checked, the physical address tag for the access.

PAGE_SIZE unsigned int

Page size as $\log_2(\text{size in bytes})$.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK, 2=Exclusive Failed, 3=Unsupported.

SIZE unsigned int

Width of the access in bytes. Only required if DATA is not traced.

TAG_CHECKED bool

Is this access tag checked.

TAG_CHECKED2 bool

Is the second page access tag checked.

TRANS bool

Is this a translated access.

VADDR unsigned int

The virtual address of the access.

SVE_ST_RETIRED

SVE store instructions architecturally executed.

SVE_UNALIGNED_LDST_RETIRED

SVE load or store instructions architecturally executed that would cause alignment fault if A bit is set.

SVE_Z_PRED_MEM_ACCESS_INFO

Information about SVE/SME predicated accesses to Z registers (non Gather/Scatter). Fields:

MEM_ELEM_SIZE unsigned int

Width of elements accessed in memory (in bytes).

RnW bool

Whether this is a Read not a Write.

VA unsigned int

Virtual base address computed for accesses.

SYNC

Called for every synchronization, that is every quantum break. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The instruction count of this CPU.

LOCAL_QUANTUM unsigned int

The local quantum of this CPU.

LOCAL_TIME unsigned int

The local time of this CPU.

SYNC_REASON enum

Reason for sync.

SYSCALL

System call instruction executed. Fields:

IMM unsigned int

Immediate value of the system call instruction.

TYPE enum

System call type.

VADDR unsigned int

Instruction that caused the system call.

SYSREG_READ128

System Coprocessor 128 bit register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

REGNUM enum

Internal register number.

REGNUM_PHYS enum

Internal register number physically accessed (e.g. in presense of VHE redirection).

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_READ64

System Coprocessor register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REGNUM enum

Internal register number.

REGNUM_OPERAND enum

Register number corresponding to the instruction operand.

REGNUM_PHYS enum

Internal register number physically accessed (e.g. in presense of VHE redirection).

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc unsigned int

opcode 1.

opc0 unsigned int
opcode 0.

opc2 unsigned int
opcode 2.

SYSREG_UPDATE128

Triggers when the system updates a register. Fields:

CRm unsigned int
CRm.

CRn unsigned int
CRn.

NS bool
Secure or nonsecure banked register is accessed.

REG enum
Register number.

UNKNOWN unsigned int
Bits of the register which became unknown.

VALUE unsigned int
Value written to the register.

opc unsigned int
opcode 1.

opc0 unsigned int
opcode 0.

opc2 unsigned int
opcode 2.

SYSREG_UPDATE32

Triggers when the system updates a register. Fields:

CRm unsigned int
CRm.

CRn unsigned int
CRn.

NS bool
Secure or nonsecure banked register is accessed.

REG enum
Register number.

UNKNOWN unsigned int
Bits of the register which became unknown.

VALUE unsigned int

Value written to the register.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_UPDATE64

Triggers when the system updates a register. Fields:

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG enum

Register number.

UNKNOWN unsigned int

Bits of the register which became unknown.

VALUE unsigned int

Value written to the register.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_WRITE128

System Coprocessor 128 bit register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

REGNUM enum

Internal register number.

REGNUM_PHYS enum

Internal register number physically accessed (e.g. in presense of VHE redirection).

UNDEF bool

The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_WRITE64

System Coprocessor register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REGNUM enum

Internal register number.

REGNUM_OPERAND enum

Register number corresponding to the instruction operand.

REGNUM_PHYS enum

Internal register number physically accessed (e.g. in presense of VHE redirection).

UNDEF bool

The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

TRBU_BUFFER_WRAP

DISPLAY Trace buffer current write pointer wrapped.

TRBU_TRIGGER_EVENT

DISPLAY Trace buffer Trigger Event.

TRBU_WRITE

DISPLAY PETU trace data has been written to the TRBU memory buffer at %{BUFFER_ADDRESS} with size %{SIZE} and data %{DATA}. Fields:

BUFFER_ADDRESS unsigned int

Virtual Address of start of buffer write.

DATA unsigned int

Data written to the buffer.

SIZE unsigned int

Size of sample written to buffer, in bytes.

UNALIGNED_LDST_RETIRED

Processor unaligned load/store. Fields:

ACCESS_SIZE unsigned int

Log2 of the access width in bytes used for alignment checking (1=>half-word, 2->word ...).

NUMBER_OF_BEATS unsigned int

Number of accesses (beats) in this burst (Total size in bytes is ACCESS_SIZE * NUMBER_OF_BEATS).

VADDR unsigned int

The virtual address of the access.

VFP_D_REGS

VFP/NEON D 64 bit register write. Fields:

ALIASING enum

Type of register aliasing used.

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a write to the corresponding bit in the register. For example, 0x1 would indicate a write to VALUE[0].

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

VFP_D_REGS_READ

VFP/NEON D 64 bit register read. Fields:

ID unsigned int

The register number.

VALUE unsigned int

The current value of the register.

VFP_Q_REGS

VFP/NEON Q 128 bit register write. Fields:

ALIASING enum

Type of register aliasing used.

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a write to the corresponding byte in the register. For example, 0x1 would indicate a write to VALUE[7:0].

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

VFP_Q_REGS_READ

VFP/NEON Q 128 bit register read. Fields:

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a read to the corresponding byte in the register. For example, 0x1 would indicate a read to VALUE[7:0].

VALUE unsigned int

The current value of the register.

VFP_SYS_REGS

Writes to the VFP/NEON units system registers. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ID enum

Which VFP system register is written.

MASK unsigned int

The mask indicating updated cumulative bits.

OLD_VALUE unsigned int

The register's old value overwritten.

VALUE unsigned int

The new value written to the VFP system register.

VFP_S_REGS

VFP/NEON S 32 bit register write. Fields:

ALIASING enum

Type of register aliasing used.

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a write to the corresponding bit in the register. For example, 0x1 would indicate a write to VALUE[0].

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

VFP_S_REGS_READ

VFP/NEON S 32 bit register read. Fields:

ID unsigned int

The register number.

VALUE unsigned int

The current value of the register.

VIRTUAL_TAG_ALLOCATED

Virtual tag information when allocated. Fields:

TAG_VADDR **unsigned int**

Virtual tag virtual address.

VADDR **unsigned int**

Data Virtual address.

VIRTUAL_TAG **unsigned int**

Virtual tag value.

WAYPOINT

Signals end of basic block execution. Fields:

CORE_NUM **unsigned int**

Core number in a multi processor.

ISET **enum**

Origin instruction set.

IS_COND **bool**

Indicates if this is a conditional waypoint.

PC **unsigned int**

Origin address (or 0 if unavailable).

TAKEN **bool**

Indicates if this waypoint was taken.

TARGET **unsigned int**

Destination address.

TARGET_ISET **enum**

Destination instruction set.

WFE_END

WFE ended. Fields:

INST_COUNT **unsigned int**

Ticks count when leaving WFE.

WFE_EVENT_REGISTER

WFE event register status: set/clear, reason. Fields:

INST_COUNT **unsigned int**

Ticks count.

REASON **enum**

Reason for set/clear. Only REASON==0 (Cleared by WFE) clears the bit, all other reasons set it.

WFE_IGNORED

WFE ignored. Fields:

EVENT bool

This WFE was ignored because the event register was set.

INST_COUNT unsigned int

Ticks count when ignoring WFE.

TRAPPED bool

This WFE was trapped.

WFE_START

WFE entered. Fields:

INST_COUNT unsigned int

Ticks count when entering WFE.

WFI_END

WFI ended. Fields:

INST_COUNT unsigned int

Ticks count when leaving WFI.

WFI_IGNORED

WFI ignored. Fields:

DISABLED bool

This WFI was ignored because WFI is disabled.

INST_COUNT unsigned int

Ticks count when ignoring WFI.

REASON enum

specifies reason why WFI trace was ignored.

TRAPPED bool

This WFI was trapped.

WFI_START

WFI entered. Fields:

INST_COUNT unsigned int

Ticks count when entering WFI.

WFI_WAKEUP

WFI wakeup. Fields:

INST_COUNT unsigned int

Ticks count when WFI wakeup occurred.

REASON enum

Reason for wakeup.

2.69 ARM_Neoverse-V3

This section describes the trace sources.

AA64_ASE_SVE_REGS

Modification of AArch64 Advanced SIMD or SVE Vector Registers (via SIMD or Scalar operations), SVE Predicate Registers and ZT0. Fields:

ID **enum**

The register identifier.

MASK **unsigned int**

Bitmask of the register to signal the modified bits in the VALUE field.

SM **bool**

Whether the PE is in Streaming Mode.

VALUE **unsigned int**

Value of AArch64 vector register.

ASYNC_MEMORY_FAULT

Context ID Register write. Fields:

FAULT **unsigned int**

Fault status present in ESR.ISS encoding for Serror in ARCH64 and present in DFSR encoding for external abort format in ARCH32.

PADDR **unsigned int**

Physical Address (or 0 if unavailable).

VADDR **signed int**

Virtual Address (or 0 if unavailable).

ATOMIC_END_ACCESS

Bus trace access for atomics. Fields:

ACCESS_FAIL **bool**

Memory access failed.

ADDR **unsigned int**

The virtual address of the access.

ATTR **unsigned int**

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

COMPARE_VALUE **unsigned int**

Compare value for CAS.

LOAD_VALUE unsigned int

Loaded values.

NSDESC unsigned int

The security state of the access.

OPERAND_VALUE unsigned int

Operation's operand.

OPERATION enum

Operation type.

PADDR unsigned int

The physical address of the access.

PRIV bool

Is this a privileged access?.

ATOMIC_START_ACCESS

Bus trace access for atomics. Fields:

ADDR unsigned int

The virtual address of the access.

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

COMPARE_VALUE unsigned int

Compare value for CAS.

NSDESC unsigned int

The security state of the access.

OPERAND_VALUE unsigned int

Operation's operand.

OPERATION enum

Operation type.

PADDR unsigned int

The physical address of the access.

PRIV bool

Is this a privileged access?.

ArchMsg.Error.exit_code

DISPLAY exit code: %{EXITCODE} component kind: %{KIND} reason: %{REASON}. Fields:

EXITCODE signed int

Exit code to be returned.

KIND string

Component kind that invoked the exit code trace.

REASON string

Reason why the exit code trace was invoked (optional).

ArchMsg.Warning.BRBInjectionUnknown

DISPLAY BRB INJ with uninitialized injection registers.

ArchMsg.Warning.IgnoredWarmResetRequest

Event fires when software attempts to request warm reset through write to deprecated DBGPRCR.CWRR.

ArchMsg.Warning.RAS_ErrselrUnpredictableState

DISPLAY ERRSELR is in unpredictable state due to value greater than number_of_error_records written to it.

ArchMsg.Warning.RAS_UnpredictableStateWrite

DISPLAY %{REG_NAME} write ignored due to unpredictable ERRSELR state. Fields:

REG_NAME string

Name of register being written.

ArchMsg.Warning.ReportedRevisionRegisterRead

DISPLAY Reported revision and patch do not match that of the cpu. Fields:

ORIGINAL_VALUE unsigned int

The value the register would have based on the modelled revision.

REGISTER string

The name of the register that has ben read.

REPORTED_VALUE unsigned int

The value the register has been overridden to.

ArchMsg.Warning.Unpredictably Indexed PM Event Register

TAGS **UNPREDICTABLE** DISPLAY Event %{IsDirect:(Selector value|Register)} %{SEL} is >= %{N}. Fields:

IsDirect unsigned int

Direct Access rather than indirect.

N unsigned int

Number of Accessible Registers.

SEL unsigned int

Selector.

ArchMsg.Warning.cache_contents_unknown

A cache has been activated without being invalidated and may contain **UNKNOWN** content. DISPLAY %{SIDE:(D|I)}-Cache was enabled in %{REGIME:(SECURE|NON_SECURE|HYP)} regime but was not invalidated since power-on: cache lines could contain **UNKNOWN** content

%{SIDE:(Note: D-side caches must invalidate SECURE lines even if the cache is only used by NON_SECURE code in order to ensure that unexpected dirty lines tagged SECURE are not naturally evicted)}. Fields:

REGIME unsigned int

Current translation regime.

SIDE unsigned int

0 d-side cache or unified cache, 1 i-side.

ArchMsg.Warning.dcimvac_matches_watchpoint

TAGS **UNPREDICTABLE** DISPLAY Watchpoints matching an AArch32 DCIMVA are implementation defined.

ArchMsg.Warning.decode_fieldmismatch

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with different fields %{FIELD1} and %{FIELD2}. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_frachitsnegative

TAGS **UNPREDICTABLE** DISPLAY VCVT Instruction is unpredictable with a negative number of fraction bits.

ArchMsg.Warning.decode_initblock

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable in an IT block.

ArchMsg.Warning.decode_initblocknotlast

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable in an IT block when not the last.

ArchMsg.Warning.decode_invalid_condition

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable when the condition is not AL.

ArchMsg.Warning.decode_invalidfieldcombination

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with the values in %{FIELD1} and %{FIELD2}. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_invalidvalue

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with the value in %{FIELD}. Fields:

FIELD enum

Field name.

ArchMsg.Warning.decode_mem_hint_unallocated

TAGS **UNPREDICTABLE** DISPLAY There is no memory hint allocated to this bit pattern.

ArchMsg.Warning.decode_registermatch

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with %{FIELD1} and %{FIELD2} as the same register. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_registermismatch

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with %{FIELD1} and %{FIELD2} as different registers. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_registeroutofrange

TAGS **UNPREDICTABLE** DISPLAY Floating-point instruction is unpredictable indexing beyond the end of the register bank.

ArchMsg.Warning.decode_sbzsbo

TAGS **UNPREDICTABLE** DISPLAY Reserved bits in the instruction are not canonical.

ArchMsg.Warning.decode_transactiontoobig

TAGS **UNPREDICTABLE** DISPLAY Attempt to perform a memory transaction of more than 128 bytes.

ArchMsg.Warning.decode_unpred_other

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable.

ArchMsg.Warning.decode_unpreduseofpc

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable when %{FIELD} is PC. Fields:

FIELD enum

Field name.

ArchMsg.Warning.decode_unpreduseofr13

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable when %{FIELD} is R13. Fields:

FIELD enum

Field name.

ArchMsg.Warning.decode_unpreduseofr13inreglist

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with R13 in register-list.

ArchMsg.Warning.decode_useofd16orhigher

TAGS **UNPREDICTABLE** DISPLAY Attempt to use register D16 or higher where it is not allowed.

ArchMsg.Warning.decode_writebackandbaseinlist

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with write back when the base register is in the transfer list.

ArchMsg.Warning.decode_zeroregistersinlist

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable when no registers are to be transferred.

ArchMsg.Warning.misaligned_ptrs

DISPLAY TRBPTR_EL1(%{TRBPTR_EL1}) or TRBTRG_EL1(%{TRBTRG_EL1}) are not aligned as per TRBIDR_EL1.Align(%{TRBIDR_EL1}). Fields:

TRBIDR_EL1 unsigned int

TRBIDR_EL1.Align value.

TRBPTR_EL1 unsigned int

Current write pointer.

TRBTRG_EL1 unsigned int

Trigger pointer.

ArchMsg.Warning.out_of_range_ptrs

DISPLAY TRBPTR_EL1(%{TRBPTR_EL1}) is out of range as per TRBBASER_EL1(%{TRBBASER_EL1}) or TRBLIMITR_EL1.LIMIT(%{TRBLIMITR_EL1}). Fields:

TRBBASER_EL1 unsigned int

Base pointer.

TRBLIMITR_EL1 unsigned int

Limit pointer.

TRBPTR_EL1 unsigned int

Current write pointer.

ArchMsg.Warning.recursive_branch

An instruction is performing a branch that targets the same instruction. If this was intended then a WFI might be more effective DISPLAY recursive branch detected at PC=%{PC}. PRIMARY KEY PC. Fields:

PC unsigned int

Address of instruction causing the branch.

ArchMsg.Warning.recursive_exception

An instruction has generated an exception that targets the same instruction. DISPLAY recursive %{TYPE} exception detected at PC=%{PC} (CPSR=%{CPSR}). PRIMARY KEY PC. Fields:

CPSR unsigned int

Processor state.

PC unsigned int

Address of instruction causing the exception.

TYPE enum

Type of exception.

ArchMsg.Warning.reentrant_vector_catch

TAGS **UNPREDICTABLE** DISPLAY Vector catch debug event on Prefetch or Data abort vector %{ADDR}. Fields:

ADDR unsigned int

vector address.

ArchMsg.Warning.reserved_it_state

TAGS **UNPREDICTABLE** DISPLAY Execution with reserved IT state %{ITSTATE}. Fields:

ITSTATE unsigned int

reserved IT state.

ArchMsg.Warning.tlb_contents_unknown

Translation has been enabled without invalidating the TLB. DISPLAY Translation was enabled in the %{PAS:(Secure|Non-secure|Root|Realm)} EL%{EL:d} translation regime (vmid=%{VMID:x}) without invalidating %{INVALIDITY:(|d-side |i-side |)}entries since power-on: %{INVALIDITY:(|D|I)}TLB could contain **UNKNOWN** content. Fields:

EL unsigned int

Exception level of translation regime.

INVALIDITY unsigned int

Which TLB may be invalid (if TLBs are separate).

NON_SECURE bool

Whether the translation regime is non-secure.

PAS unsigned int

Physical address space of translation regime.

VMID unsigned int

Current VMID, if applicable.

ArchMsg.Warning.unknown_DLR_DSPSR

Exiting debug state with unknown DLR or DSPSR.

ArchMsg.Warning.unknown_ELR_SPSR

Returning from debug exception with unknown ELR or SPSR.

ArchMsg.Warning.unpred_FAR_on_LRCPC3_non_atomic_mem_access

TAGS **UNPREDICTABLE** Unpredictable FAR/FSTATUS on LRCPC3 pair instructions resulting in two separate memory accessesDISPLAY Unpredictable FAR on LRCPC3 non atomic memory access at %{ADDR1} with FSTATUS %{FSTATUS1} or %{ADDR2} with FSTATUS %{FSTATUS2}. Fields:

ADDR1 unsigned int

Possible FAR address 1.

ADDR2 unsigned int

Possible FAR address 2.

FSTATUS1 unsigned int

Possible fault status value 1.

FSTATUS2 unsigned int

Possible fault status value 2.

ArchMsg.Warning.unpred_async_tag_check_fail_on_sve_speculative_load_do_update_ffr

Unpredictable set to FALSE of the FFR element associated to SVE speculative load with an async tag check fail TAGS **UNPREDICTABLE**.

ArchMsg.Warning.unpredictable_a32_breakpoint

TAGS **UNPREDICTABLE** DISPLAY Unpredictable breakpoint on a A32 instruction at %{ADDR1} Offset %{OFFSET} BAS %{BAS} for a %{IS_ADDRESS_MISMATCH_BREAKPOINT:(Mismatch|Match)} Breakpoint number %{N:d} Breakpoint address %{ADDR2}. Fields:

ADDR1 unsigned int

Address.

ADDR2 unsigned int

Breakpoint address.

BAS unsigned int

BAS field.

IS_ADDRESS_MISMATCH_BREAKPOINT bool

Is a Mismatch breakpoint, not a Match breakpoint.

N unsigned int

Breakpoint number.

OFFSET unsigned int

Always has a value of zero.

ArchMsg.Warning.unpredictable_banked_register_access

TAGS **UNPREDICTABLE** DISPLAY Access to banked register (R=%{R:d} SYSm=%{SYSM:2x}) is unpredictable. Fields:

R unsigned int

R bit.

SYSM unsigned int

SYSm.

ArchMsg.Warning.unpredictable_ccfail_undef

TAGS **UNPREDICTABLE** DISPLAY An instruction %{OPCODE} at VA=%{VADDR} failed its condition codes check but would otherwise have caused a non-data-dependent trap or undefined exception. Behaviour is implementation-defined choice of exception or nop. Fields:

OPCODE unsigned int

opcode of the instruction.

VADDR unsigned int

Virtual address of the instruction being executed.

ArchMsg.Warning.unpredictable_ici_bits

TAGS **UNPREDICTABLE** DISPLAY ICI bits do not apply or are architecturally unpredictable for instruction.

ArchMsg.Warning.unpredictable_t32_breakpoint

TAGS **UNPREDICTABLE** DISPLAY Unpredictable breakpoint on a T32 instruction at %{ADDR1} Offset %{OFFSET} BAS %{BAS} for a %{IS_ADDRESS_MISMATCH_BREAKPOINT:(Mismatch|Match)} Breakpoint number %{N:d} Breakpoint address %{ADDR2}. Fields:

ADDR1 unsigned int

Address.

ADDR2 unsigned int

Breakpoint address.

BAS unsigned int

BAS field.

IS_ADDRESS_MISMATCH_BREAKPOINT bool

Is a Mismatch breakpoint, not a Match breakpoint.

N unsigned int

Breakpoint number.

OFFSET unsigned int

Indicates which halfword of a 32-bit Thumb instruction the breakpoint is set on. 0 for the first halfword or if the instruction is 16 bits wide, or 1 for the second halfword.

ArchMsg.Warning.unpredictable_t32_vector_catch

TAGS **UNPREDICTABLE** DISPLAY Unpredictable Vector catch exception on a T32 instruction at %{ADDR} Offset %{OFFSET}. Fields:

ADDR unsigned int

Address.

OFFSET unsigned int

Indicates which halfword of a 32-bit Thumb instruction the vector catch exception is set on. 0 for the first halfword or if the instruction is 16 bits wide, or 1 for the second halfword.

ArchMsg.Warning.unpredictable_watchpoint_far

TAGS **UNPREDICTABLE** DISPLAY The value for FAR/EDWAR for hitting this watchpoint may be anywhere in the range %{LOWER_BOUND} to %{UPPER_BOUND}. Fields:

LOWER_BOUND unsigned int

the lowest address accessed by the instruction that triggered the watchpoint.

UPPER_BOUND unsigned int

the highest watchpointed address accessed by that instruction.

ArchMsg.Warning.unpredictable_watchpoint_far_isscmp

DISPLAY The value for FAR/EDWAR for hitting this watchpoint may be anywhere in the range %{LOWER_BOUND} to %{UPPER_BOUND}. Fields:

LOWER_BOUND unsigned int

the lowest address accessed by the instruction that triggered the watchpoint.

UPPER_BOUND unsigned int

the highest watchpointed address accessed by that instruction.

ArchMsg.Warning.warning_AdvSIMDExpandImmUnexpectedZero

TAGS UNPREDICTABLE DISPLAY AdvSIMDExpandImm may treat this immediate value as UNPREDICTABLE.

ArchMsg.Warning.warning_ConditionalSMC

TAGS UNPREDICTABLE DISPLAY SMC instruction has UNPREDICTABLE effects when conditional and when combined with traps.

ArchMsg.Warning.warning_TLBID_out_of_range

TLBID out of range in TLBI according to the configuration of NIS, NVIS, NOS and NVOS. Fields:

NIS unsigned int

NIS.

NOS unsigned int

NOS.

NVIS unsigned int

NVIS.

NVOS unsigned int

NVOS.

TLBID unsigned int

The TLBID value used in the TLBI instruction.

ArchMsg.Warning.warning_TTL_or_TG_mismatches_current_translation_granule

TTL or TG mismatches current translation granule in TLBI VA/RVA/IPA/RIPA. Fields:

CURRENT_TG unsigned int

Current TG as indicated by TCR_ELx or VTCR_EL2.

TLBI_TG unsigned int

TG (or TTL[3:2]) used in TLBI instruction.

ArchMsg.Warning.warning_access_crosses_page_boundary

TAGS UNPREDICTABLE DISPLAY Access crosses page boundary.

ArchMsg.Warning.warning_access_crosses_page_boundary_has_fault

TAGS **UNPREDICTABLE** DISPLAY Access that crosses page boundary has fault(s).

ArchMsg.Warning.warning_access_crosses_page_boundary_has_fault_on_both_pages

TAGS **UNPREDICTABLE** DISPLAY Access that crosses page boundary has a fault on each page.

Fields:

ADDR1 unsigned int

FAR address on first page.

ADDR2 unsigned int

FAR address on second page.

FSTATUS1 unsigned int

Fault status on first page.

FSTATUS2 unsigned int

Fault status on second page.

ArchMsg.Warning.warning_access_crosses_page_boundary_spanning_different_memory

TAGS **UNPREDICTABLE** DISPLAY Access at address %{ADDR} crosses page from MEM_TYPE %{MEMTYPE_PAGE1} to %{MEMTYPE_PAGE2}. Fields:

ADDR unsigned int

address of access crossing page.

MEMTYPE_PAGE1 signed int

First Page's memory type (0-Normal, 1-Non-Normal).

MEMTYPE_PAGE2 signed int

Second Page's memory type (0-Normal, 1-Non-Normal).

ArchMsg.Warning.warning_access_wraps_around_memory

TAGS **UNPREDICTABLE** DISPLAY Access wraps around memory in %{WIDTH:d}bit mode.

Fields:

WIDTH unsigned int

Address width.

ArchMsg.Warning.warning_bcr_linking_status

TAGS **UNPREDICTABLE** DISPLAY DBGBCR.{BT2,BT} combination gives unpredictable behaviour.

ArchMsg.Warning.warning_bcr_mask_reserved

TAGS **UNPREDICTABLE** DISPLAY DBGBCR_ELn.MASK is set to a reserved value.

ArchMsg.Warning.warning_bvr_masked_bit_not_zero

TAGS **UNPREDICTABLE** DISPLAY DBGBVRn_EL1.VA contains a bit masked by DBGBCRn_EL1.MASK that is not zero.

ArchMsg.Warning.warning_ccsidr_unimplemented_level

TAGS **UNPREDICTABLE** DISPLAY CCSIDR read while CSSELR %{CSSELR:x} points to an unimplemented cache level. Fields:

CSSELR unsigned int

current effective value of CSSELR.

ArchMsg.Warning.warning_change_to_ns_when_tge_set

TAGS **UNPREDICTABLE** DISPLAY Attempting to change NS to 1 when HCR.TGE = 1.

ArchMsg.Warning.warning_contiguous_bit_check_abort

TAGS **UNPREDICTABLE** DISPLAY An abort occurred whilst attempting to check TLB entry is contiguous at %{ENTRY_ADDR}. Fields:

ENTRY_ADDR unsigned int

address of conflicting TLB entry.

ArchMsg.Warning.warning_contiguous_bit_error

TAGS **UNPREDICTABLE** DISPLAY TLB entry at %{ENTRY_ADDR} was not contiguous with entry at %{CONTIG_ADDR} Contents are %{ENTRY_DATA} but expected %{CONTIG_DATA}. Fields:

CONTIG_ADDR unsigned int

address of the TLB first read, with which this entry is expected to be contiguous.

CONTIG_DATA unsigned int

expected contents based on the entry at CONTIG_ADDR.

ENTRY_ADDR unsigned int

address of conflicting TLB entry.

ENTRY_DATA unsigned int

contents of conflicting TLB entry.

ArchMsg.Warning.warning_contiguous_bit_error_D128

TAGS **UNPREDICTABLE** DISPLAY TLB entry at %{ENTRY_ADDR} was not contiguous with entry at %{CONTIG_ADDR} Contents are %{ENTRY_DATA_HI} %{ENTRY_DATA_LO} but expected %{CONTIG_DATA}. Fields:

CONTIG_ADDR unsigned int

address of the TLB first read, with which this entry is expected to be contiguous.

CONTIG_DATA unsigned int

expected contents based on the entry at CONTIG_ADDR.

ENTRY_ADDR unsigned int

address of conflicting TLB entry.

ENTRY_DATA_HI unsigned int

Higher bit contents of conflicting TLB entry.

ENTRY_DATA_LO unsigned int

Lower bit contents of conflicting TLB entry.

ArchMsg.Warning.warning_cp10_cp11_mismatch

TAGS **UNPREDICTABLE** DISPLAY Unpredictable configuration of CP10 and CP11 controls at EL
%{EL:d}. Fields:

EL unsigned int

EL owning the control.

ArchMsg.Warning.warning_cp10_cp11_reserved_value

TAGS **UNPREDICTABLE** DISPLAY Unpredictable configuration of CPACR CP10 (%{CP10}) or
CP11 (%{CP11}). Fields:

CP10 unsigned int

CP10 access.

CP11 unsigned int

CP11 access.

ArchMsg.Warning.warning_csselr_level_out_of_range

TAGS **UNPREDICTABLE** DISPLAY CSSELR.Level out of range, written value %{LEVEL} with only
%{IMPLEMENTED:d} levels implemented. Fields:

IMPLEMENTED unsigned int

number of cache levels implemented.

LEVEL unsigned int

written value of CSSELR.Level.

TYPE enum

The type of cache selected by CSSELR.InD and TnD.

ArchMsg.Warning.warning_debug_flow_control_bits_not_obeyed

TAGS **UNPREDICTABLE** DISPLAY Flow control bits for %{REGISTER} were not obeyed, causing
%{ERROR}. Fields:

ERROR enum

type of error resulting.

REGISTER enum

register under consideration.

ArchMsg.Warning.warning_debug_register_access_during_reset

TAGS **UNPREDICTABLE** DISPLAY Unpredictable %{IS_WRITE:(read|write)} access to debug
register offset %{OFFSET} during reset. Fields:

IS_WRITE unsigned int

Write Not Read.

OFFSET unsigned int

Register Offset.

ArchMsg.Warning.warning_decode_cps_inconsistent_fields

TAGS **UNPREDICTABLE** DISPLAY The mode setting fields in the CPS instruction are inconsistent.

ArchMsg.Warning.warning_decode_invalid_state

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable in current exception-level/security state.

ArchMsg.Warning.warning_default_cacheable_mmu_on

TAGS **UNPREDICTABLE** DISPLAY Default cacheable is enabled (HCR.DC = 1) while MMU is enabled.

ArchMsg.Warning.warning_default_cacheable_vmmu_off

TAGS **UNPREDICTABLE** DISPLAY Default cacheable is enabled (HCR.DC = 1) while HCR.VM = 0.

ArchMsg.Warning.warning_deprecated_wvr_bit_2_set

TAGS **UNPREDICTABLE** DISPLAY DBGWVR%{N:d}_EL1=%{WVR:h} has bit 2 set, deprecated in ARMv8. Fields:

N unsigned int

index.

WVR unsigned int

DBGWVR.

ArchMsg.Warning.warning_eret_while_software_step_active_pending

Missing ISB between setting MDSCR_EL1.SS and ERET TAGS **UNPREDICTABLE**.

ArchMsg.Warning.warning_exclusive_to_non_normal

TAGS **UNPREDICTABLE** DISPLAY Exclusive Access to Non-Normal Memory %{ADDR}. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_exclusive_to_non_writeback

TAGS **UNPREDICTABLE** DISPLAY Exclusive Access to Non-Writeback-Cacheable Memory %{ADDR}. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_execute_from_device_memory

TAGS **UNPREDICTABLE** DISPLAY An attempt was made to execute from Device memory.

ArchMsg.Warning.warning_illegal_cpsr_mode

TAGS **UNPREDICTABLE** DISPLAY Writing an illegal or unimplement mode (%{NEW_MODE:h}) to CPSR was unpredictable. Fields:

NEW_MODE unsigned int

New value of CPSR.M.

ArchMsg.Warning.warning_illegal_srs_mode

TAGS **UNPREDICTABLE** DISPLAY Illegal or **UNPREDICTABLE** mode (%{MODE:h}) used for SRS instruction. Fields:

MODE unsigned int

mode for Banked SP.

ArchMsg.Warning.warning_implementation_defined_read_debug_register_in_SCS

Read Access to Debug registers from the processor is IMPLEMENTATION DEFINED TAGS IMP_DEF. Fields:

PPB_OFFSET unsigned int

debug register offset.

ArchMsg.Warning.warning_implementation_defined_sequential_security_transitions_supported

The behavior of sequential instruction fetches that cross from non-secure to secure memory and contain an SG instruction is **CONSTRAINED UNPREDICTABLE** TAGS **UNPREDICTABLE**.

ArchMsg.Warning.warning_implementation_defined_stack_limit_check_supported

It is **IMPLEMENTATION DEFINED** whether stack pointer limit checking is performed for this instruction TAGS IMP_DEF. Fields:

ADDRESS unsigned int

Address of instruction.

OPCODE unsigned int

instruction opcode.

ArchMsg.Warning.warning_implementation_defined_taggednonsharedmemorytype

TAGS IMP_DEF DISPLAY Memory attribute has Tagged, non shared memory type; it is **IMPLEMENTATION DEFINED** whether the memory location is treated as Tagged or non-Tagged.

ArchMsg.Warning.warning_implementation_defined_write_debug_register_in_SCS

Write Access to Debug registers from the processor is IMPLEMENTATION DEFINED TAGS IMP_DEF. Fields:

DATA unsigned int

data attempted to be written.

PPB_OFFSET unsigned int

debug register offset.

ArchMsg.Warning.warning_invalid_tcr_granule

TAGS **UNPREDICTABLE** DISPLAY TCR.TG{%TG_ID:(0|1)} bits have been set to a granule size {%REQUEST:d}K not implemented - using {%SUBSTITUTE:d}K. Fields:

REQUEST signed int

page size requested (or 0 for reserved).

SUBSTITUTE signed int

best guess available page size.

TG_ID bool

bits TG0 or TG1.

ArchMsg.Warning.warning_load_multiple_user_registers_from_user_mode

TAGS UNPREDICTABLE DISPLAY An LDM(user registers) instruction executed from user mode.

ArchMsg.Warning.warning_mte_in_mem_attr_not_supported

TAGS UNPREDICTABLE DISPLAY Memory attribute %{ATTR} has Tagged memory type but Armv8.5-MTE is not implemented. Fields:

ATTR unsigned int

Memory attribute.

ArchMsg.Warning.warning_reserved_breakpoint_state_match

TAGS UNPREDICTABLE DISPLAY %{IS_BREAKPOINT:(Watchpoint|Breakpoint)} programmed with a reserved combination of HMC, SSC and %{IS_BREAKPOINT:(PAC|PMC)}. Fields:

IS_BREAKPOINT unsigned int

Is a breakpoint, not a watchpoint.

ArchMsg.Warning.warning_shareability

TAGS UNPREDICTABLE DISPLAY Unpredictable: combination of the 1st and 2nd stages of translation is Normal Inner Non-Cacheable, Outer Non-Cacheable.

ArchMsg.Warning.warning_software_step_set_while_enabled

TAGS UNPREDICTABLE DISPLAY MDSCR_EL1.SS set to 1 while software step debug exceptions are enabled.

ArchMsg.Warning.warning_thumb_instruction_wraps_around_memory

TAGS UNPREDICTABLE DISPLAY Thumb instruction wraps around memory.

ArchMsg.Warning.warning_ttbr_sbz_bits_are_not_zero

TAGS UNPREDICTABLE DISPLAY %{TTBR}[%{MSB:d}:0] should be zero, but are not. Fields:

MSB unsigned int

x-1.

TTBR enum

which TTBR register.

ArchMsg.Warning.warning_unaligned_address_dbgdtrrx_write

TAGS UNPREDICTABLE DISPLAY Unaligned address %{ADDR} in X0/R0 when DBGDTRRX written in memory access mode. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unaligned_address_dbgdtrtx_read

TAGS UNPREDICTABLE DISPLAY Unaligned address %{ADDR} in X0/R0 when DBGDTRTX read in memory access mode. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unaligned_to_device

TAGS **UNPREDICTABLE** DISPLAY Unaligned Access to Device Memory %{ADDR}. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unaligned_to_strongly_ordered

TAGS **UNPREDICTABLE** DISPLAY Unaligned Access to Strongly Ordered Memory %{ADDR}. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unknown_sau_rnr

TAGS **UNPREDICTABLE** DISPLAY SAU_RNR was set to an unsupported value.

ArchMsg.Warning.warning_unpredictable_AIRCR_PRIS_and_BFHFNMINs

TAGS **UNPREDICTABLE** DISPLAY The effect of setting both AIRCR.BFHFNMINs and AIRCR.PRIS to 1 is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_AIRCR_VECTCLRACTIVE_when_not_in_debug

TAGS **UNPREDICTABLE** DISPLAY The effect of writing a 1 to AIRCR.VECTCLRACTIVE if the processor is not halted in Debug state is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_AIRCR_VECTRESET_and_SYSRESETREQ

TAGS **UNPREDICTABLE** DISPLAY When the processor is halted in Debug state, if a write to the register writes a 1 to both VECTRESET and SYSRESETREQ, the behavior is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_AIRCR_VECTRESET_when_not_in_debug

TAGS **UNPREDICTABLE** DISPLAY The effect of writing a 1 to AIRCR.VECTRESET if the processor is not halted in Debug state is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_AIRCR_incorrect_VKEY

TAGS **UNPREDICTABLE** DISPLAY The value 0x05FA must be written to AIRCR.VKEY, otherwise the register write is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_EXC_RETURN_Reserved_Bit

EXC_RETURN[23:7] are reserved with the special condition that all bits should be written as one. Values other than all 1s are **UNPREDICTABLE**. TAGS **UNPREDICTABLE**. Fields:

EXC_RETURN unsigned int

Special PC value.

SBO_MASK unsigned int

Which bits Should Be One to not UNPRED.

ArchMsg.Warning.warning_unpredictable_PMEVTYPEPn_ELO_VS_reserved_value

TAGS **UNPREDICTABLE** DISPLAY Unpredictable reserved value of PMEVTYPEn<n>_ELO.VS.

ArchMsg.Warning.warning_unpredictable_change_to_DEMCR_MON_STEP_at_insufficient_priority

TAGS UNPREDICTABLE DISPLAY The effect of changing DEMCR.MON_STEP at an execution priority that is lower than the priority of the DebugMonitor exception is **CONSTRAINED UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_change_to_priority_of_active_exception

TAGS UNPREDICTABLE DISPLAY Changing the priority of an active exception is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_clear_lspact

TAGS UNPREDICTABLE DISPLAY **UNPREDICTABLE** state on context switch. Fields:

DRVEXC_TAKEN unsigned int

Derived exception taken, not pended.

REACHED_LAST_STORE unsigned int

Only the last faulted.

ArchMsg.Warning.warning_unpredictable_dbg_exit_unaligned_dlr

TAGS UNPREDICTABLE DISPLAY Debug exit to AArch32 with DLR[0] set to 1. PC[0] can be set to 0 or to DLR[0].

ArchMsg.Warning.warning_unpredictable_exception_catch

TAGS UNPREDICTABLE DISPLAY Generation of an exception catch event is unpredictable.

ArchMsg.Warning.warning_unpredictable_exception_return_inconsistent_state

TAGS UNPREDICTABLE DISPLAY state on exception return is unpredictable.

ArchMsg.Warning.warning_unpredictable_exception_return_instruction

TAGS UNPREDICTABLE DISPLAY exception return instruction is unpredictable.

ArchMsg.Warning.warning_unpredictable_in_debug_state

TAGS UNPREDICTABLE DISPLAY Instruction is **UNPREDICTABLE** when executed in debug state.

ArchMsg.Warning.warning_unpredictable_mair_encoding

TAGS UNPREDICTABLE DISPLAY Memory attribute %{ATTR} has invalid MAIR encoding; this may lead to unpredictable results. Fields:

ATTR unsigned int

Memory attribute.

ArchMsg.Warning.warning_unpredictable_pmu_counter_access

TAGS UNPREDICTABLE DISPLAY Access to PMU counters from non-secure EL0 or EL1 is **UNPREDICTABLE** with MDCR_EL2.HPMN set to 0.

ArchMsg.Warning.warning_unpredictable_prioritization_breakpoint_match_vector_catch

TAGS UNPREDICTABLE DISPLAY Two events with the same priority occurred at the same instruction: Address Matching Vector Catch debug event, and Breakpoint debug event. It is constrained unpredictable which is taken.

ArchMsg.Warning.warning_unpredictable_stack_selection

TAGS UNPREDICTABLE DISPLAY **UNPREDICTABLE** state on context switch.

ArchMsg.Warning.warning_unpredictable_tsize_out_of_range

TAGS **UNPREDICTABLE** DISPLAY **%{TCR}.T%{TTBR:d}SZ (%{TSIZE:d} bits)** is out of range. Must be between 25 and **%{TSIZE_MAX:d}**. Fields:

TCR enum

Which TCR.

TSIZE signed int

Value of TSize.

TSIZE_MAX signed int

The maximum allowed value.

TTBR unsigned int

Which TTBR region.

ArchMsg.Warning.warning_unpredictable_unaligned_pc_as_base_register

TAGS **UNPREDICTABLE** DISPLAY The PC value must be word-aligned, otherwise the behavior of the instruction is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_unaligned_pop_stack

TAGS **UNPREDICTABLE** DISPLAY **CONSTRAINED UNPREDICTABLE** unaligned stack frame **%{FRAME_ADDR:x}** in exception-return. Fields:

FRAME_ADDR unsigned int

unaligned stack frame address.

ArchMsg.Warning.warning_unpredictable_unaligned_to_non_writeback

TAGS **UNPREDICTABLE** DISPLAY Unaligned Access to Non-Writeback-Cacheable Memory.

ArchMsg.Warning.warning_unpredictable_vmsa_memattrib

TAGS **UNPREDICTABLE** DISPLAY Unpredictable vmsa memory attribute with texcb value programmed to **%{TEXCB}** when tex remap is **%{TEX_REMAP}**. Fields:

TEXCB unsigned int

texcb value.

TEX_REMAP bool

use tex remap.

ArchMsg.Warning.warning_unpredictable_vtcr_t0sz_sl0

TAGS **UNPREDICTABLE** DISPLAY The combination of VTCR.TOSZ=**%{TOSZ:d}** and VTCR.SL0=**%{SL0:d}** is unpredictable. Fields:

SL0 signed int

VTCR.SL0, starting level for stage 2 translation table walks.

TOSZ signed int

VTCR.TOSZ, required input address range.

ArchMsg.Warning.warning_unpredictable_write_DHCSR

TAGS **UNPREDICTABLE** DISPLAY **UNPREDICTABLE** set of bit-changes in DHCSR. Fields:

NEW unsigned int

data attempted to be written.

OLD unsigned int

previous value.

ArchMsg.Warning.warning_unpredictable_write_SHCSR

TAGS UNPREDICTABLE DISPLAY UNPREDICTABLE setting a pending bit to 1 for an exception with priority >= execution priority.

ArchMsg.Warning.warning_unsupported_access_to_memory_mapped_register

TAGS UNPREDICTABLE DISPLAY Access to memory mapped register is unsupported.

ArchMsg.Warning.warning_user_jmcr_access

TAGS UNPREDICTABLE DISPLAY User %{WRITE:(write|read)} access to JMCR is unpredictable. Fields:

WRITE unsigned int

access is write.

ArchMsg.Warning.warning_wcr_mask_and_bas

TAGS UNPREDICTABLE DISPLAY DBGWCR_EL1.MASK is non-zero and BAS is not 0xff.

ArchMsg.Warning.warning_wcr_mask_reserved

TAGS UNPREDICTABLE DISPLAY DBGWCR_EL1.MASK is set to a reserved value.

ArchMsg.Warning.warning_wcr_non_configuous_bas

TAGS UNPREDICTABLE DISPLAY DBGWCR_EL1.BAS has non-contiguous set of ones.

ArchMsg.Warning.warning_wvr_masked_bit_not_zero

TAGS UNPREDICTABLE DISPLAY DBGWVRn_EL1.VA contains a bit masked by DBGWCRn_EL1.MASK that is not zero.

ArchMsg.Why.why_branch_target_exception

A branch target exception has occurred. DISPLAY branch target exception: target pc=%{PC}, target opcode=%{OPCODE}. PRIMARY KEY PC. Fields:

BTYPE unsigned int

pstate.BTYPE at time branch taken.

OPCODE unsigned int

opcode at branch target.

PC unsigned int

Virtual Address of branch target.

ArchMsg.Why.why_illegal_state

An illegal mode change has occurred. DISPLAY illegal mode change: %{MESSAGE}. Fields:

MESSAGE string

The message.

ArchMsg.warning_invalid_interval_counter_reload_value

Invalid interval counter reload value in PMSIRR_EL1.INTERVAL: software should set this to a value greater than or equal to the minimum indicated by PMSIDR_EL1.Interval.

ArchMsg.warning_invalid_spe_buffer_write_pointer

Invalid value programmed in PMBPTR_EL1.PTR or PMBLIMITR_EL1.LIMIT: Current write pointer must be at least one sample record below the write limit pointer and PMBPTR_EL1.PTR[63:56] must be equal to PMBLIMITR_EL1.LIMIT[63:56].

BRANCH_MISPREDICT

Simulating branch mispredict. Fields:

PC unsigned int

Origin address (or 0 if unavailable).

BRA_DIR

Direct branches, to immediate address. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISSET enum

The instruction set of the branch instruction.

IS_COND bool

Indicates if this is a conditional branch.

IS_HINTED bool

Whether the conditional instruction is hinted.

IS_ISB bool

Whether the instruction is ISB.

IS_LINK bool

Indicates if this is a branch with link.

PC unsigned int

The address of the branch instruction.

TARGET_ISET enum

The instruction set after the branch.

TARGET_PC unsigned int

The address the instruction branches to.

BRA_INDIR

Indirect branches, perhaps to a register. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISET enum

The instruction set of the branch instruction.

IS_COND bool

Indicates if this is a conditional branch.

IS_HINTED bool

Whether the conditional instruction is hinted.

IS_ISB bool

Whether the instruction is ISB.

IS_LINK bool

Indicates if this is a branch with link.

PC unsigned int

The address of the branch instruction.

TARGET_ISET enum

The instruction set after the branch.

TARGET_PC unsigned int

The address the instruction branches to.

BRA_INDIR_CCFAIL

Indirect branches fail, perhaps to a register.

BRA_RET_CCFAIL

Branch return fail, perhaps to a register.

BRB_FILTRATE

Branch Record Captured event.

CACHE_MAINTENANCE_OP

Cache Maintenance Operation. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

DATA unsigned int

Specified MVA or Set/way.

FUNCTION enum

Clean / Invalidate.

SCOPE enum

Affected region.

SIDE enum

Inst / Data.

CCFAIL

Conditional instruction condition check fail. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CCFAIL_UNC

Unconditional instruction condition check fail. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CCPASS

Conditional instruction condition check pass. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CCPASS_UNC

Unconditional instruction condition check pass. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CODE_CACHE_MAINT

Code cache maintenance. Fields:

END_INCL unsigned int

Inclusive end address for VA or PA requests.

NS bool

Non-secure world for PA.

START unsigned int

Start address for VA or PA requests.

TYPE enum

Request type.

COMPILE_BLOCK_END

Last instruction of basic block translated. Fields:

VADDR unsigned int

Address of next instruction after this basic block.

COMPILE_BLOCK_START

First instruction of basic block translated. Fields:

VADDR unsigned int

Address of first instruction in basic block.

COMPILE_INST

ARM instruction compiled. Fields:

DISASS string

Disassembly of the instruction.

ISSET enum

The instruction set of this instruction.

ITSTATE unsigned int

The ITSTATE current for the instruction.

OPCODE unsigned int

The opcode of the instruction.

PC unsigned int

The address of the instruction.

SIZE unsigned int

The size of the instruction in bytes.

CONTEXTIDR

Context ID Register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

NS bool

Secure or nonsecure banked register is accessed.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The new value written.

CONTEXT_SYNC

Called for every context synchronization event. Fields:

ARCH_CSE bool

Architectural Context Synchronization Event.

PC unsigned int

The address of the synchronization instruction.

SYSREG_SYNC bool

System register synchronization event.

CORE_ENDIAN

Core BE8 Big-Endian state changed. Fields:

BE8 bool

Core BE8 Big-Endian state.

CORE_INFO

Static processor attributes. Only triggered by a call to DumpState(). Fields:

ARCH_PROFILE enum

The architecture profile of the core.

CLUSTER_ID unsigned int

The cluster ID of this processor.

CORE_NUM unsigned int

The number of this core in an MP processor.

DETAILED_CORE_FEATURES enum

Detailed core features as a bitfield. Enum values are bitmasks to extract the info from the field.

FPU_VERSION enum

The VFP version implemented by the core.

MEM_ARCH enum

The memory architecture of the core.

MIDR_EL1 unsigned int

MIDR_EL1 value.

NUM_CORES unsigned int

The number of cores in this MP processor.

QUANTUM_SIZE unsigned int

The default quantum size of the core (in ticks).

SECURITY_FEATURES bool

Does the core have security features?.

CORE_LOADS

Processor load accesses. Fields:

ACCELERATOR_ATOMIC_RESPONSE unsigned int

Accelerator atomic response.

ACCESS_TYPE enum

The type of instruction performing the access.

ACQREL enum

Is this an acquire/release.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DATA unsigned int

The data read or written.

ELEMENT_SIZE unsigned int

Width of each element.

ENDIANNESS enum

Byte ordering of this access.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

LOCK enum

Normal, exclusive or locked access.

MECID unsigned int

Memory Encryption Context ID, if applicable.

MECID2 unsigned int

Memory Encryption Context ID of the second page, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PADDR_TAG unsigned int

If tag checked, the physical address tag for the access.

PAGE_SIZE unsigned int

Page size as $\log_2(\text{size in bytes})$.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK, 2=Exclusive Failed, 3=Unsupported.

SIZE unsigned int

Width of the access in bytes. Only required if DATA is not traced.

TAG_CHECKED bool

Is this access tag checked.

TAG_CHECKED2 bool

Is the second page access tag checked.

TRANS bool

Is this a translated access.

VADDR unsigned int

The virtual address of the access.

CORE_REGS

Changes of the core registers R0 to R14. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number, 0 to 14.

MODE enum

Bank of the register accessed.

OLD_VALUE unsigned int

The old value overwritten.

PHYS_ID enum

The physical register accessed.

VALUE unsigned int

The new value written to the register.

CORE_REGS64

Changes of the core registers X0..X30, SP_ELn. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ID enum

The register number, 0-30 for X0-X30, >=32 for SP_ELn.

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

CORE_REGS64_READ

General purpose AArch64 register has been read X0..X30, SP_ELn. Fields:

ID enum

The register number, 0-30 for X0-X30, >=32 for SP_ELn.

VALUE unsigned int

The value read from the register.

CORE_STALLED_TRANSITION

Detect when a core is prevented from progressing PChannel states. Fields:

CURRENT_STATE enum

Current power mode of core.

REASON string

The reason a transition was stalled, and what must be done to unblock it.

TARGET_STATE enum

Desired power mode of core after transition.

CORE_STORES

Processor store accesses. Fields:

ACCELERATOR_ATOMIC_RESPONSE unsigned int

Accelerator atomic response.

ACCESS_TYPE enum

The type of instruction performing the access.

ACQREL enum

Is this an acquire/release.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DATA unsigned int

The data read or written.

ELEMENT_SIZE unsigned int

Width of each element.

ENDIANNESS enum

Byte ordering of this access.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

LOCK enum

Normal, exclusive or locked access.

MECID unsigned int

Memory Encryption Context ID, if applicable.

MECID2 unsigned int

Memory Encryption Context ID of the second page, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PADDR_TAG unsigned int

If tag checked, the physical address tag for the access.

PAGE_SIZE unsigned int

Page size as $\log_2(\text{size in bytes})$.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK, 2=Exclusive Failed, 3=Unsupported.

SIZE unsigned int

Width of the access in bytes. Only required if DATA is not traced.

TAG_CHECKED bool

Is this access tag checked.

TAG_CHECKED2 bool

Is the second page access tag checked.

TRANS bool

Is this a translated access.

VADDR unsigned int

The virtual address of the access.

CP14_READ

System Coprocessor register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP14 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc1 unsigned int

opcode 1.

opc2 unsigned int

opcode 2.

CP14_WRITE

System Coprocessor register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP14 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc1 unsigned int

opcode 1.

opc2 unsigned int

opcode 2.

CP15_READ

System Coprocessor register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP15 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc1 unsigned int

opcode 1.

opc2 unsigned int

opcode 2.

CP15_READ64

System Coprocessor register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP15 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc unsigned int

opcode 1.

CP15_WRITE

System Control Coprocessor register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP15 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc1 unsigned int

opcode 1.

opc2 unsigned int

opcode 2.

CP15_WRITE64

System Coprocessor register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP15 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc unsigned int

opcode 1.

CPSR

CPSR change. (In AArch64 PSTATE represented in SPSR format). Fields:

CORE_NUM unsigned int

Core number in a multi processor.

OLD_VALUE unsigned int

The old CPSR value.

UNKNOWN unsigned int

Bits within the register that have unknown value.

VALUE unsigned int

The new CPSR value.

CRYPTO_SPEC

Every crypto instruction speculatively executed.

CorePowerStateChange

Core power-up state has changed. DISPLAY Core powered %{PU:(up|down)}. Fields:

PU bool

Whether the core powered up.

DATA_CACHE_SET_ALLOC_TAGS

Set allocation tags from DC GVA instruction. Fields:

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

NSDESC unsigned int

The security state of the access.

PADDR unsigned int

The physical address of the access.

PAS enum

The physical address space of the page.

RESPONSE enum

0=Aborted, 1=OK.

SIZE unsigned int

The size of memory in bytes.

TAG unsigned int

The value of the tag being set.

TAGGING_TYPE enum

Type of Memory tagging (Physical or Virtual).

TAG_VADDR unsigned int

The virtual address of the Virtual Tag if Virtual tagging is used; otherwise 0.

VADDR unsigned int

Virtual address used for setting allocation tags.

DATA_CACHE_ZERO

Zero and invalidate cache line from DC ZVA instruction. Fields:

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

NSDESC unsigned int

The security state of the access.

PADDR unsigned int

The physical address of the access.

PADDR_TAG unsigned int

If tag checked, the physical address tag for the access.

PAS enum

The physical address space of the page.

RESPONSE enum

0=Aborted, 1=OK.

SIZE unsigned int

The size of zero'd memory in bytes.

TAG_CHECKED bool

Is this access tag checked.

VADDR unsigned int

Virtual address of the zero'd cache line.

DEBUG_EVENT

Hardware debug support event. Fields:

EVENT enum

Description of event.

VALUE unsigned int

data value.

DEBUG_TARGET_ISET

Hardware debug target ISA. Fields:

TARGET_ISET enum

The target instruction set of the debug state.

DMI_ALLOCATE

DMI allocated. Fields:

ACCESS_DIRECTION enum

READ / WRITE.

PADDR unsigned int

The physical address of the DMI.

PTR unsigned int

DMI range pointer.

SIDE enum

DSIDE / ISIDE.

SIZE unsigned int

The size of DMI.

VADDR unsigned int

The virtual address of the DMI..

DMI_HIT

DMI hit, access used DMI (only fire for small DMI regions i.e. less than a full page). Fields:

ACCESS_DIRECTION enum

READ / WRITE.

PADDR unsigned int

The physical address of the DMI.

SIDE enum

DSIDE / ISIDE.

SIZE unsigned int

The size of the DMI.

VADDR unsigned int

The virtual address of the DMI..

DMI_REVOKE

DMI revoked. Fields:

ACCESS_DIRECTION enum

READ / WRITE.

PADDR unsigned int

The physical address of the DMI.

SIDE enum

DSIDE / ISIDE.

SIZE unsigned int

The size of DMI.

VADDR unsigned int

The virtual address of the DMI..

END_COMPILE

Compilation end. Fields:

END_OF_PAGE_COUNT unsigned int

Number of basic blocks exited due to page boundary since START_COMPILE.

FETCHFAIL_COUNT unsigned int

Number of basic blocks exited due to ifetch failure START_COMPILE.

INST_COUNT unsigned int

Number of instructions compiled since START_COMPILE.

NONSEQ_COUNT unsigned int

Number of basic blocks exited due to non-sequential instructions since START_COMPILE.

PAGE_STRADDLE_COUNT unsigned int

Number of basic blocks exited due to unaligned instructions crossing page since START_COMPILE.

ESB

Error Synchronization Barrier hit. Fields:

ASYNC bool

ESB is async.

IMPLICIT bool

ESB is implicit.

EXCEPTION

Exception taken. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ESR unsigned int

The value assigned to ESR when exception is taken.

ESR_EC unsigned int

The Value assigned to ESR.EC when exception is taken.

IS_PHYSICAL bool

Physical or Virtual exception.

IS_TRAP bool

Exception coming from a trap.

LR unsigned int

The value assigned to the link register.

PC unsigned int

The location where the exception occurred.

PREFERRED_RETURN unsigned int

The preferred return address for the exception.

TAKEN_LOCALLY bool

Whether exception is taken locally(same exception level).

TARGET_ISET enum

The instruction set of the exception handler code.

TARGET_PC unsigned int

The address the exception branches to.

VECTOR enum

The exception vector.

VECTOR_OFFSET unsigned int

The offset of an exception vector from the base address.

EXCEPTION_END

Every exception completed.

EXCEPTION_INFO

Trace human-readable reason of the current exception for data/instruction aborts, floating point exceptions and unknown exceptions. Fields:

DESCRIPTION string

Reason for current exception.

ESR_EC unsigned int

The value of ESR.EC when exception is taken.

EXCEPTION_RAISE

Every exception raised.

EXCEPTION_RETURN

Branches on leaving exception. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISET enum

The instruction set of the branch instruction.

IS_COND bool

Indicates if this is a conditional branch.

IS_HINTED bool

Whether the conditional instruction is hinted.

IS_ISB bool

Whether the instruction is ISB.

IS_LINK bool

Indicates if this is a branch with link.

PC unsigned int

The address of the branch instruction.

TARGET_ISET enum

The instruction set after the branch.

TARGET_PC unsigned int

The address the instruction branches to.

EXCEPTION_RETURN_PREBRANCH

Exception return event that must occur before PC and processor state have updated.

EXCEPTION_START

Every exception started.

FIQ_TAKEN

FIQ taken exception.

FP_EXCEPTIONS_SIGNALLED

Trace which FP exceptions were signalled as a part of the FP operations performed by an instruction. Fields:

FP_EXCEPTIONS_SIGNALLED_MASK unsigned int

Indicates which FP exceptions were signalled by an instruction. This uses the same encoding as the cumulative floating-point exception bitfields in FPSR.

FP_STATE

Floating point state. Fields:

FPSR unsigned int

FPSR register value.

MASK unsigned int

The mask indicating updated cumulative bits.

FREQ_CHANGED

Core frequency changed. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The instruction count of this core.

NEW_FREQ signed int

The new frequency of this core (expressed in Hz).

OLD_FREQ signed int

The old frequency of this core (expressed in Hz).

GPT_LOOKUP

This event is triggered by GPT lookups. Fields:

ABORTED bool

Was the walk successful.

DATA unsigned int

The GPT data read.

DESCADDR unsigned int

The descriptor physical address.

GPI enum

GPI of fetched GPT entry.

LEVEL unsigned int

GPT fetch level.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

PADDR unsigned int

The physical address of the lookup.

PAGEBITS unsigned int

The GPT page bits.

SIDE enum

Inst / Data.

HLT

HLT instruction occurred. Fields:

IMM16 unsigned int

The 16-bit immediate value encoded in HLT instruction.

INST

Every instruction executed. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DEBUG_STATE bool

The instruction is executed in debug state.

DISASS string

Disassembly of instruction.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISET enum

The current instruction set.

ITSTATE unsigned int

The current ITSTATE.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

MODE enum

The mode the core is in.

NS unsigned int

The core's non-secure bit.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

OPCODE unsigned int

The opcode of the instruction.

PADDR unsigned int

The physical address of the instruction.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the instruction.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the page.

PC unsigned int

The address of the instruction.

SECURITY_STATE enum

The core's security state.

SIZE unsigned int

The size of the instruction in bytes.

INST_END

Every instruction completed.

INST_START

Every instruction started. Fields:

DEBUG_STATE bool

The instruction is executed in debug state.

ISET enum

The current instruction set.

MODE enum

The mode the core is in.

NS enum

The current Secure State.

PC unsigned int

The address of the conditional instruction.

SECURITY_STATE enum

The current Security State.

INST_STRADDLE

Instruction straddles boundary. Fields:

DEBUG_STATE bool

The instruction is executed in debug state.

ISET enum

The current instruction set.

MODE enum

The mode the core is in.

NS enum

The current Secure State.

PC unsigned int

The address of the conditional instruction.

SECURITY_STATE enum

The current Security State.

IRQ_TAKEN

IRQ taken exception.

ISB

ISB instruction occurred. Fields:

PC unsigned int

The address of the ISB instruction.

LOAD_MULTIPLE_REGS_LIST

List of destination registers (not source) of a multiple registers load instruction. Fields:

REG string

Register to add into the list.

LOCAL_MONITOR

Local monitor activity. Fields:

PADDR unsigned int

Local Monitor Address.

State enum

State of the monitor (Open/Exclusive).

MAPPING_CREATE

A memory mapping has been created. This is an implementation detail of the model and so may change without notice. Fields:

ASID unsigned int

The ASID used during translation, or 0 if no ASID was used.

IS_VALID bool

Whether the mapping is valid or not.

PADDR unsigned int

Base physical address of region, or 0 if not known.

SIDE enum

I-side or D-side.

SIZE **unsigned int**

Size of the page, or 0 if not known.

VADDR **unsigned int**

Base virtual address of region.

VMID **unsigned int**

The VMID used during translation, or 0 if no VMID was used.

MAPPING_DESTROY

A memory mapping has been destroyed. This is an implementation detail of the model and so may change without notice. Fields:

ASID **unsigned int**

The ASID used during translation.

CAUSE **enum**

The cause of invalidation.

PADDR **unsigned int**

Base physical address of page.

SIDE **enum**

I-side or D-side.

SIZE **unsigned int**

Size of the page.

VADDR **unsigned int**

Base virtual address of page.

VMID **unsigned int**

The VMID used during translation.

MEMMAP_DEBUG_READ

Memory mapped reads to the debug registers. Fields:

ADDR **unsigned int**

address.

CORE_NUM **unsigned int**

Core number in a multi processor.

EXT **bool**

Whether access is from an external device (such as the DAP).

NS **enum**

Secure state of the access.

REG_NAME **string**

Name of the debug register accessed.

UNDEF **bool**

The register accessed is undefined.

VALUE unsigned int

The value read.

MEMMAP_DEBUG_WRITE

Memory mapped writes to the debug registers. Fields:

ADDR unsigned int

address.

CORE_NUM unsigned int

Core number in a multi processor.

EXT bool

Whether access is from an external device (such as the DAP).

NS enum

Secure state of the access.

REG_NAME string

Name of the debug register accessed.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The new value written.

MENTAG_LOADS

Allocation tag memory read; when attributes and system configuration allow tag accesses.

Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

NUM_GRANULES unsigned int

Number of tags to be Loaded/stored.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK.

TAGGING_TYPE enum

Type of Memory tagging (Physical or Virtual).

TAGS unsigned int

The tags read or written. Tags for subsequent granules would be found at subsequent bytes in this field (One byte for each tag).

TAG_VADDR unsigned int

The virtual address of the Virtual Tag if Virtual tagging is used; otherwise 0.

VADDR unsigned int

The virtual address of the access.

MEMTAG_LOAD_INST

Allocation tag load instruction; regardless of tags being actually accessed. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

NUM_GRANULES unsigned int

Number of tags to be Loaded/stored.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK.

TAGGING_TYPE enum

Type of Memory tagging (Physical or Virtual).

TAGS unsigned int

The tags read or written. Tags for subsequent granules would be found at subsequent bytes in this field (One byte for each tag).

TAG_VADDR unsigned int

The virtual address of the Virtual Tag if Virtual tagging is used; otherwise 0.

VADDR unsigned int

The virtual address of the access.

MEMTAG_STORES

Allocation tag memory write; when attributes and system configuration allow tag accesses.
Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

NUM_GRANULES unsigned int

Number of tags to be Loaded/stored.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK.

TAGGING_TYPE enum

Type of Memory tagging (Physical or Virtual).

TAGS unsigned int

The tags read or written. Tags for subsequent granules would be found at subsequent bytes in this field (One byte for each tag).

TAG_VADDR unsigned int

The virtual address of the Virtual Tag if Virtual tagging is used; otherwise 0.

VADDR unsigned int

The virtual address of the access.

MENTAG_STORE_INST

Allocation tag store instruction; regardless of tags being actually accessed. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate

(Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

NUM_GRANULES unsigned int

Number of tags to be Loaded/stored.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK.

TAGGING_TYPE enum

Type of Memory tagging (Physical or Virtual).

TAGS unsigned int

The tags read or written. Tags for subsequent granules would be found at subsequent bytes in this field (One byte for each tag).

TAG_VADDR unsigned int

The virtual address of the Virtual Tag if Virtual tagging is used; otherwise 0.

VADDR unsigned int

The virtual address of the access.

MMU_D128_TTB_READ

This event is triggered by reads caused by a D128 translation table walk. Fields:

ABORTED bool

Was the walk successful.

DATA_HIGH unsigned int

The data upper bits read.

DATA_LOW unsigned int

The data lower bits read.

IPA unsigned int

For stage 1, the IPA of the read.

LEVEL unsigned int

Translation table level.

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

OUTADDR unsigned int

The output physical address.

PADDR unsigned int

The physical address of the read.

PASmem enum

Physical Address Space of the accessed memory.

PASreq enum

Physical Address Space of the requested lookup.

REGIME_EL enum

Entry matches in this translation regime.

REQUESTER enum

The requester of this table walk.

SIDE enum

Inst / Data.

STAGE unsigned int

Translation stage.

MMU_D128_TTB_WRITE

This event is triggered by writes caused by a D128 translation table walk. Fields:

ABORTED bool

Was the walk successful.

DATA_HIGH unsigned int

The data upper bits written.

DATA_LOW unsigned int

The data lower bits written.

IPA unsigned int

For stage 1, the IPA of the write.

IS_HACDBS_ACCESS bool

If the access was due to HACDBS_CLEAN.

LEVEL unsigned int

Translation table level.

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

OUTADDR unsigned int

The output physical address.

PADDR unsigned int

The physical address of the write.

PASmem enum

Physical Address Space of the accessed memory.

PASreq enum

Physical Address Space of the requested lookup.

REGIME_EL enum

Entry matches in this translation regime.

SIDE enum

Inst / Data.

STAGE unsigned int

Translation stage.

MMU_TRANS

Address translation information. Fields:

ASID unsigned int

Address space identifier.

CORE_NUM unsigned int

Core number in a multi processor.

Hyp bool

Entry matches in Hyp state only.

INNERCACHE_RA bool

Is the inner cache allocate on read.

INNERCACHE_TRANSIENT bool

Is the inner write-through transient.

INNERCACHE_TYPE enum

Inner Caching scheme (NC/MB/WA).

INNERCACHE_WA bool

Is the inner cache allocate on write.

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MEMTYPE enum

Memory type.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC enum

Is secure side supposed to access secure or nonsecure memory.

OUTERCACHE_RA bool

Is the outer cache allocate on read.

OUTERCACHE_TRANSIENT bool

Is the outer write-through transient.

OUTERCACHE_TYPE enum

Outer Caching scheme (NC/MB/WA).

OUTERCACHE_WA bool

Is the outer cache allocate on write.

PADDR unsigned int

Physical address of the access.

PAGESIZE unsigned int

Page size as log2(size).

PAS enum

PAS supposed to be accessed is secure or nonsecure memory or Root/Realm if RME is implemented.

SH enum

Shareability.

SIDE enum

Inst / Data.

STAGE1_PERM unsigned int

Stage 1 permissions, [5:3] Privileged access (XWR) [2:0] User access (XWR).

STAGE2_PERM unsigned int

Stage 2 permission mask [5:3] Privileged access (XWR) [2:0] User access (XWR).

VADDR unsigned int

Virtual address of the access.

VMID unsigned int

Virtual machine identifier.

XS bool

XS Attribute.

nG enum

Flag indicating whether ASID will be matched.

MMU_TTB_READ

This event is triggered by reads caused by a translation table walk. Fields:

ABORTED bool

Was the walk successful.

DATA unsigned int

The data read.

DESC_KIND enum

The encoding scheme of the descriptors used for this walk.

IPA unsigned int

For stage 1, the IPA of the read.

LEVEL unsigned int

Translation table level.

LPAE bool

Is this for an LPAE translation (*DEPRECATED*, use DESC_KIND).

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSmem bool

Non secure memory access.

NSreq bool

Non secure request.

OUTADDR unsigned int

The output physical address.

PADDR unsigned int

The physical address of the read.

PASmem enum

Physical Address Space of the accessed memory.

PASreq enum

Physical Address Space of the requested lookup.

REGIME_EL enum

Entry matches in this translation regime.

REQUESTER enum

The requester of this table walk.

SIDE enum

Inst / Data.

STAGE unsigned int

Translation stage.

MMU_TTB_WRITE

This event is triggered by writes caused by a translation table walk. Fields:

ABORTED bool

Was the walk successful.

DATA unsigned int

The data written.

DESC_KIND enum

The encoding scheme of the descriptors used for this walk.

IPA unsigned int

For stage 1, the IPA of the write.

IS_HACDBS_ACCESS bool

If the access was due to HACDBS_CLEAN.

LEVEL unsigned int

Translation table level.

LPAE bool

Is this for an LPAE translation (*DEPRECATED*, use DESC_KIND).

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MPAM_PARTID unsigned int	MPAM Partition ID.
MPAM_PMG unsigned int	MPAM Performance Monitoring Group.
MPAM_SP unsigned int	MPAM Partition ID Space.
NSmem bool	Non secure memory access.
NSreq bool	Non secure request.
OUTADDR unsigned int	The output physical address.
PADDR unsigned int	The physical address of the write.
PASmem enum	Physical Address Space of the accessed memory.
PASreq enum	Physical Address Space of the requested lookup.
REGIME_EL enum	Entry matches in this translation regime.
SIDE enum	Inst / Data.
STAGE unsigned int	Translation stage.
MODE_CHANGE	Mode change. Fields:
CORE_NUM unsigned int	Core number in a multi processor.
MODE enum	The new mode.
NON_SECURE enum	The core's new non-secure bit.
OLD_MODE enum	The old mode.
SECURITY_STATE enum	The new security state.

NOT_TAG_CHECKED_INST

MTE Tag not checked for current instruction.

NV2_CORE_128_LOADS

Processor load accesses for nested virtualization of v9.4 128-bit registers. Fields:

ACCELERATOR_ATOMIC_RESPONSE unsigned int

Accelerator atomic response.

ACCESS_TYPE enum

The type of instruction performing the access.

ACQREL enum

Is this an acquire/release.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DATA unsigned int

The data read or written.

ELEMENT_SIZE unsigned int

Width of each element.

ENDIANNESS enum

Byte ordering of this access.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

LOCK enum

Normal, exclusive or locked access.

MECID unsigned int

Memory Encryption Context ID, if applicable.

MECID2 unsigned int

Memory Encryption Context ID of the second page, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PADDR_TAG unsigned int

If tag checked, the physical address tag for the access.

PAGE_SIZE unsigned int

Page size as $\log_2(\text{size in bytes})$.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

REGNUM_OPERAND unsigned int

Register number corresponding to the instruction operand.

RESPONSE enum

0=Aborted, 1=OK, 2=Exclusive Failed, 3=Unsupported.

SIZE unsigned int

Width of the access in bytes. Only required if DATA is not traced.

TAG_CHECKED bool

Is this access tag checked.

TAG_CHECKED2 bool

Is the second page access tag checked.

TRANS bool

Is this a translated access.

VADDR unsigned int

The virtual address of the access.

NV2_CORE_128_STORES

Processor store accesses for nested virtualization of v9.4 128-bit registers. Fields:

ACCELERATOR_ATOMIC_RESPONSE unsigned int

Accelerator atomic response.

ACCESS_TYPE enum

The type of instruction performing the access.

ACQREL enum

Is this an acquire/release.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DATA unsigned int

The data read or written.

ELEMENT_SIZE unsigned int

Width of each element.

ENDIANNESS enum

Byte ordering of this access.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

LOCK enum

Normal, exclusive or locked access.

MECID unsigned int

Memory Encryption Context ID, if applicable.

MECID2 unsigned int

Memory Encryption Context ID of the second page, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PADDR_TAG unsigned int

If tag checked, the physical address tag for the access.

PAGE_SIZE unsigned int

Page size as $\log_2(\text{size in bytes})$.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

REGNUM_OPERAND unsigned int

Register number corresponding to the instruction operand.

RESPONSE enum

0=Aborted, 1=OK, 2=Exclusive Failed, 3=Unsupported.

SIZE unsigned int

Width of the access in bytes. Only required if DATA is not traced.

TAG_CHECKED bool

Is this access tag checked.

TAG_CHECKED2 bool

Is the second page access tag checked.

TRANS bool

Is this a translated access.

VADDR unsigned int

The virtual address of the access.

NV2_CORE_LOADS

Processor load accesses for v8.4 nested virtualization. Fields:

ACCELERATOR_ATOMIC_RESPONSE unsigned int

Accelerator atomic response.

ACCESS_TYPE enum

The type of instruction performing the access.

ACQREL enum

Is this an acquire/release.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DATA unsigned int

The data read or written.

ELEMENT_SIZE unsigned int

Width of each element.

ENDIANNESS enum

Byte ordering of this access.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

LOCK enum

Normal, exclusive or locked access.

MECID unsigned int

Memory Encryption Context ID, if applicable.

MECID2 unsigned int

Memory Encryption Context ID of the second page, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PADDR_TAG unsigned int

If tag checked, the physical address tag for the access.

PAGE_SIZE unsigned int

Page size as $\log_2(\text{size in bytes})$.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

REGNUM_OPERAND unsigned int

Register number corresponding to the instruction operand.

RESPONSE enum

0=Aborted, 1=OK, 2=Exclusive Failed, 3=Unsupported.

SIZE unsigned int

Width of the access in bytes. Only required if DATA is not traced.

TAG_CHECKED bool

Is this access tag checked.

TAG_CHECKED2 bool

Is the second page access tag checked.

TRANS bool

Is this a translated access.

VADDR unsigned int

The virtual address of the access.

NV2_CORE_STORES

Processor store accesses for v8.4 nested virtualization. Fields:

ACCELERATOR_ATOMIC_RESPONSE unsigned int

Accelerator atomic response.

ACCESS_TYPE enum

The type of instruction performing the access.

ACQREL enum

Is this an acquire/release.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes

([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DATA unsigned int

The data read or written.

ELEMENT_SIZE unsigned int

Width of each element.

ENDIANNESS enum

Byte ordering of this access.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

LOCK enum

Normal, exclusive or locked access.

MECID unsigned int

Memory Encryption Context ID, if applicable.

MECID2 unsigned int

Memory Encryption Context ID of the second page, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PADDR_TAG unsigned int

If tag checked, the physical address tag for the access.

PAGE_SIZE unsigned int

Page size as $\log_2(\text{size in bytes})$.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

REGNUM_OPERAND unsigned int

Register number corresponding to the instruction operand.

RESPONSE enum

0=Aborted, 1=OK, 2=Exclusive Failed, 3=Unsupported.

SIZE unsigned int

Width of the access in bytes. Only required if DATA is not traced.

TAG_CHECKED bool

Is this access tag checked.

TAG_CHECKED2 bool

Is the second page access tag checked.

TRANS bool

Is this a translated access.

VADDR unsigned int

The virtual address of the access.

PERIODIC

Called for every quantum. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The instruction count of this CPU.

PC unsigned int

The address of the next instruction to be executed on this CPU.

PMU_COUNTER_OVERFLOW

PMU counter overflow. Fields:

COUNTER_GROUP enum

Counter group.

COUNTER_RANGE enum

Counter range.

EVENT_ID unsigned int

Event Identifier, field is valid only for Event counter group.

INDEX unsigned int

Counter index (as selected by PMSELR), field is valid only for Event/Cycle counter groups.

INTERRUPT bool

Is interrupt enabled on overflow for this counter.

POINTER_AUTH

Called when execution hits a pointer authentication check. Fields:

AUTH_STATE bool

Authentication result.

KEY_USED enum

Pointer authentication key.

POTINDEX_UPDATE

Value of POT index for current pc changed. Fields:

NEW_POTINDEX unsigned int

New POTIndex.

OLD_POTINDEX unsigned int

Old POTIndex.

PREFETCH_MEMORY64

Prefetch from PRFM or PRFUM instructions. Fields:

PRFOP enum

Prefetch hint.

VADDR unsigned int

Virtual address of the location that should be prefetched.

PRELOAD64_REQUEST

Preload requests from PRFM or PRFUM instructions. Fields:

PRFOP enum

Prefetch hint.

VADDR unsigned int

Virtual address of the location that should be prefetched.

PRELOAD_DATA

Data preload from PLD instruction. Fields:

VADDR unsigned int

Virtual address of the data that should be preloaded.

PRELOAD_INST

Instruction preload from PLI instruction. Fields:

VADDR unsigned int

Virtual address of the instruction that should be preloaded.

PRE_CORE_LOAD

Trace just before a core load. Fields:

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

PRE_CORE_STORE

Trace just before a core store. Fields:

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

PRE_TTB_READ

This event is triggered before reads caused by a translation table walk. Fields:

ADDR unsigned int

The physical address of the read.

PSTATE

PSTATE fields update. Fields:

ALLINT unsigned int

All interrupt mask bit.

BTYPE unsigned int

Branch target identification bit.

CORE_NUM unsigned int

Core number in a multi processor.

DAIF unsigned int

Exception mask bits.

DIT unsigned int

Data Independent Timing bit.

EL unsigned int

Current Exception level field.

EXLOCK unsigned int

Exception return state lock bit.

IL unsigned int

Illegal Execution state bit.

NZCV unsigned int

Condition flags.

PACM unsigned int

PACM bit.

PAN unsigned int

Privileged Access Never state bit.

PM unsigned int

PMU exception mask bit.

PPEND unsigned int

PMU exception pending bit.

SM unsigned int

StreamingMode status bit.

SP unsigned int

Stack pointer register selection bit.

SS unsigned int

Software Step bit.

SSBS unsigned int

Speculative Store Bypass Safe bit.

TCO unsigned int

Tag Check Override bit.

TOUCHED unsigned int

Mask of the fields that have been touched since this trace has last been emitted. Note that touched does not necessarily mean the value changed. The index of the field is the bit index in the mask plus the index of this field plus one.

UAO unsigned int

User Access Override bit.

UINJ unsigned int

Undefined Instruction exceptions bit.

ZA unsigned int

ZA storage and ZT0 register enable bit.

nRW unsigned int

Current Execution state bit.

PSTATE_PACTIVE_OPMODE_TRANSITION

Trace pactive and Opmode mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Operating mode after the transition.

OLD_STATE enum

Operating mode before the transition.

PSTATE_PACTIVE_OPMODE_TRANSITION_DENIED

Trace the denied pactive and Opmode mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Operating mode after the transition.

OLD_STATE enum

Operating mode before the transition.

REASON string

Reason for the transition being denied.

PSTATE_PACTIVE_POWER_TRANSITION

Trace pactive and power mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Power mode after the transition.

OLD_STATE enum

Power mode before the transition.

PSTATE_PACTIVE_POWER_TRANSITION_DENIED

Trace the denied pactive and power mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Power mode after the transition.

OLD_STATE enum

Power mode before the transition.

REASON string

Reason for the transition being denied.

PSTATE_PACTIVE_THREAD_OPMODE_TRANSITION

Trace pactive and Opmode mode transitions for threads on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Operating mode after the transition.

OLD_STATE enum

Operating mode before the transition.

QUANTUM_START

Called at the start of every quantum.

RANGE_PRELOAD_REQUEST

Preload requests from RPRFM. Fields:

COUNT unsigned int

Number of blocks of data to access.

LENGTH unsigned int

Number of contiguous bytes to be accessed in each block.

OP unsigned int

Range prefetch operation .

REUSE unsigned int

Maximum number of bytes to be accessed before next RPRFM .

STRIDE signed int

Stride to advance after accessing LENGTH bytes.

VADDR unsigned int

Base virtual address of range prefetch request.

RUN_STATE

Run state transition. Fields:

INST_COUNT unsigned int

Ticks count at point of transition.

NEW enum

New run state.

OLD enum

Old run state.

SBP_SAMPLE_COLLISION

DISPLAY Statistical profiling samples collided .

SBP_SAMPLE_DISCARDED

DISPLAY Operation discarded by statistical profiling unit when configured in discard mode.
Fields:

BUFFER_POINTER unsigned int

Value of buffer pointer.

DATA unsigned int

Data discarded.

SIZE unsigned int

Size of sample, in bytes.

SBP_SAMPLE_END

DISPLAY Statistical profiling of an operation ended.

SBP_SAMPLE_POP

DISPLAY Operation which might be sampled by statistical profiling.

SBP_SAMPLE_RECORDED

DISPLAY Operation sampled by statistical profiling has been written to the buffer at %{BUFFER_ADDRESS} with size %{SIZE} and data %{DATA}. Fields:

BUFFER_ADDRESS unsigned int

Virtual Address of start of buffer write.

DATA unsigned int

Data written to the buffer.

SIZE unsigned int

Size of sample written to buffer, in bytes.

SBP_SAMPLE_START

DISPLAY Statistical profiling of an operation started.

SBP_SAMPLE_TAKEN

DISPLAY Operation sampled by statistical profiling after filtering.

SEMIHOSTING_CALL

Call of a semihost function occurred. Fields:

PC unsigned int

The program counter after the semihosting call.

SEMIHOSTING_PRECALL

About to call semihost. Fields:

EL2 bool

Use EL2 translation regime.

NS bool

Is Non-Secure.

REG_WIDTH unsigned int

The current register width in bytes.

SIGNAL

External signal state change. Fields:

SIGNAL enum

Signal that changed.

STATE bool

Signal asserted state.

SOFTWARE_STEP

Return the Debug Software Step state. Fields:

STATE enum

Software Step state.

SPSR

SPSR change. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

MODE enum

Which of the banked SPSR registers is written.

OLD_VALUE unsigned int

The old SPSR value.

VALUE unsigned int

The new SPSR value.

START_COMPILE

Compilation started. Fields:

VADDR unsigned int

Instruction where compilation begins.

STG_MULTIPLE_MICRO_OPS_ADDRESS_RANGE

Address range for MTE stores with multiple uops. Fields:

ADDR_END unsigned int

Final address.

ADDR_START unsigned int

Starting address.

STORE_MULTIPLE_REGS_ADDRESS_RANGE

Address range for stores with multiple registers. Fields:

ADDR_END unsigned int

Final address.

ADDR_START unsigned int

Starting address.

SUBOPTIMAL_LDST_RETIRED

suboptimal load/store. Fields:

ACCESS_SIZE unsigned int

Log2 of the access width in bytes used for alignment checking (1: half-word, 2: word ...).

OPTIMAL_ALIGN_SELECTOR unsigned int

Selector to decide the optimal alignment size (1: access width, 2: 4-bytes, 3: 8-bytes, ..., 12: 4-kbytes).

VADDR unsigned int

The virtual address of the access.

SVE_INST_SPEC

SVE operations speculatively executed.

SVE_LD_RETIRED

SVE load instructions architecturally executed.

SVE_LOADS

SVE load accesses. Fields:

ACCELERATOR_ATOMIC_RESPONSE unsigned int

Accelerator atomic response.

ACCESS_TYPE enum

The type of instruction performing the access.

ACQREL enum

Is this an acquire/release.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DATA unsigned int

The data read or written.

ELEMENT_SIZE unsigned int

Width of each element.

ENDIANNESS enum

Byte ordering of this access.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

LOCK enum

Normal, exclusive or locked access.

MECID unsigned int

Memory Encryption Context ID, if applicable.

MECID2 unsigned int

Memory Encryption Context ID of the second page, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PADDR_TAG unsigned int

If tag checked, the physical address tag for the access.

PAGE_SIZE unsigned int

Page size as $\log_2(\text{size in bytes})$.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK, 2=Exclusive Failed, 3=Unsupported.

SIZE unsigned int

Width of the access in bytes. Only required if DATA is not traced.

TAG_CHECKED bool

Is this access tag checked.

TAG_CHECKED2 bool

Is the second page access tag checked.

TRANS bool

Is this a translated access.

VADDR unsigned int

The virtual address of the access.

SVE_MEM_PREFETCH

Scalable Vector Extension memory prefetch. Fields:

CACHE_LEVEL enum

The preferred cache level.

HINT enum

What the memory is expected to be used for.

SIZE unsigned int

The prefetch size in bytes.

TEMPORAL bool

Whether the data is temporal.

VADDR unsigned int

The virtual address of the prefetch.

SVE_REG_READ

Scalable Vector Extension register read. Fields:

ID enum

The register identifier.

SM bool

Whether the PE is in Streaming Mode.

VALUE unsigned int

The read data.

SVE_REG_UPDATE

Scalable Vector Extension register update. Fields:

ID enum

The register identifier.

SM bool

Whether the PE is in Streaming Mode.

VALUE unsigned int

The written data.

SVE_REG_WRITE

Scalable Vector Extension register write. Fields:

ID enum

The register identifier.

SM bool

Whether the PE is in Streaming Mode.

VALUE unsigned int

The written data.

SVE_STORES

SVE store accesses. Fields:

ACCELERATOR_ATOMIC_RESPONSE unsigned int

Accelerator atomic response.

ACCESS_TYPE enum

The type of instruction performing the access.

ACQREL enum

Is this an acquire/release.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DATA unsigned int

The data read or written.

ELEMENT_SIZE unsigned int

Width of each element.

ENDIANNESS enum

Byte ordering of this access.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

LOCK enum

Normal, exclusive or locked access.

MECID unsigned int

Memory Encryption Context ID, if applicable.

MECID2 unsigned int

Memory Encryption Context ID of the second page, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PADDR_TAG unsigned int

If tag checked, the physical address tag for the access.

PAGE_SIZE unsigned int

Page size as $\log_2(\text{size in bytes})$.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK, 2=Exclusive Failed, 3=Unsupported.

SIZE unsigned int

Width of the access in bytes. Only required if DATA is not traced.

TAG_CHECKED bool

Is this access tag checked.

TAG_CHECKED2 bool

Is the second page access tag checked.

TRANS bool

Is this a translated access.

VADDR unsigned int

The virtual address of the access.

SVE_ST_RETIRED

SVE store instructions architecturally executed.

SVE_UNALIGNED_LDST_RETIRED

SVE load or store instructions architecturally executed that would cause alignment fault if A bit is set.

SVE_Z_PRED_MEM_ACCESS_INFO

Information about SVE/SME predicated accesses to Z registers (non Gather/Scatter). Fields:

MEM_ELEM_SIZE unsigned int

Width of elements accessed in memory (in bytes).

RnW bool

Whether this is a Read not a Write.

VA unsigned int

Virtual base address computed for accesses.

SYNC

Called for every synchronization, that is every quantum break. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The instruction count of this CPU.

LOCAL_QUANTUM unsigned int

The local quantum of this CPU.

LOCAL_TIME unsigned int

The local time of this CPU.

SYNC_REASON enum

Reason for sync.

SYSCALL

System call instruction executed. Fields:

IMM unsigned int

Immediate value of the system call instruction.

TYPE enum

System call type.

VADDR unsigned int

Instruction that caused the system call.

SYSREG_READ128

System Coprocessor 128 bit register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

REGNUM enum

Internal register number.

REGNUM_PHYS enum

Internal register number physically accessed (e.g. in presense of VHE redirection).

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_READ64

System Coprocessor register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REGNUM enum

Internal register number.

REGNUM_OPERAND enum

Register number corresponding to the instruction operand.

REGNUM_PHYS enum

Internal register number physically accessed (e.g. in presense of VHE redirection).

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_UPDATE128

Triggers when the system updates a register. Fields:

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG enum

Register number.

UNKNOWN unsigned int

Bits of the register which became unknown.

VALUE unsigned int

Value written to the register.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_UPDATE32

Triggers when the system updates a register. Fields:

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG enum

Register number.

UNKNOWN unsigned int

Bits of the register which became unknown.

VALUE unsigned int

Value written to the register.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_UPDATE64

Triggers when the system updates a register. Fields:

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG enum

Register number.

UNKNOWN unsigned int

Bits of the register which became unknown.

VALUE unsigned int

Value written to the register.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_WRITE128

System Coprocessor 128 bit register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

REGNUM enum

Internal register number.

REGNUM_PHYS enum

Internal register number physically accessed (e.g. in presense of VHE redirection).

UNDEF bool

The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_WRITE64

System Coprocessor register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REGNUM enum

Internal register number.

REGNUM_OPERAND enum

Register number corresponding to the instruction operand.

REGNUM_PHYS enum

Internal register number physically accessed (e.g. in presense of VHE redirection).

UNDEF bool

The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

TRBU_BUFFER_WRAP

DISPLAY Trace buffer current write pointer wrapped.

TRBU_TRIGGER_EVENT

DISPLAY Trace buffer Trigger Event.

TRBU_WRITE

DISPLAY PETU trace data has been written to the TRBU memory buffer at %{BUFFER_ADDRESS} with size %{SIZE} and data %{DATA}. Fields:

BUFFER_ADDRESS unsigned int

Virtual Address of start of buffer write.

DATA unsigned int

Data written to the buffer.

SIZE unsigned int

Size of sample written to buffer, in bytes.

UNALIGNED_LDST_RETIRED

Processor unaligned load/store. Fields:

ACCESS_SIZE unsigned int

Log2 of the access width in bytes used for alignment checking (1=>half-word, 2->word ...).

NUMBER_OF_BEATS unsigned int

Number of accesses (beats) in this burst (Total size in bytes is ACCESS_SIZE * NUMBER_OF_BEATS).

VADDR unsigned int

The virtual address of the access.

VFP_D_REGS

VFP/NEON D 64 bit register write. Fields:

ALIASING enum

Type of register aliasing used.

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a write to the corresponding bit in the register. For example, 0x1 would indicate a write to VALUE[0].

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

VFP_D_REGS_READ

VFP/NEON D 64 bit register read. Fields:

ID unsigned int

The register number.

VALUE unsigned int

The current value of the register.

VFP_Q_REGS

VFP/NEON Q 128 bit register write. Fields:

ALIASING enum

Type of register aliasing used.

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a write to the corresponding byte in the register. For example, 0x1 would indicate a write to VALUE[7:0].

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

VFP_Q_REGS_READ

VFP/NEON Q 128 bit register read. Fields:

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a read to the corresponding byte in the register. For example, 0x1 would indicate a read to VALUE[7:0].

VALUE unsigned int

The current value of the register.

VFP_SYS_REGS

Writes to the VFP/NEON units system registers. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ID enum

Which VFP system register is written.

MASK unsigned int

The mask indicating updated cumulative bits.

OLD_VALUE unsigned int

The register's old value overwritten.

VALUE unsigned int

The new value written to the VFP system register.

VFP_S_REGS

VFP/NEON S 32 bit register write. Fields:

ALIASING enum

Type of register aliasing used.

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a write to the corresponding bit in the register. For example, 0x1 would indicate a write to VALUE[0].

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

VFP_S_REGS_READ

VFP/NEON S 32 bit register read. Fields:

ID unsigned int

The register number.

VALUE unsigned int

The current value of the register.

VIRTUAL_TAG_ALLOCATED

Virtual tag information when allocated. Fields:

TAG_VADDR unsigned int

Virtual tag virtual address.

VADDR unsigned int

Data Virtual address.

VIRTUAL_TAG unsigned int

Virtual tag value.

WAYPOINT

Signals end of basic block execution. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ISET enum

Origin instruction set.

IS_COND bool

Indicates if this is a conditional waypoint.

PC unsigned int

Origin address (or 0 if unavailable).

TAKEN bool

Indicates if this waypoint was taken.

TARGET unsigned int

Destination address.

TARGET_ISET enum

Destination instruction set.

WFE_END

WFE ended. Fields:

INST_COUNT unsigned int

Ticks count when leaving WFE.

WFE_EVENT_REGISTER

WFE event register status: set/clear, reason. Fields:

INST_COUNT unsigned int

Ticks count.

REASON enum

Reason for set/clear. Only REASON==0 (Cleared by WFE) clears the bit, all other reasons set it.

WFE_IGNORED

WFE ignored. Fields:

EVENT bool

This WFE was ignored because the event register was set.

INST_COUNT unsigned int

Ticks count when ignoring WFE.

TRAPPED bool

This WFE was trapped.

WFE_START

WFE entered. Fields:

INST_COUNT unsigned int

Ticks count when entering WFE.

WFI_END

WFI ended. Fields:

INST_COUNT unsigned int

Ticks count when leaving WFI.

WFI_IGNORED

WFI ignored. Fields:

DISABLED bool

This WFI was ignored because WFI is disabled.

INST_COUNT unsigned int

Ticks count when ignoring WFI.

REASON enum

specifies reason why WFI trace was ignored.

TRAPPED bool

This WFI was trapped.

WFI_START

WFI entered. Fields:

INST_COUNT unsigned int

Ticks count when entering WFI.

WFI_WAKEUP

WFI wakeup. Fields:

INST_COUNT unsigned int

Ticks count when WFI wakeup occurred.

REASON enum

Reason for wakeup.

2.70 ARM_Neoverse-V3AE

This section describes the trace sources.

AA64_ASE_SVE_REGS

Modification of AArch64 Advanced SIMD or SVE Vector Registers (via SIMD or Scalar operations), SVE Predicate Registers and ZT0. Fields:

ID enum

The register identifier.

MASK unsigned int

Bitmask of the register to signal the modified bits in the VALUE field.

SM bool

Whether the PE is in Streaming Mode.

VALUE unsigned int

Value of AArch64 vector register.

ASYNC_MEMORY_FAULT

Context ID Register write. Fields:

FAULT unsigned int

Fault status present in ESR.ISS encoding for Serror in ARCH64 and present in DFSR encoding for external abort format in ARCH32.

PADDR unsigned int

Physical Address (or 0 if unavailable).

VADDR signed int

Virtual Address (or 0 if unavailable).

ATOMIC_END_ACCESS

Bus trace access for atomics. Fields:

ACCESS_FAIL bool

Memory access failed.

ADDR unsigned int

The virtual address of the access.

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory

attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

COMPARE_VALUE unsigned int

Compare value for CAS.

LOAD_VALUE unsigned int

Loaded values.

NSDESC unsigned int

The security state of the access.

OPERAND_VALUE unsigned int

Operation's operand.

OPERATION enum

Operation type.

PADDR unsigned int

The physical address of the access.

PRIV bool

Is this a privileged access?.

ATOMIC_START_ACCESS

Bus trace access for atomics. Fields:

ADDR unsigned int

The virtual address of the access.

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

COMPARE_VALUE unsigned int

Compare value for CAS.

NSDESC unsigned int

The security state of the access.

OPERAND_VALUE unsigned int

Operation's operand.

OPERATION enum

Operation type.

PADDR unsigned int

The physical address of the access.

PRIV bool

Is this a privileged access?.

ArchMsg.Error.exit_code

DISPLAY exit code: %{EXITCODE} component kind: %{KIND} reason: %{REASON}. Fields:

EXITCODE signed int

Exit code to be returned.

KIND string

Component kind that invoked the exit code trace.

REASON string

Reason why the exit code trace was invoked (optional).

ArchMsg.Warning.BRBInjectionUnknown

DISPLAY BRB INJ with uninitialized injection registers.

ArchMsg.Warning.IgnoredWarmResetRequest

Event fires when software attempts to request warm reset through write to deprecated DBGPRCR.CWRR.

ArchMsg.Warning.RAS_ErrselrUnpredictableState

DISPLAY ERRSELR is in unpredictable state due to value greater than number_of_error_records written to it.

ArchMsg.Warning.RAS_UnpredictableStateWrite

DISPLAY %{REG_NAME} write ignored due to unpredictable ERRSELR state. Fields:

REG_NAME string

Name of register being written.

ArchMsg.Warning.ReportedRevisionRegisterRead

DISPLAY Reported revision and patch do not match that of the cpu. Fields:

ORIGINAL_VALUE unsigned int

The value the register would have based on the modelled revision.

REGISTER string

The name of the register that has ben read.

REPORTED_VALUE unsigned int

The value the register has been overridden to.

ArchMsg.Warning.Unpredictably Indexed PM Event Register

TAGS UNPREDICTABLE DISPLAY Event %{IsDirect:(Selector value|Register)} %{SEL} is >= %{N}. Fields:

IsDirect unsigned int

Direct Access rather than indirect.

N unsigned int

Number of Accessible Registers.

SEL unsigned int

Selector.

ArchMsg.Warning.cache_contents_unknown

A cache has been activated without being invalidated and may contain **UNKNOWN** content. DISPLAY %{SIDE:(D|I)}-Cache was enabled in %{REGIME:(SECURE|NON_SECURE|HYP)} regime but was not invalidated since power-on: cache lines could contain **UNKNOWN** content %{SIDE:(Note: D-side caches must invalidate SECURE lines even if the cache is only used by NON_SECURE code in order to ensure that unexpected dirty lines tagged SECURE are not naturally evicted)}. Fields:

REGIME unsigned int

Current translation regime.

SIDE unsigned int

0 d-side cache or unified cache, 1 i-side.

ArchMsg.Warning.dcimvac_matches_watchpoint

TAGS **UNPREDICTABLE** DISPLAY Watchpoints matching an AArch32 DCIMVA are implementation defined.

ArchMsg.Warning.decode_fieldmismatch

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with different fields %{FIELD1} and %{FIELD2}. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_fracbitsnegative

TAGS **UNPREDICTABLE** DISPLAY VCVT Instruction is unpredictable with a negative number of fraction bits.

ArchMsg.Warning.decode_initblock

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable in an IT block.

ArchMsg.Warning.decode_initblocknotlast

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable in an IT block when not the last.

ArchMsg.Warning.decode_invalid_condition

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable when the condition is not AL.

ArchMsg.Warning.decode_invalidfieldcombination

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with the values in %{FIELD1} and %{FIELD2}. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_invalidvalue

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with the value in %{FIELD}. Fields:

FIELD enum

Field name.

ArchMsg.Warning.decode_mem_hint_unallocated

TAGS **UNPREDICTABLE** DISPLAY There is no memory hint allocated to this bit pattern.

ArchMsg.Warning.decode_registermatch

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with %{FIELD1} and %{FIELD2} as the same register. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_registermismatch

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with %{FIELD1} and %{FIELD2} as different registers. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_registeroutofrange

TAGS **UNPREDICTABLE** DISPLAY Floating-point instruction is unpredictable indexing beyond the end of the register bank.

ArchMsg.Warning.decode_sbzsbo

TAGS **UNPREDICTABLE** DISPLAY Reserved bits in the instruction are not canonical.

ArchMsg.Warning.decode_transactiontoobig

TAGS **UNPREDICTABLE** DISPLAY Attempt to perform a memory transaction of more than 128 bytes.

ArchMsg.Warning.decode_unpred_other

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable.

ArchMsg.Warning.decode_unpreduseofpc

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable when %{FIELD} is PC. Fields:

FIELD enum

Field name.

ArchMsg.Warning.decode_unpreduseofr13

TAGS UNPREDICTABLE DISPLAY Instruction is unpredictable when %{FIELD} is R13. Fields:

FIELD enum

Field name.

ArchMsg.Warning.decode_unpreduseofr13inreglist

TAGS UNPREDICTABLE DISPLAY Instruction is unpredictable with R13 in register-list.

ArchMsg.Warning.decode_useofd16orhigher

TAGS UNPREDICTABLE DISPLAY Attempt to use register D16 or higher where it is not allowed.

ArchMsg.Warning.decode_writebackandbaseinlist

TAGS UNPREDICTABLE DISPLAY Instruction is unpredictable with write back when the base register is in the transfer list.

ArchMsg.Warning.decode_zeroregistersinlist

TAGS UNPREDICTABLE DISPLAY Instruction is unpredictable when no registers are to be transferred.

ArchMsg.Warning.misaligned_ptrs

DISPLAY TRBPTR_EL1(%{TRBPTR_EL1}) or TRBTRG_EL1(%{TRBTRG_EL1}) are not aligned as per TRBIDR_EL1.Align(%{TRBIDR_EL1}). Fields:

TRBIDR_EL1 unsigned int

TRBIDR_EL1.Align value.

TRBPTR_EL1 unsigned int

Current write pointer.

TRBTRG_EL1 unsigned int

Trigger pointer.

ArchMsg.Warning.out_of_range_ptrs

DISPLAY TRBPTR_EL1(%{TRBPTR_EL1}) is out of range as per TRBBASER_EL1(%{TRBBASER_EL1}) or TRBLIMITR_EL1.LIMIT(%{TRBLIMITR_EL1}). Fields:

TRBBASER_EL1 unsigned int

Base pointer.

TRBLIMITR_EL1 unsigned int

Limit pointer.

TRBPTR_EL1 unsigned int

Current write pointer.

ArchMsg.Warning.recursive_branch

An instruction is performing a branch that targets the same instruction. If this was intended then a WFI might be more effective DISPLAY recursive branch detected at PC=%{PC}.

PRIMARY KEY PC. Fields:

PC unsigned int

Address of instruction causing the branch.

ArchMsg.Warning.recursive_exception

An instruction has generated an exception that targets the same instruction. DISPLAY recursive %{TYPE} exception detected at PC=%{PC} (CPSR=%{CPSR}). PRIMARY KEY PC. Fields:

CPSR unsigned int

Processor state.

PC unsigned int

Address of instruction causing the exception.

TYPE enum

Type of exception.

ArchMsg.Warning.reentrant_vector_catch

TAGS **UNPREDICTABLE** DISPLAY Vector catch debug event on Prefetch or Data abort vector %{ADDR}. Fields:

ADDR unsigned int

vector address.

ArchMsg.Warning.reserved_it_state

TAGS **UNPREDICTABLE** DISPLAY Execution with reserved IT state %{ITSTATE}. Fields:

ITSTATE unsigned int

reserved IT state.

ArchMsg.Warning.tlb_contents_unknown

Translation has been enabled without invalidating the TLB. DISPLAY Translation was enabled in the %{PAS:(Secure|Non-secure|Root|Realm)} EL%{EL:d} translation regime (vmid=%{VMID:x}) without invalidating %{INVALIDITY:(|d-side |i-side |)}entries since power-on: %{INVALIDITY:(|D||)}TLB could contain **UNKNOWN** content. Fields:

EL unsigned int

Exception level of translation regime.

INVALIDITY unsigned int

Which TLB may be invalid (if TLBs are separate).

NON_SECURE bool

Whether the translation regime is non-secure.

PAS unsigned int

Physical address space of translation regime.

VMID unsigned int

Current VMID, if applicable.

ArchMsg.Warning.unknown_DLR_DSPSR

Exiting debug state with unknown DLR or DSPSR.

ArchMsg.Warning.unknown_ELR_SPSR

Returning from debug exception with unknown ELR or SPSR.

ArchMsg.Warning.unpred_FAR_on_LRCPC3_non_atomic_mem_access

TAGS **UNPREDICTABLE** Unpredictable FAR/FSTATUS on LRCPC3 pair instructions resulting in two separate memory accesses
 DISPLAY Unpredictable FAR on LRCPC3 non atomic memory access at %{ADDR1} with FSTATUS %{FSTATUS1} or %{ADDR2} with FSTATUS %{FSTATUS2}. Fields:

ADDR1 unsigned int

Possible FAR address 1.

ADDR2 unsigned int

Possible FAR address 2.

FSTATUS1 unsigned int

Possible fault status value 1.

FSTATUS2 unsigned int

Possible fault status value 2.

ArchMsg.Warning.unpred_async_tag_check_fail_on_sve_speculative_load_do_update_ffr

Unpredictable set to FALSE of the FFR element associated to SVE speculative load with an async tag check fail TAGS **UNPREDICTABLE**.

ArchMsg.Warning.unpredictable_a32_breakpoint

TAGS **UNPREDICTABLE** DISPLAY Unpredictable breakpoint on a A32 instruction at %{ADDR1} Offset %{OFFSET} BAS %{BAS} for a %{IS_ADDRESS_MISMATCH_BREAKPOINT:(Mismatch|Match)} Breakpoint number %{N:d} Breakpoint address %{ADDR2}. Fields:

ADDR1 unsigned int

Address.

ADDR2 unsigned int

Breakpoint address.

BAS unsigned int

BAS field.

IS_ADDRESS_MISMATCH_BREAKPOINT bool

Is a Mismatch breakpoint, not a Match breakpoint.

N unsigned int

Breakpoint number.

OFFSET unsigned int

Always has a value of zero.

ArchMsg.Warning.unpredictable_banked_register_access

TAGS **UNPREDICTABLE** DISPLAY Access to banked register (R=%{R:d} SYSm=%{SYSM:2x}) is unpredictable. Fields:

R unsigned int

R bit.

SYSM unsigned int

SYSm.

ArchMsg.Warning.unpredictable_ccfail_undef

TAGS **UNPREDICTABLE** DISPLAY An instruction %{OPCODE} at VA=%{VADDR} failed its condition codes check but would otherwise have caused a non-data-dependent trap or undefined exception. Behaviour is implementation-defined choice of exception or nop. Fields:

OPCODE unsigned int

opcode of the instruction.

VADDR unsigned int

Virtual address of the instruction being executed.

ArchMsg.Warning.unpredictable_ici_bits

TAGS **UNPREDICTABLE** DISPLAY ICI bits do not apply or are architecturally unpredictable for instruction.

ArchMsg.Warning.unpredictable_t32_breakpoint

TAGS **UNPREDICTABLE** DISPLAY Unpredictable breakpoint on a T32 instruction at %{ADDR1} Offset %{OFFSET} BAS %{BAS} for a %{IS_ADDRESS_MISMATCH_BREAKPOINT:(Mismatch|Match)} Breakpoint number %{N:d} Breakpoint address %{ADDR2}. Fields:

ADDR1 unsigned int

Address.

ADDR2 unsigned int

Breakpoint address.

BAS unsigned int

BAS field.

IS_ADDRESS_MISMATCH_BREAKPOINT bool

Is a Mismatch breakpoint, not a Match breakpoint.

N unsigned int

Breakpoint number.

OFFSET unsigned int

Indicates which halfword of a 32-bit Thumb instruction the breakpoint is set on. 0 for the first halfword or if the instruction is 16 bits wide, or 1 for the second halfword.

ArchMsg.Warning.unpredictable_t32_vector_catch

TAGS **UNPREDICTABLE** DISPLAY Unpredictable Vector catch exception on a T32 instruction at %{ADDR} Offset %{OFFSET}. Fields:

ADDR unsigned int

Address.

OFFSET unsigned int

Indicates which halfword of a 32-bit Thumb instruction the vector catch exception is set on. 0 for the first halfword or if the instruction is 16 bits wide, or 1 for the second halfword.

ArchMsg.Warning.unpredictable_watchpoint_far

TAGS **UNPREDICTABLE** DISPLAY The value for FAR/EDWAR for hitting this watchpoint may be anywhere in the range %{LOWER_BOUND} to %{UPPER_BOUND}. Fields:

LOWER_BOUND unsigned int

the lowest address accessed by the instruction that triggered the watchpoint.

UPPER_BOUND unsigned int

the highest watchpointed address accessed by that instruction.

ArchMsg.Warning.unpredictable_watchpoint_far_isncmp

DISPLAY The value for FAR/EDWAR for hitting this watchpoint may be anywhere in the range %{LOWER_BOUND} to %{UPPER_BOUND}. Fields:

LOWER_BOUND unsigned int

the lowest address accessed by the instruction that triggered the watchpoint.

UPPER_BOUND unsigned int

the highest watchpointed address accessed by that instruction.

ArchMsg.Warning.warning_AdvSIMDExpandImmUnexpectedZero

TAGS **UNPREDICTABLE** DISPLAY AdvSIMDExpandImm may treat this immediate value as **UNPREDICTABLE**.

ArchMsg.Warning.warning_ConditionalSMC

TAGS **UNPREDICTABLE** DISPLAY SMC instruction has **UNPREDICTABLE** effects when conditional and when combined with traps.

ArchMsg.Warning.warning_TLBID_out_of_range

TLBID out of range in TLBI according to the configuration of NIS, NVIS, NOS and NVOS. Fields:

NIS unsigned int

NIS.

NOS unsigned int

NOS.

NVIS unsigned int

NVIS.

NVOS unsigned int

NVOS.

TLBID unsigned int

The TLBID value used in the TLBI instruction.

ArchMsg.Warning.warning_TTL_or_TG_mismatches_current_translation_granule

TTL or TG mismatches current translation granule in TLBI VA/RVA/IPA/RIPA. Fields:

CURRENT_TG unsigned int

Current TG as indicated by TCR_ELx or VTCR_EL2.

TLBI_TG unsigned int

TG (or TTL[3:2]) used in TLBI instruction.

ArchMsg.Warning.warning_access_crosses_page_boundary

TAGS **UNPREDICTABLE** DISPLAY Access crosses page boundary.

ArchMsg.Warning.warning_access_crosses_page_boundary_has_fault

TAGS **UNPREDICTABLE** DISPLAY Access that crosses page boundary has fault(s).

ArchMsg.Warning.warning_access_crosses_page_boundary_has_fault_on_both_pages

TAGS **UNPREDICTABLE** DISPLAY Access that crosses page boundary has a fault on each page.
Fields:

ADDR1 unsigned int

FAR address on first page.

ADDR2 unsigned int

FAR address on second page.

FSTATUS1 unsigned int

Fault status on first page.

FSTATUS2 unsigned int

Fault status on second page.

ArchMsg.Warning.warning_access_crosses_page_boundary_spanning_different_memory

TAGS **UNPREDICTABLE** DISPLAY Access at address %{ADDR} crosses page from MEM_TYPE %{MEMTYPE_PAGE1} to %{MEMTYPE_PAGE2}. Fields:

ADDR unsigned int

address of access crossing page.

MEMTYPE_PAGE1 signed int

First Page's memory type (0-Normal, 1-Non-Normal).

MEMTYPE_PAGE2 signed int

Second Page's memory type (0-Normal, 1-Non-Normal).

ArchMsg.Warning.warning_access_wraps_around_memory

TAGS **UNPREDICTABLE** DISPLAY Access wraps around memory in %{WIDTH:d}bit mode.
Fields:

WIDTH unsigned int

Address width.

ArchMsg.Warning.warning_bcr_linking_status

TAGS **UNPREDICTABLE** DISPLAY DBGBCR.{BT2,BT} combination gives unpredictable behaviour.

ArchMsg.Warning.warning_bcr_mask_reserved

TAGS **UNPREDICTABLE** DISPLAY DBGBCR_ELn.MASK is set to a reserved value.

ArchMsg.Warning.warning_bvr_masked_bit_not_zero

TAGS **UNPREDICTABLE** DISPLAY DBGBVRn_EL1.VA contains a bit masked by DBGBCRn_EL1.MASK that is not zero.

ArchMsg.Warning.warning_ccsidr_unimplemented_level

TAGS **UNPREDICTABLE** DISPLAY CCSIDR read while CSSELR %{CSSELR:x} points to an unimplemented cache level. Fields:

CSSELR unsigned int

current effective value of CSSELR.

ArchMsg.Warning.warning_change_to_ns_when_tge_set

TAGS **UNPREDICTABLE** DISPLAY Attempting to change NS to 1 when HCR.TGE = 1.

ArchMsg.Warning.warning_contiguous_bit_check_abort

TAGS **UNPREDICTABLE** DISPLAY An abort occurred whilst attempting to check TLB entry is contiguous at %{ENTRY_ADDR}. Fields:

ENTRY_ADDR unsigned int

address of conflicting TLB entry.

ArchMsg.Warning.warning_contiguous_bit_error

TAGS **UNPREDICTABLE** DISPLAY TLB entry at %{ENTRY_ADDR} was not contiguous with entry at %{CONTIG_ADDR} Contents are %{ENTRY_DATA} but expected %{CONTIG_DATA}. Fields:

CONTIG_ADDR unsigned int

address of the TLB first read, with which this entry is expected to be contiguous.

CONTIG_DATA unsigned int

expected contents based on the entry at CONTIG_ADDR.

ENTRY_ADDR unsigned int

address of conflicting TLB entry.

ENTRY_DATA unsigned int

contents of conflicting TLB entry.

ArchMsg.Warning.warning_contiguous_bit_error_D128

TAGS **UNPREDICTABLE** DISPLAY TLB entry at %{ENTRY_ADDR} was not contiguous with entry at %{CONTIG_ADDR} Contents are %{ENTRY_DATA_HI} %{ENTRY_DATA_LO} but expected %{CONTIG_DATA}. Fields:

CONTIG_ADDR unsigned int

address of the TLB first read, with which this entry is expected to be contiguous.

CONTIG_DATA unsigned int

expected contents based on the entry at CONTIG_ADDR.

ENTRY_ADDR unsigned int

address of conflicting TLB entry.

ENTRY_DATA_HI unsigned int

Higher bit contents of conflicting TLB entry.

ENTRY_DATA_LO unsigned int

Lower bit contents of conflicting TLB entry.

ArchMsg.Warning.warning_cp10_cp11_mismatch

TAGS **UNPREDICTABLE** DISPLAY Unpredictable configuration of CP10 and CP11 controls at EL
%{EL:d}. Fields:

EL unsigned int

EL owning the control.

ArchMsg.Warning.warning_cp10_cp11_reserved_value

TAGS **UNPREDICTABLE** DISPLAY Unpredictable configuration of CPACR CP10 (%{CP10}) or
CP11 (%{CP11}). Fields:

CP10 unsigned int

CP10 access.

CP11 unsigned int

CP11 access.

ArchMsg.Warning.warning_csselr_level_out_of_range

TAGS **UNPREDICTABLE** DISPLAY CSSELR.Level out of range, written value %{LEVEL} with only
%{IMPLEMENTED:d} levels implemented. Fields:

IMPLEMENTED unsigned int

number of cache levels implemented.

LEVEL unsigned int

written value of CSSELR.Level.

TYPE enum

The type of cache selected by CSSELR.InD and TnD.

ArchMsg.Warning.warning_debug_flow_control_bits_not_obeyed

TAGS **UNPREDICTABLE** DISPLAY Flow control bits for %{REGISTER} were not obeyed, causing
%{ERROR}. Fields:

ERROR enum

type of error resulting.

REGISTER enum

register under consideration.

ArchMsg.Warning.warning_debug_register_access_during_reset

TAGS **UNPREDICTABLE** DISPLAY Unpredictable %{IS_WRITE:(read|write)} access to debug register offset %{OFFSET} during reset. Fields:

IS_WRITE unsigned int

Write Not Read.

OFFSET unsigned int

Register Offset.

ArchMsg.Warning.warning_decode_cps_inconsistent_fields

TAGS **UNPREDICTABLE** DISPLAY The mode setting fields in the CPS instruction are inconsistent.

ArchMsg.Warning.warning_decode_invalid_state

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable in current exception-level/security state.

ArchMsg.Warning.warning_default_cacheable_mmu_on

TAGS **UNPREDICTABLE** DISPLAY Default cacheable is enabled (HCR.DC = 1) while MMU is enabled.

ArchMsg.Warning.warning_default_cacheable_vmmu_off

TAGS **UNPREDICTABLE** DISPLAY Default cacheable is enabled (HCR.DC = 1) while HCR.VM = 0.

ArchMsg.Warning.warning_deprecated_wvr_bit_2_set

TAGS **UNPREDICTABLE** DISPLAY DBGWVR%{N:d}_EL1=%{WVR:h} has bit 2 set, deprecated in ARMv8. Fields:

N unsigned int

index.

WVR unsigned int

DBGWVR.

ArchMsg.Warning.warning_eret_while_software_step_active_pending

Missing ISB between setting MDSCR_EL1.SS and ERET TAGS **UNPREDICTABLE**.

ArchMsg.Warning.warning_exclusive_to_non_normal

TAGS **UNPREDICTABLE** DISPLAY Exclusive Access to Non-Normal Memory %{ADDR}. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_exclusive_to_non_writeback

TAGS **UNPREDICTABLE** DISPLAY Exclusive Access to Non-Writeback-Cacheable Memory %{ADDR}. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_execute_from_device_memory

TAGS **UNPREDICTABLE** DISPLAY An attempt was made to execute from Device memory.

ArchMsg.Warning.warning_illegal_cpsr_mode

TAGS **UNPREDICTABLE** DISPLAY Writing an illegal or unimplement mode ({NEW_MODE:h}) to CPSR was unpredictable. Fields:

NEW_MODE unsigned int

New value of CPSR.M.

ArchMsg.Warning.warning_illegal_srs_mode

TAGS **UNPREDICTABLE** DISPLAY Illegal or **UNPREDICTABLE** mode ({MODE:h}) used for SRS instruction. Fields:

MODE unsigned int

mode for Banked SP.

ArchMsg.Warning.warning_implementation_defined_read_debug_register_in_SCS

Read Access to Debug registers from the processor is **IMPLEMENTATION DEFINED** TAGS **IMP_DEF**. Fields:

PPB_OFFSET unsigned int

debug register offset.

ArchMsg.Warning.warning_implementation_defined_sequential_security_transitions_supported

The behavior of sequential instruction fetches that cross from non-secure to secure memory and contain an SG instruction is **CONSTRAINED UNPREDICTABLE** TAGS **UNPREDICTABLE**.

ArchMsg.Warning.warning_implementation_defined_stack_limit_check_supported

It is **IMPLEMENTATION DEFINED** whether stack pointer limit checking is performed for this instruction TAGS **IMP_DEF**. Fields:

ADDRESS unsigned int

Address of instruction.

OPCODE unsigned int

instruction opcode.

ArchMsg.Warning.warning_implementation_defined_taggednonsharedmemorytype

TAGS **IMP_DEF** DISPLAY Memory attribute has Tagged, non shared memory type; it is **IMPLEMENTATION DEFINED** whether the memory location is treated as Tagged or non-Tagged.

ArchMsg.Warning.warning_implementation_defined_write_debug_register_in_SCS

Write Access to Debug registers from the processor is **IMPLEMENTATION DEFINED** TAGS **IMP_DEF**. Fields:

DATA unsigned int

data attempted to be written.

PPB_OFFSET unsigned int

debug register offset.

ArchMsg.Warning.warning_invalid_tcr_granule

TAGS UNPREDICTABLE DISPLAY TCR.TG{%TG_ID:(0|1)} bits have been set to a granule size {%REQUEST:d}K not implemented - using {%SUBSTITUTE:d}K. Fields:

REQUEST signed int

page size requested (or 0 for reserved).

SUBSTITUTE signed int

best guess available page size.

TG_ID bool

bits TG0 or TG1.

ArchMsg.Warning.warning_load_multiple_user_registers_from_user_mode

TAGS UNPREDICTABLE DISPLAY An LDM(user registers) instruction executed from user mode.

ArchMsg.Warning.warning_mte_in_mem_attr_not_supported

TAGS UNPREDICTABLE DISPLAY Memory attribute {%ATTR} has Tagged memory type but Armv8.5-MTE is not implemented. Fields:

ATTR unsigned int

Memory attribute.

ArchMsg.Warning.warning_reserved_breakpoint_state_match

TAGS UNPREDICTABLE DISPLAY {%IS_BREAKPOINT:(Watchpoint|Breakpoint)} programmed with a reserved combination of HMC, SSC and {%IS_BREAKPOINT:(PAC|PMC)}. Fields:

IS_BREAKPOINT unsigned int

Is a breakpoint, not a watchpoint.

ArchMsg.Warning.warning_shareability

TAGS UNPREDICTABLE DISPLAY Unpredictable: combination of the 1st and 2nd stages of translation is Normal Inner Non-Cacheable, Outer Non-Cacheable.

ArchMsg.Warning.warning_software_step_set_while_enabled

TAGS UNPREDICTABLE DISPLAY MDSCR_EL1.SS set to 1 while software step debug exceptions are enabled.

ArchMsg.Warning.warning_thumb_instruction_wraps_around_memory

TAGS UNPREDICTABLE DISPLAY Thumb instruction wraps around memory.

ArchMsg.Warning.warning_ttbr_sbz_bits_are_not_zero

TAGS UNPREDICTABLE DISPLAY {%TTBR}[{%MSB:d}:0] should be zero, but are not. Fields:

MSB unsigned int

x-1.

TTBR enum

which TTBR register.

ArchMsg.Warning.warning_unaligned_address_dbgdtrrx_write

TAGS **UNPREDICTABLE** DISPLAY Unaligned address %{ADDR} in X0/R0 when DBGDTRRX written in memory access mode. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unaligned_address_dbgdtrtx_read

TAGS **UNPREDICTABLE** DISPLAY Unaligned address %{ADDR} in X0/R0 when DBGDTRTX read in memory access mode. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unaligned_to_device

TAGS **UNPREDICTABLE** DISPLAY Unaligned Access to Device Memory %{ADDR}. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unaligned_to_strongly_ordered

TAGS **UNPREDICTABLE** DISPLAY Unaligned Access to Strongly Ordered Memory %{ADDR}. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unknown_sau_rnr

TAGS **UNPREDICTABLE** DISPLAY SAU_RNR was set to an unsupported value.

ArchMsg.Warning.warning_unpredictable_AIRCR_PRIS_and_BFHFNMINs

TAGS **UNPREDICTABLE** DISPLAY The effect of setting both AIRCR.BFHFNMINs and AIRCR.PRIS to 1 is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_AIRCR_VECTCLRACTIVE_when_not_in_debug

TAGS **UNPREDICTABLE** DISPLAY The effect of writing a 1 to AIRCR.VECTCLRACTIVE if the processor is not halted in Debug state is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_AIRCR_VECTRESET_and_SYSRESETREQ

TAGS **UNPREDICTABLE** DISPLAY When the processor is halted in Debug state, if a write to the register writes a 1 to both VECTRESET and SYSRESETREQ, the behavior is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_AIRCR_VECTRESET_when_not_in_debug

TAGS **UNPREDICTABLE** DISPLAY The effect of writing a 1 to AIRCR.VECTRESET if the processor is not halted in Debug state is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_AIRCR_incorrect_VKEY

TAGS **UNPREDICTABLE** DISPLAY The value 0x05FA must be written to AIRCR.VKEY, otherwise the register write is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_EXC_RETURN_Reserved_Bit

EXC_RETURN[23:7] are reserved with the special condition that all bits should be written as one. Values other than all 1s are **UNPREDICTABLE**. TAGS **UNPREDICTABLE**. Fields:

EXC_RETURN unsigned int

Special PC value.

SBO_MASK unsigned int

Which bits Should Be One to not UNPRED.

ArchMsg.Warning.warning_unpredictable_PMEVTYPEPn_ELO_VS_reserved_value

TAGS **UNPREDICTABLE** DISPLAY Unpredictable reserved value of PMEVTYPEN<n>_ELO.VS.

ArchMsg.Warning.warning_unpredictable_change_to_DEMCR_MON_STEP_at_insufficient_priority

TAGS **UNPREDICTABLE** DISPLAY The effect of changing DEMCR.MON_STEP at an execution priority that is lower than the priority of the DebugMonitor exception is **CONSTRAINED UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_change_to_priority_of_active_exception

TAGS **UNPREDICTABLE** DISPLAY Changing the priority of an active exception is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_clear_lspact

TAGS **UNPREDICTABLE** DISPLAY **UNPREDICTABLE** state on context switch. Fields:

DRVEXC_TAKEN unsigned int

Derived exception taken, not pended.

REACHED_LAST_STORE unsigned int

Only the last faulted.

ArchMsg.Warning.warning_unpredictable_dbg_exit_unaligned_dlr

TAGS **UNPREDICTABLE** DISPLAY Debug exit to AArch32 with DLR[0] set to 1. PC[0] can be set to 0 or to DLR[0].

ArchMsg.Warning.warning_unpredictable_exception_catch

TAGS **UNPREDICTABLE** DISPLAY Generation of an exception catch event is unpredictable.

ArchMsg.Warning.warning_unpredictable_exception_return_inconsistent_state

TAGS **UNPREDICTABLE** DISPLAY state on exception return is unpredictable.

ArchMsg.Warning.warning_unpredictable_exception_return_instruction

TAGS **UNPREDICTABLE** DISPLAY exception return instruction is unpredictable.

ArchMsg.Warning.warning_unpredictable_in_debug_state

TAGS **UNPREDICTABLE** DISPLAY Instruction is **UNPREDICTABLE** when executed in debug state.

ArchMsg.Warning.warning_unpredictable_mair_encoding

TAGS **UNPREDICTABLE** DISPLAY Memory attribute %{ATTR} has invalid MAIR encoding; this may lead to unpredictable results. Fields:

ATTR unsigned int

Memory attribute.

ArchMsg.Warning.warning_unpredictable_pmu_counter_access

TAGS **UNPREDICTABLE** DISPLAY Access to PMU counters from non-secure EL0 or EL1 is **UNPREDICTABLE** with MDCR_EL2.HPMN set to 0.

ArchMsg.Warning.warning_unpredictable_prioritization_breakpoint_match_vector_catch

TAGS **UNPREDICTABLE** DISPLAY Two events with the same priority occurred at the same instruction: Address Matching Vector Catch debug event, and Breakpoint debug event. It is constrained unpredictable which is taken.

ArchMsg.Warning.warning_unpredictable_stack_selection

TAGS **UNPREDICTABLE** DISPLAY **UNPREDICTABLE** state on context switch.

ArchMsg.Warning.warning_unpredictable_tsize_out_of_range

TAGS **UNPREDICTABLE** DISPLAY %{TCR}.T%{TTBR:d}SZ (%{TSIZE:d} bits) is out of range. Must be between 25 and %{TSIZE_MAX:d}. Fields:

TCR enum

Which TCR.

TSIZE signed int

Value of TSize.

TSIZE_MAX signed int

The maximum allowed value.

TTBR unsigned int

Which TTBR region.

ArchMsg.Warning.warning_unpredictable_unaligned_pc_as_base_register

TAGS **UNPREDICTABLE** DISPLAY The PC value must be word-aligned, otherwise the behavior of the instruction is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_unaligned_pop_stack

TAGS **UNPREDICTABLE** DISPLAY **CONSTRAINED UNPREDICTABLE** unaligned stack frame %{FRAME_ADDR:x} in exception-return. Fields:

FRAME_ADDR unsigned int

unaligned stack frame address.

ArchMsg.Warning.warning_unpredictable_unaligned_to_non_writeback

TAGS **UNPREDICTABLE** DISPLAY Unaligned Access to Non-Writeback-Cacheable Memory.

ArchMsg.Warning.warning_unpredictable_vmsa_memattrib

TAGS **UNPREDICTABLE** DISPLAY Unpredictable vmsa memory attribute with texcb value programmed to %{TEXCB} when tex remap is %{TEX_REMAP}. Fields:

TEXCB unsigned int

texcb value.

TEX_REMAP bool

use tex remap.

ArchMsg.Warning.warning_unpredictable_vtcr_t0sz_sl0

TAGS **UNPREDICTABLE** DISPLAY The combination of VTCR.TOSZ=%{TOSZ:d} and VTCR.SLO=%{SLO:d} is unpredictable. Fields:

SLO signed int

VTCR.SLO, starting level for stage 2 translation table walks.

TOSZ signed int

VTCR.TOSZ, required input address range.

ArchMsg.Warning.warning_unpredictable_write_DHCSR

TAGS **UNPREDICTABLE** DISPLAY **UNPREDICTABLE** set of bit-changes in DHCSR. Fields:

NEW unsigned int

data attempted to be written.

OLD unsigned int

previous value.

ArchMsg.Warning.warning_unpredictable_write_SHCSR

TAGS **UNPREDICTABLE** DISPLAY **UNPREDICTABLE** setting a pending bit to 1 for an exception with priority >= execution priority.

ArchMsg.Warning.warning_unsupported_access_to_memory_mapped_register

TAGS **UNPREDICTABLE** DISPLAY Access to memory mapped register is unsupported.

ArchMsg.Warning.warning_user_jmcr_access

TAGS **UNPREDICTABLE** DISPLAY User %{WRITE:(write|read)} access to JMCR is unpredictable. Fields:

WRITE unsigned int

access is write.

ArchMsg.Warning.warning_wcr_mask_and_bas

TAGS **UNPREDICTABLE** DISPLAY DBGWCR_EL1.MASK is non-zero and BAS is not 0xff.

ArchMsg.Warning.warning_wcr_mask_reserved

TAGS **UNPREDICTABLE** DISPLAY DBGWCR_ELn.MASK is set to a reserved value.

ArchMsg.Warning.warning_wcr_non_configuous_bas

TAGS **UNPREDICTABLE** DISPLAY DBGWCR_EL1.BAS has non-contiguous set of ones.

ArchMsg.Warning.warning_wvr_masked_bit_not_zero

TAGS **UNPREDICTABLE** DISPLAY DBGWVRn_EL1.VA contains a bit masked by DBGWCRn_EL1.MASK that is not zero.

ArchMsg.Why.why_branch_target_exception

A branch target exception has occurred. DISPLAY branch target exception: target pc=%{PC}, target opcode=%{OPCODE}. PRIMARY KEY PC. Fields:

BTYPE unsigned int

pstate.BTYPE at time branch taken.

OPCODE unsigned int

opcode at branch target.

PC unsigned int

Virtual Address of branch target.

ArchMsg.Why.why_illegal_state

An illegal mode change has occurred. DISPLAY illegal mode change: %{MESSAGE}. Fields:

MESSAGE string

The message.

ArchMsg.warning_invalid_interval_counter_reload_value

Invalid interval counter reload value in PMSIRR_EL1.INTERVAL: software should set this to a value greater than or equal to the minimum indicated by PMSIDR_EL1.Interval.

ArchMsg.warning_invalid_spe_buffer_write_pointer

Invalid value programmed in PMBPTR_EL1.PTR or PMBLIMITR_EL1.LIMIT: Current write pointer must be at least one sample record below the write limit pointer and PMBPTR_EL1.PTR[63:56] must be equal to PMBLIMITR_EL1.LIMIT[63:56].

BRANCH_MISPREDICT

Simulating branch mispredict. Fields:

PC unsigned int

Origin address (or 0 if unavailable).

BRA_DIR

Direct branches, to immediate address. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISSET enum

The instruction set of the branch instruction.

IS_COND bool

Indicates if this is a conditional branch.

IS_HINTED bool

Whether the conditional instruction is hinted.

IS_ISB bool

Whether the instruction is ISB.

IS_LINK bool

Indicates if this is a branch with link.

PC unsigned int

The address of the branch instruction.

TARGET_ISET enum

The instruction set after the branch.

TARGET_PC unsigned int

The address the instruction branches to.

BRA_INDIR

Indirect branches, perhaps to a register. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISET enum

The instruction set of the branch instruction.

IS_COND bool

Indicates if this is a conditional branch.

IS_HINTED bool

Whether the conditional instruction is hinted.

IS_ISB bool

Whether the instruction is ISB.

IS_LINK bool

Indicates if this is a branch with link.

PC unsigned int

The address of the branch instruction.

TARGET_ISET enum

The instruction set after the branch.

TARGET_PC unsigned int

The address the instruction branches to.

BRA_INDIR_CCFAIL

Indirect branches fail, perhaps to a register.

BRA_RET_CCFAIL

Branch return fail, perhaps to a register.

BRB_FILTRATE

Branch Record Captured event.

CACHE_MAINTENANCE_OP

Cache Maintenance Operation. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

DATA unsigned int

Specified MVA or Set/way.

FUNCTION enum

Clean / Invalidate.

SCOPE enum

Affected region.

SIDE enum

Inst / Data.

CCFAIL

Conditional instruction condition check fail. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CCFAIL_UNC

Unconditional instruction condition check fail. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CCPASS

Conditional instruction condition check pass. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CCPASS_UNC

Unconditional instruction condition check pass. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CODE_CACHE_MAINT

Code cache maintenance. Fields:

END_INCL unsigned int

Inclusive end address for VA or PA requests.

NS bool

Non-secure world for PA.

START unsigned int

Start address for VA or PA requests.

TYPE enum

Request type.

COMPILE_BLOCK_END

Last instruction of basic block translated. Fields:

VADDR unsigned int

Address of next instruction after this basic block.

COMPILE_BLOCK_START

First instruction of basic block translated. Fields:

VADDR unsigned int

Address of first instruction in basic block.

COMPILE_INST

ARM instruction compiled. Fields:

DISASS string

Disassembly of the instruction.

ISSET enum

The instruction set of this instruction.

ITSTATE unsigned int

The ITSTATE current for the instruction.

OPCODE unsigned int

The opcode of the instruction.

PC unsigned int

The address of the instruction.

SIZE unsigned int

The size of the instruction in bytes.

CONTEXTIDR

Context ID Register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

NS bool

Secure or nonsecure banked register is accessed.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The new value written.

CONTEXT_SYNC

Called for every context synchronization event. Fields:

ARCH_CSE bool

Architectural Context Synchronization Event.

PC unsigned int

The address of the synchronization instruction.

SYSREG_SYNC bool

System register synchronization event.

CORE_ENDIAN

Core BE8 Big-Endian state changed. Fields:

BE8 bool

Core BE8 Big-Endian state.

CORE_INFO

Static processor attributes. Only triggered by a call to DumpState(). Fields:

ARCH_PROFILE enum

The architecture profile of the core.

CLUSTER_ID unsigned int

The cluster ID of this processor.

CORE_NUM unsigned int

The number of this core in an MP processor.

DETAILED_CORE_FEATURES enum

Detailed core features as a bitfield. Enum values are bitmasks to extract the info from the field.

FPU_VERSION enum

The VFP version implemented by the core.

MEM_ARCH enum

The memory architecture of the core.

MIDR_EL1 unsigned int

MIDR_EL1 value.

NUM_CORES unsigned int

The number of cores in this MP processor.

QUANTUM_SIZE unsigned int

The default quantum size of the core (in ticks).

SECURITY_FEATURES bool

Does the core have security features?.

CORE_LOADS

Processor load accesses. Fields:

ACCELERATOR_ATOMIC_RESPONSE unsigned int

Accelerator atomic response.

ACCESS_TYPE enum

The type of instruction performing the access.

ACQREL enum

Is this an acquire/release.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DATA unsigned int

The data read or written.

ELEMENT_SIZE unsigned int

Width of each element.

ENDIANNESS enum

Byte ordering of this access.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

LOCK enum

Normal, exclusive or locked access.

MECID unsigned int

Memory Encryption Context ID, if applicable.

MECID2 unsigned int

Memory Encryption Context ID of the second page, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PADDR_TAG unsigned int

If tag checked, the physical address tag for the access.

PAGE_SIZE unsigned int

Page size as $\log_2(\text{size in bytes})$.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK, 2=Exclusive Failed, 3=Unsupported.

SIZE unsigned int

Width of the access in bytes. Only required if DATA is not traced.

TAG_CHECKED bool

Is this access tag checked.

TAG_CHECKED2 bool

Is the second page access tag checked.

TRANS bool

Is this a translated access.

VADDR unsigned int

The virtual address of the access.

CORE_REGS

Changes of the core registers R0 to R14. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number, 0 to 14.

MODE enum

Bank of the register accessed.

OLD_VALUE unsigned int

The old value overwritten.

PHYS_ID enum

The physical register accessed.

VALUE unsigned int

The new value written to the register.

CORE_REGS64

Changes of the core registers X0..X30, SP_ELn. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ID enum

The register number, 0-30 for X0-X30, >=32 for SP_ELn.

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

CORE_REGS64_READ

General purpose AArch64 register has been read X0..X30, SP_ELn. Fields:

ID enum

The register number, 0-30 for X0-X30, >=32 for SP_ELn.

VALUE unsigned int

The value read from the register.

CORE_STALLED_TRANSITION

Detect when a core is prevented from progressing PChannel states. Fields:

CURRENT_STATE enum

Current power mode of core.

REASON string

The reason a transition was stalled, and what must be done to unblock it.

TARGET_STATE enum

Desired power mode of core after transition.

CORE_STORES

Processor store accesses. Fields:

ACCELERATOR_ATOMIC_RESPONSE unsigned int

Accelerator atomic response.

ACCESS_TYPE enum

The type of instruction performing the access.

ACQREL enum

Is this an acquire/release.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DATA unsigned int

The data read or written.

ELEMENT_SIZE unsigned int

Width of each element.

ENDIANNESS enum

Byte ordering of this access.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

LOCK enum

Normal, exclusive or locked access.

MECID unsigned int

Memory Encryption Context ID, if applicable.

MECID2 unsigned int

Memory Encryption Context ID of the second page, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PADDR_TAG unsigned int

If tag checked, the physical address tag for the access.

PAGE_SIZE unsigned int

Page size as $\log_2(\text{size in bytes})$.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK, 2=Exclusive Failed, 3=Unsupported.

SIZE unsigned int

Width of the access in bytes. Only required if DATA is not traced.

TAG_CHECKED bool

Is this access tag checked.

TAG_CHECKED2 bool

Is the second page access tag checked.

TRANS bool

Is this a translated access.

VADDR unsigned int

The virtual address of the access.

CP14_READ

System Coprocessor register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP14 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc1 unsigned int

opcode 1.

opc2 unsigned int

opcode 2.

CP14_WRITE

System Coprocessor register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP14 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc1 unsigned int

opcode 1.

opc2 unsigned int

opcode 2.

CP15_READ

System Coprocessor register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP15 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc1 unsigned int

opcode 1.

opc2 unsigned int

opcode 2.

CP15_READ64

System Coprocessor register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP15 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc unsigned int

opcode 1.

CP15_WRITE

System Control Coprocessor register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP15 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc1 unsigned int

opcode 1.

opc2 unsigned int

opcode 2.

CP15_WRITE64

System Coprocessor register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

NS bool

Secure or nonsecure banked register is accessed.

REG_NAME string

Name of the CP15 register accessed.

SECURITY_STATE enum

The current security state.

UNDEF bool

The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc unsigned int

opcode 1.

CPSR

CPSR change. (In AArch64 PSTATE represented in SPSR format). Fields:

CORE_NUM unsigned int

Core number in a multi processor.

OLD_VALUE unsigned int

The old CPSR value.

UNKNOWN unsigned int

Bits within the register that have unknown value.

VALUE unsigned int

The new CPSR value.

CRYPTO_SPEC

Every crypto instruction speculatively executed.

CorePowerStateChange

Core power-up state has changed. DISPLAY Core powered %{PU:(up|down)}. Fields:

PU bool

Whether the core powered up.

DATA_CACHE_SET_ALLOC_TAGS

Set allocation tags from DC GVA instruction. Fields:

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

NSDESC unsigned int

The security state of the access.

PADDR unsigned int

The physical address of the access.

PAS enum

The physical address space of the page.

RESPONSE enum

0=Aborted, 1=OK.

SIZE unsigned int

The size of memory in bytes.

TAG unsigned int

The value of the tag being set.

TAGGING_TYPE enum

Type of Memory tagging (Physical or Virtual).

TAG_VADDR unsigned int

The virtual address of the Virtual Tag if Virtual tagging is used; otherwise 0.

VADDR unsigned int

Virtual address used for setting allocation tags.

DATA_CACHE_ZERO

Zero and invalidate cache line from DC ZVA instruction. Fields:

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

NSDESC unsigned int

The security state of the access.

PADDR unsigned int

The physical address of the access.

PADDR_TAG unsigned int

If tag checked, the physical address tag for the access.

PAS enum

The physical address space of the page.

RESPONSE enum

0=Aborted, 1=OK.

SIZE unsigned int

The size of zero'd memory in bytes.

TAG_CHECKED bool

Is this access tag checked.

VADDR unsigned int

Virtual address of the zero'd cache line.

DEBUG_EVENT

Hardware debug support event. Fields:

EVENT enum

Description of event.

VALUE unsigned int

data value.

DEBUG_TARGET_ISET

Hardware debug target ISA. Fields:

TARGET_ISET enum

The target instruction set of the debug state.

DMI_ALLOCATE

DMI allocated. Fields:

ACCESS_DIRECTION enum

READ / WRITE.

PADDR unsigned int

The physical address of the DMI.

PTR unsigned int

DMI range pointer.

SIDE enum

DSIDE / ISIDE.

SIZE unsigned int

The size of DMI.

VADDR unsigned int

The virtual address of the DMI..

DMI_HIT

DMI hit, access used DMI (only fire for small DMI regions i.e. less than a full page). Fields:

ACCESS_DIRECTION enum

READ / WRITE.

PADDR unsigned int

The physical address of the DMI.

SIDE enum

DSIDE / ISIDE.

SIZE unsigned int

The size of the DMI.

VADDR unsigned int

The virtual address of the DMI..

DMI_REVOKE

DMI revoked. Fields:

ACCESS_DIRECTION enum

READ / WRITE.

PADDR unsigned int

The physical address of the DMI.

SIDE enum

DSIDE / ISIDE.

SIZE unsigned int

The size of DMI.

VADDR unsigned int

The virtual address of the DMI..

END_COMPILE

Compilation end. Fields:

END_OF_PAGE_COUNT unsigned int

Number of basic blocks exited due to page boundary since START_COMPILE.

FETCHFAIL_COUNT unsigned int

Number of basic blocks exited due to ifetch failure START_COMPILE.

INST_COUNT unsigned int

Number of instructions compiled since START_COMPILE.

NONSEQ_COUNT unsigned int

Number of basic blocks exited due to non-sequential instructions since START_COMPILE.

PAGE_STRADDLE_COUNT unsigned int

Number of basic blocks exited due to unaligned instructions crossing page since START_COMPILE.

ESB

Error Synchronization Barrier hit. Fields:

ASYNC bool

ESB is async.

IMPLICIT bool

ESB is implicit.

EXCEPTION

Exception taken. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ESR unsigned int

The value assigned to ESR when exception is taken.

ESR_EC unsigned int

The Value assigned to ESR.EC when exception is taken.

IS_PHYSICAL bool

Physical or Virtual exception.

IS_TRAP bool

Exception coming from a trap.

LR unsigned int

The value assigned to the link register.

PC unsigned int

The location where the exception occurred.

PREFERRED_RETURN unsigned int

The preferred return address for the exception.

TAKEN_LOCALLY bool

Whether exception is taken locally(same exception level).

TARGET_ISET enum

The instruction set of the exception handler code.

TARGET_PC unsigned int

The address the exception branches to.

VECTOR enum

The exception vector.

VECTOR_OFFSET unsigned int

The offset of an exception vector from the base address.

EXCEPTION_END

Every exception completed.

EXCEPTION_INFO

Trace human-readable reason of the current exception for data/instruction aborts, floating point exceptions and unknown exceptions. Fields:

DESCRIPTION string

Reason for current exception.

ESR_EC unsigned int

The value of ESR.EC when exception is taken.

EXCEPTION_RAISE

Every exception raised.

EXCEPTION_RETURN

Branches on leaving exception. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISSET enum

The instruction set of the branch instruction.

IS_COND bool

Indicates if this is a conditional branch.

IS_HINTED bool

Whether the conditional instruction is hinted.

IS_ISB bool

Whether the instruction is ISB.

IS_LINK bool

Indicates if this is a branch with link.

PC unsigned int

The address of the branch instruction.

TARGET_ISET enum

The instruction set after the branch.

TARGET_PC unsigned int

The address the instruction branches to.

EXCEPTION_RETURN_PREBRANCH

Exception return event that must occur before PC and processor state have updated.

EXCEPTION_START

Every exception started.

FIQ_TAKEN

FIQ taken exception.

FP_EXCEPTIONS_SIGNALLED

Trace which FP exceptions were signalled as a part of the FP operations performed by an instruction. Fields:

FP_EXCEPTIONS_SIGNALLED_MASK unsigned int

Indicates which FP exceptions were signalled by an instruction. This uses the same encoding as the cumulative floating-point exception bitfields in FPSR.

FP_STATE

Floating point state. Fields:

FPSR unsigned int

FPSR register value.

MASK unsigned int

The mask indicating updated cumulative bits.

FREQ_CHANGED

Core frequency changed. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The instruction count of this core.

NEW_FREQ signed int

The new frequency of this core (expressed in Hz).

OLD_FREQ signed int

The old frequency of this core (expressed in Hz).

GPT_LOOKUP

This event is triggered by GPT lookups. Fields:

ABORTED bool

Was the walk successful.

DATA unsigned int

The GPT data read.

DESCADDR unsigned int

The descriptor physical address.

GPI enum

GPI of fetched GPT entry.

LEVEL unsigned int

GPT fetch level.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

PADDR unsigned int

The physical address of the lookup.

PAGEBITS unsigned int

The GPT page bits.

SIDE enum

Inst / Data.

HLT

HLT instruction occurred. Fields:

IMM16 unsigned int

The 16-bit immediate value encoded in HLT instruction.

INST

Every instruction executed. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DEBUG_STATE bool

The instruction is executed in debug state.

DISASS string

Disassembly of instruction.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISSET enum

The current instruction set.

ITSTATE unsigned int

The current ITSTATE.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

MODE enum

The mode the core is in.

NS unsigned int

The core's non-secure bit.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

OPCODE unsigned int

The opcode of the instruction.

PADDR unsigned int

The physical address of the instruction.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the instruction.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the page.

PC unsigned int

The address of the instruction.

SECURITY_STATE enum

The core's security state.

SIZE unsigned int

The size of the instruction in bytes.

INST_END

Every instruction completed.

INST_START

Every instruction started. Fields:

DEBUG_STATE bool

The instruction is executed in debug state.

ISET enum

The current instruction set.

MODE enum

The mode the core is in.

NS enum

The current Secure State.

PC unsigned int

The address of the conditional instruction.

SECURITY_STATE enum

The current Security State.

INST_STRADDLE

Instruction straddles boundary. Fields:

DEBUG_STATE *bool*

The instruction is executed in debug state.

ISET *enum*

The current instruction set.

MODE *enum*

The mode the core is in.

NS *enum*

The current Secure State.

PC *unsigned int*

The address of the conditional instruction.

SECURITY_STATE *enum*

The current Security State.

IRQ_TAKEN

IRQ taken exception.

ISB

ISB instruction occurred. Fields:

PC *unsigned int*

The address of the ISB instruction.

LOAD_MULTIPLE_REGS_LIST

List of destination registers (not source) of a multiple registers load instruction. Fields:

REG *string*

Register to add into the list.

LOCAL_MONITOR

Local monitor activity. Fields:

PADDR *unsigned int*

Local Monitor Address.

State *enum*

State of the monitor (Open/Exclusive).

MAPPING_CREATE

A memory mapping has been created. This is an implementation detail of the model and so may change without notice. Fields:

ASID *unsigned int*

The ASID used during translation, or 0 if no ASID was used.

IS_VALID bool

Whether the mapping is valid or not.

PADDR unsigned int

Base physical address of region, or 0 if not known.

SIDE enum

I-side or D-side.

SIZE unsigned int

Size of the page, or 0 if not known.

VADDR unsigned int

Base virtual address of region.

VMID unsigned int

The VMID used during translation, or 0 if no VMID was used.

MAPPING_DESTROY

A memory mapping has been destroyed. This is an implementation detail of the model and so may change without notice. Fields:

ASID unsigned int

The ASID used during translation.

CAUSE enum

The cause of invalidation.

PADDR unsigned int

Base physical address of page.

SIDE enum

I-side or D-side.

SIZE unsigned int

Size of the page.

VADDR unsigned int

Base virtual address of page.

VMID unsigned int

The VMID used during translation.

MEMMAP_DEBUG_READ

Memory mapped reads to the debug registers. Fields:

ADDR unsigned int

address.

CORE_NUM unsigned int

Core number in a multi processor.

EXT bool

Whether access is from an external device (such as the DAP).

NS enum

Secure state of the access.

REG_NAME string

Name of the debug register accessed.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

MEMMAP_DEBUG_WRITE

Memory mapped writes to the debug registers. Fields:

ADDR unsigned int

address.

CORE_NUM unsigned int

Core number in a multi processor.

EXT bool

Whether access is from an external device (such as the DAP).

NS enum

Secure state of the access.

REG_NAME string

Name of the debug register accessed.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The new value written.

MEMTAG_LOADS

Allocation tag memory read; when attributes and system configuration allow tag accesses.
Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate

(Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

NUM_GRANULES unsigned int

Number of tags to be Loaded/stored.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK.

TAGGING_TYPE enum

Type of Memory tagging (Physical or Virtual).

TAGS unsigned int

The tags read or written. Tags for subsequent granules would be found at subsequent bytes in this field (One byte for each tag).

TAG_VADDR unsigned int

The virtual address of the Virtual Tag if Virtual tagging is used; otherwise 0.

VADDR unsigned int

The virtual address of the access.

MEMTAG_LOAD_INST

Allocation tag load instruction; regardless of tags being actually accessed. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

NUM_GRANULES unsigned int

Number of tags to be Loaded/stored.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK.

TAGGING_TYPE enum

Type of Memory tagging (Physical or Virtual).

TAGS unsigned int

The tags read or written. Tags for subsequent granules would be found at subsequent bytes in this field (One byte for each tag).

TAG_VADDR unsigned int

The virtual address of the Virtual Tag if Virtual tagging is used; otherwise 0.

VADDR unsigned int

The virtual address of the access.

MEMTAG_STORES

Allocation tag memory write; when attributes and system configuration allow tag accesses. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes

([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

NUM_GRANULES unsigned int

Number of tags to be Loaded/stored.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK.

TAGGING_TYPE enum

Type of Memory tagging (Physical or Virtual).

TAGS unsigned int

The tags read or written. Tags for subsequent granules would be found at subsequent bytes in this field (One byte for each tag).

TAG_VADDR unsigned int

The virtual address of the Virtual Tag if Virtual tagging is used; otherwise 0.

VADDR unsigned int

The virtual address of the access.

MEMTAG_STORE_INST

Allocation tag store instruction; regardless of tags being actually accessed. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

NUM_GRANULES unsigned int

Number of tags to be Loaded/stored.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK.

TAGGING_TYPE enum

Type of Memory tagging (Physical or Virtual).

TAGS unsigned int

The tags read or written. Tags for subsequent granules would be found at subsequent bytes in this field (One byte for each tag).

TAG_VADDR unsigned int

The virtual address of the Virtual Tag if Virtual tagging is used; otherwise 0.

VADDR unsigned int

The virtual address of the access.

MMU_D128_TTB_READ

This event is triggered by reads caused by a D128 translation table walk. Fields:

ABORTED bool

Was the walk successful.

DATA_HIGH unsigned int

The data upper bits read.

DATA_LOW unsigned int

The data lower bits read.

IPA unsigned int

For stage 1, the IPA of the read.

LEVEL unsigned int

Translation table level.

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

OUTADDR unsigned int

The output physical address.

PADDR unsigned int

The physical address of the read.

PASmem enum

Physical Address Space of the accessed memory.

PASreq enum

Physical Address Space of the requested lookup.

REGIME_EL enum

Entry matches in this translation regime.

REQUESTER enum

The requester of this table walk.

SIDE enum

Inst / Data.

STAGE unsigned int

Translation stage.

MMU_D128_TTB_WRITE

This event is triggered by writes caused by a D128 translation table walk. Fields:

ABORTED bool

Was the walk successful.

DATA_HIGH unsigned int

The data upper bits written.

DATA_LOW unsigned int

The data lower bits written.

IPA unsigned int

For stage 1, the IPA of the write.

IS_HACDBS_ACCESS bool

If the access was due to HACDBS_CLEAN.

LEVEL unsigned int

Translation table level.

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

OUTADDR unsigned int

The output physical address.

PADDR unsigned int

The physical address of the write.

PASmem enum

Physical Address Space of the accessed memory.

PASreq enum

Physical Address Space of the requested lookup.

REGIME_EL enum

Entry matches in this translation regime.

SIDE enum

Inst / Data.

STAGE unsigned int

Translation stage.

MMU_TRANS

Address translation information. Fields:

ASID unsigned int

Address space identifier.

CORE_NUM unsigned int

Core number in a multi processor.

Hyp bool

Entry matches in Hyp state only.

INNERCACHE_RA bool

Is the inner cache allocate on read.

INNERCACHE_TRANSIENT bool

Is the inner write-through transient.

INNERCACHE_TYPE enum

Inner Caching scheme (NC/MB/WA).

INNERCACHE_WA bool

Is the inner cache allocate on write.

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MEMTYPE enum

Memory type.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC enum

Is secure side supposed to access secure or nonsecure memory.

OUTERCACHE_RA bool

Is the outer cache allocate on read.

OUTERCACHE_TRANSIENT bool

Is the outer write-through transient.

OUTERCACHE_TYPE enum

Outer Caching scheme (NC/MB/WA).

OUTERCACHE_WA bool

Is the outer cache allocate on write.

PADDR unsigned int

Physical address of the access.

PAGESIZE unsigned int

Page size as log2(size).

PAS enum

PAS supposed to be accessed is secure or nonsecure memory or Root/Realm if RME is implemented.

SH enum

Shareability.

SIDE enum

Inst / Data.

STAGE1_PERM unsigned int

Stage 1 permissions, [5:3] Privileged access (XWR) [2:0] User access (XWR).

STAGE2_PERM unsigned int

Stage 2 permission mask [5:3] Privileged access (XWR) [2:0] User access (XWR).

VADDR unsigned int

Virtual address of the access.

VMID unsigned int

Virtual machine identifier.

XS bool

XS Attribute.

nG enum

Flag indicating whether ASID will be matched.

MMU_TTB_READ

This event is triggered by reads caused by a translation table walk. Fields:

ABORTED bool

Was the walk successful.

DATA unsigned int

The data read.

DESC_KIND enum

The encoding scheme of the descriptors used for this walk.

IPA unsigned int

For stage 1, the IPA of the read.

LEVEL unsigned int

Translation table level.

LPAAE bool

Is this for an LPAAE translation (*DEPRECATED*, use DESC_KIND).

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSmem bool

Non secure memory access.

NSreq bool

Non secure request.

OUTADDR unsigned int

The output physical address.

PADDR unsigned int

The physical address of the read.

PASmem enum

Physical Address Space of the accessed memory.

PASreq enum

Physical Address Space of the requested lookup.

REGIME_EL enum

Entry matches in this translation regime.

REQUESTER enum

The requester of this table walk.

SIDE enum

Inst / Data.

STAGE unsigned int

Translation stage.

MMU_TTB_WRITE

This event is triggered by writes caused by a translation table walk. Fields:

ABORTED bool

Was the walk successful.

DATA unsigned int

The data written.

DESC_KIND enum

The encoding scheme of the descriptors used for this walk.

IPA unsigned int

For stage 1, the IPA of the write.

IS_HACDBS_ACCESS bool

If the access was due to HACDBS_CLEAN.

LEVEL unsigned int

Translation table level.

LPAE bool

Is this for an LPAE translation (*DEPRECATED*, use DESC_KIND).

MECID unsigned int

Memory Encryption Context(MEC) ID, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSmem bool

Non secure memory access.

NSreq bool

Non secure request.

OUTADDR unsigned int

The output physical address.

PADDR unsigned int

The physical address of the write.

PASmem enum

Physical Address Space of the accessed memory.

PASreq enum

Physical Address Space of the requested lookup.

REGIME_EL enum

Entry matches in this translation regime.

SIDE enum

Inst / Data.

STAGE unsigned int

Translation stage.

MODE_CHANGE

Mode change. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

MODE enum

The new mode.

NON_SECURE enum

The core's new non-secure bit.

OLD_MODE enum

The old mode.

SECURITY_STATE enum

The new security state.

NOT_TAG_CHECKED_INST

MTE Tag not checked for current instruction.

NV2_CORE_128_LOADS

Processor load accesses for nested virtualization of v9.4 128-bit registers. Fields:

ACCELERATOR_ATOMIC_RESPONSE unsigned int

Accelerator atomic response.

ACCESS_TYPE enum

The type of instruction performing the access.

ACQREL enum

Is this an acquire/release.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DATA unsigned int

The data read or written.

ELEMENT_SIZE unsigned int

Width of each element.

ENDIANNESS enum

Byte ordering of this access.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

LOCK enum

Normal, exclusive or locked access.

MECID unsigned int

Memory Encryption Context ID, if applicable.

MECID2 unsigned int

Memory Encryption Context ID of the second page, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PADDR_TAG unsigned int

If tag checked, the physical address tag for the access.

PAGE_SIZE unsigned int

Page size as $\log_2(\text{size in bytes})$.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

REGNUM_OPERAND unsigned int

Register number corresponding to the instruction operand.

RESPONSE enum

0=Aborted, 1=OK, 2=Exclusive Failed, 3=Unsupported.

SIZE unsigned int

Width of the access in bytes. Only required if DATA is not traced.

TAG_CHECKED bool

Is this access tag checked.

TAG_CHECKED2 bool

Is the second page access tag checked.

TRANS bool

Is this a translated access.

VADDR unsigned int

The virtual address of the access.

NV2_CORE_128_STORES

Processor store accesses for nested virtualization of v9.4 128-bit registers. Fields:

ACCELERATOR_ATOMIC_RESPONSE unsigned int

Accelerator atomic response.

ACCESS_TYPE enum

The type of instruction performing the access.

ACQREL enum

Is this an acquire/release.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DATA unsigned int

The data read or written.

ELEMENT_SIZE unsigned int

Width of each element.

ENDIANNESS enum

Byte ordering of this access.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

LOCK enum

Normal, exclusive or locked access.

MECID unsigned int

Memory Encryption Context ID, if applicable.

MECID2 unsigned int

Memory Encryption Context ID of the second page, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PADDR_TAG unsigned int

If tag checked, the physical address tag for the access.

PAGE_SIZE unsigned int

Page size as $\log_2(\text{size in bytes})$.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

REGNUM_OPERAND unsigned int

Register number corresponding to the instruction operand.

RESPONSE enum

0=Aborted, 1=OK, 2=Exclusive Failed, 3=Unsupported.

SIZE unsigned int

Width of the access in bytes. Only required if DATA is not traced.

TAG_CHECKED bool

Is this access tag checked.

TAG_CHECKED2 bool

Is the second page access tag checked.

TRANS bool

Is this a translated access.

VADDR unsigned int

The virtual address of the access.

NV2_CORE_LOADS

Processor load accesses for v8.4 nested virtualization. Fields:

ACCELERATOR_ATOMIC_RESPONSE unsigned int

Accelerator atomic response.

ACCESS_TYPE enum

The type of instruction performing the access.

ACQREL enum

Is this an acquire/release.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DATA unsigned int

The data read or written.

ELEMENT_SIZE unsigned int

Width of each element.

ENDIANNESS enum

Byte ordering of this access.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

LOCK enum

Normal, exclusive or locked access.

MECID unsigned int

Memory Encryption Context ID, if applicable.

MECID2 unsigned int

Memory Encryption Context ID of the second page, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PADDR_TAG unsigned int

If tag checked, the physical address tag for the access.

PAGE_SIZE unsigned int

Page size as $\log_2(\text{size in bytes})$.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

REGNUM_OPERAND unsigned int

Register number corresponding to the instruction operand.

RESPONSE enum

0=Aborted, 1=OK, 2=Exclusive Failed, 3=Unsupported.

SIZE unsigned int

Width of the access in bytes. Only required if DATA is not traced.

TAG_CHECKED bool

Is this access tag checked.

TAG_CHECKED2 bool

Is the second page access tag checked.

TRANS bool

Is this a translated access.

VADDR unsigned int

The virtual address of the access.

NV2_CORE_STORES

Processor store accesses for v8.4 nested virtualization. Fields:

ACCELERATOR_ATOMIC_RESPONSE unsigned int

Accelerator atomic response.

ACCESS_TYPE enum

The type of instruction performing the access.

ACQREL enum

Is this an acquire/release.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DATA unsigned int

The data read or written.

ELEMENT_SIZE unsigned int

Width of each element.

ENDIANNESS enum

Byte ordering of this access.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

LOCK enum

Normal, exclusive or locked access.

MECID unsigned int

Memory Encryption Context ID, if applicable.

MECID2 unsigned int

Memory Encryption Context ID of the second page, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PADDR_TAG unsigned int

If tag checked, the physical address tag for the access.

PAGE_SIZE unsigned int

Page size as $\log_2(\text{size in bytes})$.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

REGNUM_OPERAND unsigned int

Register number corresponding to the instruction operand.

RESPONSE enum

0=Aborted, 1=OK, 2=Exclusive Failed, 3=Unsupported.

SIZE unsigned int

Width of the access in bytes. Only required if DATA is not traced.

TAG_CHECKED bool

Is this access tag checked.

TAG_CHECKED2 bool

Is the second page access tag checked.

TRANS bool

Is this a translated access.

VADDR unsigned int

The virtual address of the access.

PERIODIC

Called for every quantum. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The instruction count of this CPU.

PC unsigned int

The address of the next instruction to be executed on this CPU.

PMU_COUNTER_OVERFLOW

PMU counter overflow. Fields:

COUNTER_GROUP enum

Counter group.

COUNTER_RANGE enum

Counter range.

EVENT_ID unsigned int

Event Identifier, field is valid only for Event counter group.

INDEX unsigned int

Counter index (as selected by PMSELR), field is valid only for Event/Cycle counter groups.

INTERRUPT bool

Is interrupt enabled on overflow for this counter.

POINTER_AUTH

Called when execution hits a pointer authentication check. Fields:

AUTH_STATE bool

Authentication result.

KEY_USED enum

Pointer authentication key.

POTINDEX_UPDATE

Value of POT index for current pc changed. Fields:

NEW_POTINDEX unsigned int

New POTIndex.

OLD_POTINDEX unsigned int

Old POTIndex.

PREFETCH_MEMORY64

Prefetch from PRFM or PRFUM instructions. Fields:

PRFOP enum

Prefetch hint.

VADDR unsigned int

Virtual address of the location that should be prefetched.

PRELOAD64_REQUEST

Preload requests from PRFM or PRFUM instructions. Fields:

PRFOP enum

Prefetch hint.

VADDR unsigned int

Virtual address of the location that should be prefetched.

PRELOAD_DATA

Data preload from PLD instruction. Fields:

VADDR unsigned int

Virtual address of the data that should be preloaded.

PRELOAD_INST

Instruction preload from PLI instruction. Fields:

VADDR unsigned int

Virtual address of the instruction that should be preloaded.

PRE_CORE_LOAD

Trace just before a core load. Fields:

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

PRE_CORE_STORE

Trace just before a core store. Fields:

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

PRE_TTB_READ

This event is triggered before reads caused by a translation table walk. Fields:

ADDR unsigned int

The physical address of the read.

PSTATE

PSTATE fields update. Fields:

ALLINT unsigned int

All interrupt mask bit.

BTYPE unsigned int

Branch target identification bit.

CORE_NUM unsigned int

Core number in a multi processor.

DAIF unsigned int

Exception mask bits.

DIT unsigned int

Data Independent Timing bit.

EL unsigned int

Current Exception level field.

EXLOCK unsigned int

Exception return state lock bit.

IL unsigned int

Illegal Execution state bit.

NZCV unsigned int

Condition flags.

PACM unsigned int

PACM bit.

PAN unsigned int

Privileged Access Never state bit.

PM unsigned int

PMU exception mask bit.

PPEND unsigned int

PMU exception pending bit.

SM unsigned int

StreamingMode status bit.

SP unsigned int

Stack pointer register selection bit.

SS unsigned int

Software Step bit.

SSBS unsigned int

Speculative Store Bypass Safe bit.

TCO unsigned int

Tag Check Override bit.

TOUCHED unsigned int

Mask of the fields that have been touched since this trace has last been emitted. Note that touched does not necessarily mean the value changed. The index of the field is the bit index in the mask plus the index of this field plus one.

UAO unsigned int

User Access Override bit.

UINJ unsigned int

Undefined Instruction exceptions bit.

ZA unsigned int

ZA storage and ZT0 register enable bit.

nRW unsigned int

Current Execution state bit.

PSTATE_PACTIVE_OPMODE_TRANSITION

Trace pactive and Opmode mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Operating mode after the transition.

OLD_STATE enum

Operating mode before the transition.

PSTATE_PACTIVE_OPMODE_TRANSITION_DENIED

Trace the denied pactive and Opmode mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Operating mode after the transition.

OLD_STATE enum

Operating mode before the transition.

REASON string

Reason for the transition being denied.

PSTATE_PACTIVE_POWER_TRANSITION

Trace pactive and power mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Power mode after the transition.

OLD_STATE enum

Power mode before the transition.

PSTATE_PACTIVE_POWER_TRANSITION_DENIED

Trace the denied pactive and power mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Power mode after the transition.

OLD_STATE enum

Power mode before the transition.

REASON string

Reason for the transition being denied.

PSTATE_PACTIVE_THREAD_OPMODE_TRANSITION

Trace pactive and Opmode mode transitions for threads on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Operating mode after the transition.

OLD_STATE enum

Operating mode before the transition.

QUANTUM_START

Called at the start of every quantum.

RANGE_PRELOAD_REQUEST

Preload requests from RPRFM. Fields:

COUNT unsigned int

Number of blocks of data to access.

LENGTH unsigned int

Number of contiguous bytes to be accessed in each block.

OP unsigned int

Range prefetch operation .

REUSE unsigned int

Maximum number of bytes to be accessed before next RPRFM .

STRIDE signed int

Stride to advance after accessing LENGTH bytes.

VADDR unsigned int

Base virtual address of range prefetch request.

RUN_STATE

Run state transition. Fields:

INST_COUNT unsigned int

Ticks count at point of transition.

NEW enum

New run state.

OLD enum

Old run state.

SBP_SAMPLE_COLLISION

DISPLAY Statistical profiling samples collided .

SBP_SAMPLE_DISCARDED

DISPLAY Operation discarded by statistical profiling unit when configured in discard mode.
Fields:

BUFFER_POINTER unsigned int

Value of buffer pointer.

DATA unsigned int

Data discarded.

SIZE unsigned int

Size of sample, in bytes.

SBP_SAMPLE_END

DISPLAY Statistical profiling of an operation ended.

SBP_SAMPLE_POP

DISPLAY Operation which might be sampled by statistical profiling.

SBP_SAMPLE_RECORDED

DISPLAY Operation sampled by statistical profiling has been written to the buffer at %{BUFFER_ADDRESS} with size %{SIZE} and data %{DATA}. Fields:

BUFFER_ADDRESS unsigned int

Virtual Address of start of buffer write.

DATA unsigned int

Data written to the buffer.

SIZE unsigned int

Size of sample written to buffer, in bytes.

SBP_SAMPLE_START

DISPLAY Statistical profiling of an operation started.

SBP_SAMPLE_TAKEN

DISPLAY Operation sampled by statistical profiling after filtering.

SEMIHOSTING_CALL

Call of a semihost function occurred. Fields:

PC unsigned int

The program counter after the semihosting call.

SEMIHOSTING_PRECALL

About to call semihost. Fields:

EL2 bool

Use EL2 translation regime.

NS bool

Is Non-Secure.

REG_WIDTH unsigned int

The current register width in bytes.

SIGNAL

External signal state change. Fields:

SIGNAL *enum*

Signal that changed.

STATE *bool*

Signal asserted state.

SOFTWARE_STEP

Return the Debug Software Step state. Fields:

STATE *enum*

Software Step state.

SPSR

SPSR change. Fields:

CORE_NUM *unsigned int*

Core number in a multi processor.

MODE *enum*

Which of the banked SPSR registers is written.

OLD_VALUE *unsigned int*

The old SPSR value.

VALUE *unsigned int*

The new SPSR value.

START_COMPILE

Compilation started. Fields:

VADDR *unsigned int*

Instruction where compilation begins.

STG_MULTIPLE_MICRO_OPS_ADDRESS_RANGE

Address range for MTE stores with multiple uops. Fields:

ADDR_END *unsigned int*

Final address.

ADDR_START *unsigned int*

Starting address.

STORE_MULTIPLE_REGS_ADDRESS_RANGE

Address range for stores with multiple registers. Fields:

ADDR_END *unsigned int*

Final address.

ADDR_START *unsigned int*

Starting address.

SUBOPTIMAL_LDST_RETIRED

suboptimal load/store. Fields:

ACCESS_SIZE unsigned int

Log2 of the access width in bytes used for alignment checking (1: half-word, 2: word ...).

OPTIMAL_ALIGN_SELECTOR unsigned int

Selector to decide the optimal alignment size (1: access width, 2: 4-bytes, 3: 8-bytes, ..., 12: 4-kbytes).

VADDR unsigned int

The virtual address of the access.

SVE_INST_SPEC

SVE operations speculatively executed.

SVE_LD_RETIRED

SVE load instructions architecturally executed.

SVE_LOADS

SVE load accesses. Fields:

ACCELERATOR_ATOMIC_RESPONSE unsigned int

Accelerator atomic response.

ACCESS_TYPE enum

The type of instruction performing the access.

ACQREL enum

Is this an acquire/release.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DATA unsigned int

The data read or written.

ELEMENT_SIZE unsigned int

Width of each element.

ENDIANNESS enum

Byte ordering of this access.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

LOCK enum

Normal, exclusive or locked access.

MECID unsigned int

Memory Encryption Context ID, if applicable.

MECID2 unsigned int

Memory Encryption Context ID of the second page, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PADDR_TAG unsigned int

If tag checked, the physical address tag for the access.

PAGE_SIZE unsigned int

Page size as $\log_2(\text{size in bytes})$.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK, 2=Exclusive Failed, 3=Unsupported.

SIZE unsigned int

Width of the access in bytes. Only required if DATA is not traced.

TAG_CHECKED bool

Is this access tag checked.

TAG_CHECKED2 bool

Is the second page access tag checked.

TRANS bool

Is this a translated access.

VADDR unsigned int

The virtual address of the access.

SVE_MEM_PREFETCH

Scalable Vector Extension memory prefetch. Fields:

CACHE_LEVEL enum

The preferred cache level.

HINT enum

What the memory is expected to be used for.

SIZE unsigned int

The prefetch size in bytes.

TEMPORAL bool

Whether the data is temporal.

VADDR unsigned int

The virtual address of the prefetch.

SVE_REG_READ

Scalable Vector Extension register read. Fields:

ID enum

The register identifier.

SM bool

Whether the PE is in Streaming Mode.

VALUE unsigned int

The read data.

SVE_REG_UPDATE

Scalable Vector Extension register update. Fields:

ID enum

The register identifier.

SM bool

Whether the PE is in Streaming Mode.

VALUE unsigned int

The written data.

SVE_REG_WRITE

Scalable Vector Extension register write. Fields:

ID enum

The register identifier.

SM bool

Whether the PE is in Streaming Mode.

VALUE unsigned int

The written data.

SVE_STORES

SVE store accesses. Fields:

ACCELERATOR_ATOMIC_RESPONSE unsigned int

Accelerator atomic response.

ACCESS_TYPE enum

The type of instruction performing the access.

ACQREL enum

Is this an acquire/release.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DATA unsigned int

The data read or written.

ELEMENT_SIZE unsigned int

Width of each element.

ENDIANNESS enum

Byte ordering of this access.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

LOCK enum

Normal, exclusive or locked access.

MECID unsigned int

Memory Encryption Context ID, if applicable.

MECID2 unsigned int

Memory Encryption Context ID of the second page, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PADDR_TAG unsigned int

If tag checked, the physical address tag for the access.

PAGE_SIZE unsigned int

Page size as $\log_2(\text{size in bytes})$.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK, 2=Exclusive Failed, 3=Unsupported.

SIZE unsigned int

Width of the access in bytes. Only required if DATA is not traced.

TAG_CHECKED bool

Is this access tag checked.

TAG_CHECKED2 bool

Is the second page access tag checked.

TRANS bool

Is this a translated access.

VADDR unsigned int

The virtual address of the access.

SVE_ST_RETIRED

SVE store instructions architecturally executed.

SVE_UNALIGNED_LDST_RETIRED

SVE load or store instructions architecturally executed that would cause alignment fault if A bit is set.

SVE_Z_PRED_MEM_ACCESS_INFO

Information about SVE/SME predicated accesses to Z registers (non Gather/Scatter). Fields:

MEM_ELEM_SIZE unsigned int

Width of elements accessed in memory (in bytes).

RnW bool

Whether this is a Read not a Write.

VA unsigned int

Virtual base address computed for accesses.

SYNC

Called for every synchronization, that is every quantum break. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The instruction count of this CPU.

LOCAL_QUANTUM unsigned int

The local quantum of this CPU.

LOCAL_TIME unsigned int

The local time of this CPU.

SYNC_REASON enum

Reason for sync.

SYSCALL

System call instruction executed. Fields:

IMM unsigned int

Immediate value of the system call instruction.

TYPE enum

System call type.

VADDR unsigned int

Instruction that caused the system call.

SYSREG_READ128

System Coprocessor 128 bit register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

REGNUM enum

Internal register number.

REGNUM_PHYS enum

Internal register number physically accessed (e.g. in presense of VHE redirection).

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_READ64

System Coprocessor register read. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REGNUM enum

Internal register number.

REGNUM_OPERAND enum

Register number corresponding to the instruction operand.

REGNUM_PHYS enum

Internal register number physically accessed (e.g. in presense of VHE redirection).

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The value read.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_UPDATE128

Triggers when the system updates a register. Fields:

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG enum

Register number.

UNKNOWN unsigned int

Bits of the register which became unknown.

VALUE unsigned int

Value written to the register.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_UPDATE32

Triggers when the system updates a register. Fields:

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG enum

Register number.

UNKNOWN unsigned int

Bits of the register which became unknown.

VALUE unsigned int

Value written to the register.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_UPDATE64

Triggers when the system updates a register. Fields:

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REG enum

Register number.

UNKNOWN unsigned int

Bits of the register which became unknown.

VALUE unsigned int

Value written to the register.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_WRITE128

System Coprocessor 128 bit register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

REGNUM enum

Internal register number.

REGNUM_PHYS enum

Internal register number physically accessed (e.g. in presense of VHE redirection).

UNDEF bool

The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

SYSREG_WRITE64

System Coprocessor register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CRm unsigned int

CRm.

CRn unsigned int

CRn.

NS bool

Secure or nonsecure banked register is accessed.

REGNUM enum

Internal register number.

REGNUM_OPERAND enum

Register number corresponding to the instruction operand.

REGNUM_PHYS enum

Internal register number physically accessed (e.g. in presense of VHE redirection).

UNDEF bool

The register accessed is undefined.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

The new value written.

opc unsigned int

opcode 1.

opc0 unsigned int

opcode 0.

opc2 unsigned int

opcode 2.

TRBU_BUFFER_WRAP

DISPLAY Trace buffer current write pointer wrapped.

TRBU_TRIGGER_EVENT

DISPLAY Trace buffer Trigger Event.

TRBU_WRITE

DISPLAY PETU trace data has been written to the TRBU memory buffer at `{BUFFER_ADDRESS}` with size `{SIZE}` and data `{DATA}`. Fields:

BUFFER_ADDRESS unsigned int

Virtual Address of start of buffer write.

DATA unsigned int

Data written to the buffer.

SIZE unsigned int

Size of sample written to buffer, in bytes.

UNALIGNED_LDST_RETIRED

Processor unaligned load/store. Fields:

ACCESS_SIZE unsigned int

Log2 of the access width in bytes used for alignment checking (1=>half-word, 2->word ...).

NUMBER_OF_BEATS unsigned int

Number of accesses (beats) in this burst (Total size in bytes is `ACCESS_SIZE * NUMBER_OF_BEATS`).

VADDR unsigned int

The virtual address of the access.

VFP_D_REGS

VFP/NEON D 64 bit register write. Fields:

ALIASING enum

Type of register aliasing used.

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a write to the corresponding bit in the register. For example, 0x1 would indicate a write to VALUE[0].

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

VFP_D_REGS_READ

VFP/NEON D 64 bit register read. Fields:

ID unsigned int

The register number.

VALUE unsigned int

The current value of the register.

VFP_Q_REGS

VFP/NEON Q 128 bit register write. Fields:

ALIASING enum

Type of register aliasing used.

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a write to the corresponding byte in the register. For example, 0x1 would indicate a write to VALUE[7:0].

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

VFP_Q_REGS_READ

VFP/NEON Q 128 bit register read. Fields:

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a read to the corresponding byte in the register. For example, 0x1 would indicate a read to VALUE[7:0].

VALUE unsigned int

The current value of the register.

VFP_SYS_REGS

Writes to the VFP/NEON units system registers. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ID enum

Which VFP system register is written.

MASK unsigned int

The mask indicating updated cumulative bits.

OLD_VALUE unsigned int

The register's old value overwritten.

VALUE unsigned int

The new value written to the VFP system register.

VFP_S_REGS

VFP/NEON S 32 bit register write. Fields:

ALIASING enum

Type of register aliasing used.

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number.

MASK unsigned int

Each bit indicates a write to the corresponding bit in the register. For example, 0x1 would indicate a write to VALUE[0].

OLD_VALUE unsigned int

The old value overwritten.

VALUE unsigned int

The new value written to the register.

VFP_S_REGS_READ

VFP/NEON S 32 bit register read. Fields:

ID unsigned int

The register number.

VALUE unsigned int

The current value of the register.

VIRTUAL_TAG_ALLOCATED

Virtual tag information when allocated. Fields:

TAG_VADDR unsigned int

Virtual tag virtual address.

VADDR unsigned int

Data Virtual address.

VIRTUAL_TAG unsigned int

Virtual tag value.

WAYPOINT

Signals end of basic block execution. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ISET enum

Origin instruction set.

IS_COND bool

Indicates if this is a conditional waypoint.

PC unsigned int

Origin address (or 0 if unavailable).

TAKEN bool

Indicates if this waypoint was taken.

TARGET unsigned int

Destination address.

TARGET_ISET enum

Destination instruction set.

WFE_END

WFE ended. Fields:

INST_COUNT unsigned int

Ticks count when leaving WFE.

WFE_EVENT_REGISTER

WFE event register status: set/clear, reason. Fields:

INST_COUNT unsigned int

Ticks count.

REASON enum

Reason for set/clear. Only REASON==0 (Cleared by WFE) clears the bit, all other reasons set it.

WFE_IGNORED

WFE ignored. Fields:

EVENT bool

This WFE was ignored because the event register was set.

INST_COUNT unsigned int

Ticks count when ignoring WFE.

TRAPPED bool

This WFE was trapped.

WFE_START

WFE entered. Fields:

INST_COUNT unsigned int

Ticks count when entering WFE.

WFI_END

WFI ended. Fields:

INST_COUNT unsigned int

Ticks count when leaving WFI.

WFI_IGNORED

WFI ignored. Fields:

DISABLED bool

This WFI was ignored because WFI is disabled.

INST_COUNT unsigned int

Ticks count when ignoring WFI.

REASON enum

specifies reason why WFI trace was ignored.

TRAPPED bool

This WFI was trapped.

WFI_START

WFI entered. Fields:

INST_COUNT unsigned int

Ticks count when entering WFI.

WFI_WAKEUP

WFI wakeup. Fields:

INST_COUNT unsigned int

Ticks count when WFI wakeup occurred.

REASON enum

Reason for wakeup.

2.71 ARM_SC000

This section describes the trace sources.

ASYNC_MEMORY_FAULT

Context ID Register write. Fields:

FAULT unsigned int

Fault status present in ESR.ISS encoding for Error in ARCH64 and present in DFSR encoding for external abort format in ARCH32.

PADDR unsigned int

Physical Address (or 0 if unavailable).

ATOMIC_END_ACCESS

Bus trace access for atomics. Fields:

ACCESS_FAIL bool

Memory access failed.

ADDR unsigned int

The virtual address of the access.

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

COMPARE_VALUE unsigned int

Compare value for CAS.

LOAD_VALUE unsigned int

Loaded values.

NSDESC unsigned int

The security state of the access.

OPERAND_VALUE unsigned int

Operation's operand.

OPERATION enum

Operation type.

PADDR **unsigned int**

The physical address of the access.

PRIV **bool**

Is this a privileged access?.

ATOMIC_START_ACCESS

Bus trace access for atomics. Fields:

ADDR **unsigned int**

The virtual address of the access.

ATTR **unsigned int**

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

COMPARE_VALUE **unsigned int**

Compare value for CAS.

NSDESC **unsigned int**

The security state of the access.

OPERAND_VALUE **unsigned int**

Operation's operand.

OPERATION **enum**

Operation type.

PADDR **unsigned int**

The physical address of the access.

PRIV **bool**

Is this a privileged access?.

ArchMsg.Error.exit_code

DISPLAY exit code: %{EXITCODE} component kind: %{KIND} reason: %{REASON}. Fields:

EXITCODE **signed int**

Exit code to be returned.

KIND **string**

Component kind that invoked the exit code trace.

REASON **string**

Reason why the exit code trace was invoked (optional).

ArchMsg.Warning.Unpredictably Indexed PM Event Register

TAGS **UNPREDICTABLE** DISPLAY Event %{IsDirect:(Selector value|Register)} %{SEL} is >= %{N}. Fields:

IsDirect unsigned int

Direct Access rather than indirect.

N unsigned int

Number of Accessible Registers.

SEL unsigned int

Selector.

ArchMsg.Warning.dap_csw_bad_size

A write to CM3DAP CSW has an invalid size field. DISPLAY Write %{DATA} to CM3DAP CSW has invalid size field. Fields:

DATA unsigned int

bits[2:0] are size.

ArchMsg.Warning.dcimvac_matches_watchpoint

TAGS **UNPREDICTABLE** DISPLAY Watchpoints matching an AArch32 DCIMVA are implementation defined.

ArchMsg.Warning.decode_fieldmismatch

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with different fields %{FIELD1} and %{FIELD2}. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_frachbitsnegative

TAGS **UNPREDICTABLE** DISPLAY VCVT Instruction is unpredictable with a negative number of fraction bits.

ArchMsg.Warning.decode_initblock

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable in an IT block.

ArchMsg.Warning.decode_initblocknotlast

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable in an IT block when not the last.

ArchMsg.Warning.decode_invalid_condition

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable when the condition is not AL.

ArchMsg.Warning.decode_invalidfieldcombination

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with the values in %{FIELD1} and %{FIELD2}. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_invalidvalue

TAGS UNPREDICTABLE DISPLAY Instruction is unpredictable with the value in %{FIELD}. Fields:

FIELD enum

Field name.

ArchMsg.Warning.decode_mem_hint_unallocated

TAGS UNPREDICTABLE DISPLAY There is no memory hint allocated to this bit pattern.

ArchMsg.Warning.decode_registermatch

TAGS UNPREDICTABLE DISPLAY Instruction is unpredictable with %{FIELD1} and %{FIELD2} as the same register. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_registermismatch

TAGS UNPREDICTABLE DISPLAY Instruction is unpredictable with %{FIELD1} and %{FIELD2} as different registers. Fields:

FIELD1 enum

Field name.

FIELD2 enum

Field name.

ArchMsg.Warning.decode_registeroutofrange

TAGS UNPREDICTABLE DISPLAY Floating-point instruction is unpredictable indexing beyond the end of the register bank.

ArchMsg.Warning.decode_sbzsbo

TAGS UNPREDICTABLE DISPLAY Reserved bits in the instruction are not canonical.

ArchMsg.Warning.decode_transactiontoobig

TAGS UNPREDICTABLE DISPLAY Attempt to perform a memory transaction of more than 128 bytes.

ArchMsg.Warning.decode_unpred_other

TAGS UNPREDICTABLE DISPLAY Instruction is unpredictable.

ArchMsg.Warning.decode_unpreduseofpc

TAGS UNPREDICTABLE DISPLAY Instruction is unpredictable when %{FIELD} is PC. Fields:

FIELD enum

Field name.

ArchMsg.Warning.decode_unpreduseofr13

TAGS UNPREDICTABLE DISPLAY Instruction is unpredictable when %{FIELD} is R13. Fields:

FIELD enum

Field name.

ArchMsg.Warning.decode_unpreduseofr13inreglist

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with R13 in register-list.

ArchMsg.Warning.decode_useofd16orhigher

TAGS **UNPREDICTABLE** DISPLAY Attempt to use register D16 or higher where it is not allowed.

ArchMsg.Warning.decode_writebackandbaseinlist

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable with write back when the base register is in the transfer list.

ArchMsg.Warning.decode_zeroregistersinlist

TAGS **UNPREDICTABLE** DISPLAY Instruction is unpredictable when no registers are to be transferred.

ArchMsg.Warning.recursive_branch

An instruction is performing a branch that targets the same instruction. If this was intended then a WFI might be more effective DISPLAY recursive branch detected at PC=%{PC}. PRIMARY KEY PC. Fields:

PC unsigned int

Address of instruction causing the branch.

ArchMsg.Warning.recursive_exception

An instruction has generated an exception that targets the same instruction. DISPLAY recursive %{TYPE} exception detected at PC=%{PC} (CPSR=%{CPSR}). PRIMARY KEY PC. Fields:

CPSR unsigned int

Processor state.

PC unsigned int

Address of instruction causing the exception.

TYPE enum

Type of exception.

ArchMsg.Warning.reentrant_vector_catch

TAGS **UNPREDICTABLE** DISPLAY Vector catch debug event on Prefetch or Data abort vector %{ADDR}. Fields:

ADDR unsigned int

vector address.

ArchMsg.Warning.reserved_it_state

TAGS **UNPREDICTABLE** DISPLAY Execution with reserved IT state %{ITSTATE}. Fields:

ITSTATE unsigned int

reserved IT state.

ArchMsg.Warning.secure_vector_fetch_from_nonsecure

DISPLAY Secure vector table is being fetched from Non-secure memory.

ArchMsg.Warning.sg_in_it_block

Arm recommends that software does not place SG (security gate instruction) inside an IT instruction block.

ArchMsg.Warning.unpred_FAR_on_LRCPC3_non_atomic_mem_access

TAGS **UNPREDICTABLE** Unpredictable FAR/FSTATUS on LRCPC3 pair instructions resulting in two separate memory accessesDISPLAY Unpredictable FAR on LRCPC3 non atomic memory access at %{ADDR1} with FSTATUS %{FSTATUS1} or %{ADDR2} with FSTATUS %{FSTATUS2}. Fields:

ADDR1 unsigned int

Possible FAR address 1.

ADDR2 unsigned int

Possible FAR address 2.

FSTATUS1 unsigned int

Possible fault status value 1.

FSTATUS2 unsigned int

Possible fault status value 2.

ArchMsg.Warning.unpred_async_tag_check_fail_on_sve_speculative_load_do_update_ffr

Unpredictable set to FALSE of the FFR element associated to SVE speculative load with an async tag check fail TAGS **UNPREDICTABLE**.

ArchMsg.Warning.unpredictable_a32_breakpoint

TAGS **UNPREDICTABLE** DISPLAY Unpredictable breakpoint on a A32 instruction at %{ADDR1} Offset %{OFFSET} BAS %{BAS} for a %{IS_ADDRESS_MISMATCH_BREAKPOINT:(Mismatch|Match)} Breakpoint number %{N:d} Breakpoint address %{ADDR2}. Fields:

ADDR1 unsigned int

Address.

ADDR2 unsigned int

Breakpoint address.

BAS unsigned int

BAS field.

IS_ADDRESS_MISMATCH_BREAKPOINT bool

Is a Mismatch breakpoint, not a Match breakpoint.

N unsigned int

Breakpoint number.

OFFSET unsigned int

Always has a value of zero.

ArchMsg.Warning.unpredictable_banked_register_access

TAGS **UNPREDICTABLE** DISPLAY Access to banked register (R=%{R:d} SYSm=%{SYSM:2x}) is unpredictable. Fields:

R unsigned int

R bit.

SYSM unsigned int

SYSm.

ArchMsg.Warning.unpredictable_ccfail_undef

TAGS **UNPREDICTABLE** DISPLAY An instruction %{OPCODE} at VA=%{VADDR} failed its condition codes check but would otherwise have caused a non-data-dependent trap or undefined exception. Behaviour is implementation-defined choice of exception or nop. Fields:

OPCODE unsigned int

opcode of the instruction.

VADDR unsigned int

Virtual address of the instruction being executed.

ArchMsg.Warning.unpredictable_ici_bits

TAGS **UNPREDICTABLE** DISPLAY ICI bits do not apply or are architecturally unpredictable for instruction.

ArchMsg.Warning.unpredictable_t32_breakpoint

TAGS **UNPREDICTABLE** DISPLAY Unpredictable breakpoint on a T32 instruction at %{ADDR1} Offset %{OFFSET} BAS %{BAS} for a %{IS_ADDRESS_MISMATCH_BREAKPOINT:(Mismatch|Match)} Breakpoint number %{N:d} Breakpoint address %{ADDR2}. Fields:

ADDR1 unsigned int

Address.

ADDR2 unsigned int

Breakpoint address.

BAS unsigned int

BAS field.

IS_ADDRESS_MISMATCH_BREAKPOINT bool

Is a Mismatch breakpoint, not a Match breakpoint.

N unsigned int

Breakpoint number.

OFFSET unsigned int

Indicates which halfword of a 32-bit Thumb instruction the breakpoint is set on. 0 for the first halfword or if the instruction is 16 bits wide, or 1 for the second halfword.

ArchMsg.Warning.unpredictable_t32_vector_catch

TAGS **UNPREDICTABLE** DISPLAY Unpredictable Vector catch exception on a T32 instruction at %{ADDR} Offset %{OFFSET}. Fields:

ADDR unsigned int

Address.

OFFSET unsigned int

Indicates which halfword of a 32-bit Thumb instruction the vector catch exception is set on. 0 for the first halfword or if the instruction is 16 bits wide, or 1 for the second halfword.

ArchMsg.Warning.unpredictable_watchpoint_far

TAGS **UNPREDICTABLE** DISPLAY The value for FAR/EDWAR for hitting this watchpoint may be anywhere in the range %{LOWER_BOUND} to %{UPPER_BOUND}. Fields:

LOWER_BOUND unsigned int

the lowest address accessed by the instruction that triggered the watchpoint.

UPPER_BOUND unsigned int

the highest watchpointed address accessed by that instruction.

ArchMsg.Warning.warning_AdvSIMDExpandImmUnexpectedZero

TAGS **UNPREDICTABLE** DISPLAY AdvSIMDExpandImm may treat this immediate value as **UNPREDICTABLE**.

ArchMsg.Warning.warning_ConditionalSMC

TAGS **UNPREDICTABLE** DISPLAY SMC instruction has **UNPREDICTABLE** effects when conditional and when combined with traps.

ArchMsg.Warning.warning_access_crosses_page_boundary

TAGS **UNPREDICTABLE** DISPLAY Access crosses page boundary.

ArchMsg.Warning.warning_access_crosses_page_boundary_has_fault

TAGS **UNPREDICTABLE** DISPLAY Access that crosses page boundary has fault(s).

ArchMsg.Warning.warning_access_crosses_page_boundary_has_fault_on_both_pages

TAGS **UNPREDICTABLE** DISPLAY Access that crosses page boundary has a fault on each page. Fields:

ADDR1 unsigned int

FAR address on first page.

ADDR2 unsigned int

FAR address on second page.

FSTATUS1 unsigned int

Fault status on first page.

FSTATUS2 unsigned int

Fault status on second page.

ArchMsg.Warning.warning_access_crosses_page_boundary_spanning_different_memory

TAGS **UNPREDICTABLE** DISPLAY Access at address %{ADDR} crosses page from MEM_TYPE %{MEMTYPE_PAGE1} to %{MEMTYPE_PAGE2}. Fields:

ADDR unsigned int

address of access crossing page.

MEMTYPE_PAGE1 signed int

First Page's memory type (0-Normal, 1-Non-Normal).

MEMTYPE_PAGE2 signed int

Second Page's memory type (0-Normal, 1-Non-Normal).

ArchMsg.Warning.warning_access_wraps_around_memory

TAGS **UNPREDICTABLE** DISPLAY Access wraps around memory in %{WIDTH:d}bit mode.
Fields:

WIDTH unsigned int

Address width.

ArchMsg.Warning.warning_bcr_linking_status

TAGS **UNPREDICTABLE** DISPLAY DBGBCR.{BT2,BT} combination gives unpredictable behaviour.

ArchMsg.Warning.warning_bcr_mask_reserved

TAGS **UNPREDICTABLE** DISPLAY DBGBCR_ELn.MASK is set to a reserved value.

ArchMsg.Warning.warning_bvr_masked_bit_not_zero

TAGS **UNPREDICTABLE** DISPLAY DBGBVRn_EL1.VA contains a bit masked by DBGBCRn_EL1.MASK that is not zero.

ArchMsg.Warning.warning_ccsidr_unimplemented_level

TAGS **UNPREDICTABLE** DISPLAY CCSIDR read while CSSELR %{CSSELR:x} points to an unimplemented cache level. Fields:

CSSELR unsigned int

current effective value of CSSELR.

ArchMsg.Warning.warning_change_to_ns_when_tge_set

TAGS **UNPREDICTABLE** DISPLAY Attempting to change NS to 1 when HCR.TGE = 1.

ArchMsg.Warning.warning_contiguous_bit_check_abort

TAGS **UNPREDICTABLE** DISPLAY An abort occurred whilst attempting to check TLB entry is contiguous at %{ENTRY_ADDR}. Fields:

ENTRY_ADDR unsigned int

address of conflicting TLB entry.

ArchMsg.Warning.warning_contiguous_bit_error

TAGS **UNPREDICTABLE** DISPLAY TLB entry at %{ENTRY_ADDR} was not contiguous with entry at %{CONTIG_ADDR} Contents are %{ENTRY_DATA} but expected %{CONTIG_DATA}.
Fields:

CONTIG_ADDR unsigned int

address of the TLB first read, with which this entry is expected to be contiguous.

CONTIG_DATA unsigned int

expected contents based on the entry at CONTIG_ADDR.

ENTRY_ADDR unsigned int

address of conflicting TLB entry.

ENTRY_DATA unsigned int

contents of conflicting TLB entry.

ArchMsg.Warning.warning_contiguous_bit_error_D128

TAGS **UNPREDICTABLE** DISPLAY TLB entry at %{ENTRY_ADDR} was not contiguous with entry at %{CONTIG_ADDR} Contents are %{ENTRY_DATA_HI} %{ENTRY_DATA_LO} but expected %{CONTIG_DATA}. Fields:

CONTIG_ADDR unsigned int

address of the TLB first read, with which this entry is expected to be contiguous.

CONTIG_DATA unsigned int

expected contents based on the entry at CONTIG_ADDR.

ENTRY_ADDR unsigned int

address of conflicting TLB entry.

ENTRY_DATA_HI unsigned int

Higher bit contents of conflicting TLB entry.

ENTRY_DATA_LO unsigned int

Lower bit contents of conflicting TLB entry.

ArchMsg.Warning.warning_cp10_cp11_mismatch

TAGS **UNPREDICTABLE** DISPLAY Unpredictable configuration of CP10 and CP11 controls at EL %{EL:d}. Fields:

EL unsigned int

EL owning the control.

ArchMsg.Warning.warning_cp10_cp11_reserved_value

TAGS **UNPREDICTABLE** DISPLAY Unpredictable configuration of CPACR CP10 (%{CP10}) or CP11 (%{CP11}). Fields:

CP10 unsigned int

CP10 access.

CP11 unsigned int

CP11 access.

ArchMsg.Warning.warning_csselr_level_out_of_range

TAGS **UNPREDICTABLE** DISPLAY CSSELR.Level out of range, written value %{LEVEL} with only %{IMPLEMENTED:d} levels implemented. Fields:

IMPLEMENTED unsigned int

number of cache levels implemented.

LEVEL unsigned int

written value of CSSELR.Level.

TYPE enum

The type of cache selected by CSSELR.InD and TnD.

ArchMsg.Warning.warning_debug_flow_control_bits_not_obeyed

TAGS UNPREDICTABLE DISPLAY Flow control bits for %{REGISTER} were not obeyed, causing %{ERROR}. Fields:

ERROR enum

type of error resulting.

REGISTER enum

register under consideration.

ArchMsg.Warning.warning_debug_register_access_during_reset

TAGS UNPREDICTABLE DISPLAY Unpredictable %{IS_WRITE:(read|write)} access to debug register offset %{OFFSET} during reset. Fields:

IS_WRITE unsigned int

Write Not Read.

OFFSET unsigned int

Register Offset.

ArchMsg.Warning.warning_decode_cps_inconsistent_fields

TAGS UNPREDICTABLE DISPLAY The mode setting fields in the CPS instruction are inconsistent.

ArchMsg.Warning.warning_decode_invalid_state

TAGS UNPREDICTABLE DISPLAY Instruction is unpredictable in current exception-level/security state.

ArchMsg.Warning.warning_default_cacheable_mmu_on

TAGS UNPREDICTABLE DISPLAY Default cacheable is enabled (HCR.DC = 1) while MMU is enabled.

ArchMsg.Warning.warning_default_cacheable_vmmu_off

TAGS UNPREDICTABLE DISPLAY Default cacheable is enabled (HCR.DC = 1) while HCR.VM = 0.

ArchMsg.Warning.warning_deprecated_wvr_bit_2_set

TAGS UNPREDICTABLE DISPLAY DBGWVR%{N:d}_EL1=%{WVR:h} has bit 2 set, deprecated in ARMv8. Fields:

N unsigned int

index.

WVR unsigned int

DBGWVR.

ArchMsg.Warning.warning_eret_while_software_step_active_pending

Missing ISB between setting MDSCR_EL1.SS and ERET TAGS **UNPREDICTABLE**.

ArchMsg.Warning.warning_exclusive_to_non_normal

TAGS **UNPREDICTABLE** DISPLAY Exclusive Access to Non-Normal Memory %{ADDR}. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_exclusive_to_non_writeback

TAGS **UNPREDICTABLE** DISPLAY Exclusive Access to Non-Writeback-Cacheable Memory %{ADDR}. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_execute_from_device_memory

TAGS **UNPREDICTABLE** DISPLAY An attempt was made to execute from Device memory.

ArchMsg.Warning.warning_illegal_cpsr_mode

TAGS **UNPREDICTABLE** DISPLAY Writing an illegal or unimplement mode (%{NEW_MODE:h}) to CPSR was unpredictable. Fields:

NEW_MODE unsigned int

New value of CPSR.M.

ArchMsg.Warning.warning_illegal_srs_mode

TAGS **UNPREDICTABLE** DISPLAY Illegal or **UNPREDICTABLE** mode (%{MODE:h}) used for SRS instruction. Fields:

MODE unsigned int

mode for Banked SP.

ArchMsg.Warning.warning_implementation_defined_read_debug_register_in_SCS

Read Access to Debug registers from the processor is IMPLEMENTATION DEFINED TAGS IMP_DEF. Fields:

PPB_OFFSET unsigned int

debug register offset.

ArchMsg.Warning.warning_implementation_defined_sequential_security_transitions_supported

The behavior of sequential instruction fetches that cross from non-secure to secure memory and contain an SG instruction is CONSTRAINED **UNPREDICTABLE** TAGS **UNPREDICTABLE**.

ArchMsg.Warning.warning_implementation_defined_stack_limit_check_supported

It is IMPLEMENTATION DEFINED whether stack pointer limit checking is performed for this instruction TAGS IMP_DEF. Fields:

ADDRESS unsigned int

Address of instruction.

OPCODE unsigned int

instruction opcode.

ArchMsg.Warning.warning_implementation_defined_taggednonsharedmemorytype

TAGS IMP_DEF DISPLAY Memory attribute has Tagged, non shared memory type; it is **IMPLEMENTATION DEFINED** whether the memory location is treated as Tagged or non-Tagged.

ArchMsg.Warning.warning_implementation_defined_write_debug_register_in_SCS

Write Access to Debug registers from the processor is **IMPLEMENTATION DEFINED** TAGS IMP_DEF. Fields:

DATA unsigned int

data attempted to be written.

PPB_OFFSET unsigned int

debug register offset.

ArchMsg.Warning.warning_invalid_tcr_granule

TAGS **UNPREDICTABLE** DISPLAY TCR.TG%{TG_ID:(0|1)} bits have been set to a granule size %{{REQUEST:d}K not implemented - using %{{SUBSTITUTE:d}K. Fields:

REQUEST signed int

page size requested (or 0 for reserved).

SUBSTITUTE signed int

best guess available page size.

TG_ID bool

bits TG0 or TG1.

ArchMsg.Warning.warning_load_multiple_user_registers_from_user_mode

TAGS **UNPREDICTABLE** DISPLAY An LDM(user registers) instruction executed from user mode.

ArchMsg.Warning.warning_mte_in_mem_attr_not_supported

TAGS **UNPREDICTABLE** DISPLAY Memory attribute %{{ATTR}} has Tagged memory type but Armv8.5-MTE is not implemented. Fields:

ATTR unsigned int

Memory attribute.

ArchMsg.Warning.warning_reserved_breakpoint_state_match

TAGS **UNPREDICTABLE** DISPLAY %{{IS_BREAKPOINT:(Watchpoint|Breakpoint)}} programmed with a reserved combination of HMC, SSC and %{{IS_BREAKPOINT:(PAC|PMC)}}. Fields:

IS_BREAKPOINT unsigned int

Is a breakpoint, not a watchpoint.

ArchMsg.Warning.warning_shareability

TAGS **UNPREDICTABLE** DISPLAY Unpredictable: combination of the 1st and 2nd stages of translation is Normal Inner Non-Cacheable, Outer Non-Cacheable.

ArchMsg.Warning.warning_software_step_set_while_enabled

TAGS UNPREDICTABLE DISPLAY MDSCR_EL1.SS set to 1 while software step debug exceptions are enabled.

ArchMsg.Warning.warning_thumb_instruction_wraps_around_memory

TAGS UNPREDICTABLE DISPLAY Thumb instruction wraps around memory.

ArchMsg.Warning.warning_ttbr_sbz_bits_are_not_zero

TAGS UNPREDICTABLE DISPLAY %{TTBR}[%{MSB:d}:0] should be zero, but are not. Fields:

MSB unsigned int

x-1.

TTBR enum

which TTBR register.

ArchMsg.Warning.warning_unaligned_address_dbgdtrrx_write

TAGS UNPREDICTABLE DISPLAY Unaligned address %{ADDR} in X0/R0 when DBGDTRRX written in memory access mode. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unaligned_address_dbgdtrtx_read

TAGS UNPREDICTABLE DISPLAY Unaligned address %{ADDR} in X0/R0 when DBGDTRTX read in memory access mode. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unaligned_to_device

TAGS UNPREDICTABLE DISPLAY Unaligned Access to Device Memory %{ADDR}. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unaligned_to_strongly_ordered

TAGS UNPREDICTABLE DISPLAY Unaligned Access to Strongly Ordered Memory %{ADDR}. Fields:

ADDR unsigned int

Address.

ArchMsg.Warning.warning_unknown_sau_rnr

TAGS UNPREDICTABLE DISPLAY SAU_RNR was set to an unsupported value.

ArchMsg.Warning.warning_unpredictable_AIRCR_PRIS_and_BFHFNMINs

TAGS UNPREDICTABLE DISPLAY The effect of setting both AIRCR.BFHFNMINs and AIRCR.PRIS to 1 is UNPREDICTABLE.

ArchMsg.Warning.warning_unpredictable_AIRCR_VECTCLRACTIVE_when_not_in_debug

TAGS **UNPREDICTABLE** DISPLAY The effect of writing a 1 to AIRCR.VECTCLRACTIVE if the processor is not halted in Debug state is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_AIRCR_VECTRESET_and_SYSRESETREQ

TAGS **UNPREDICTABLE** DISPLAY When the processor is halted in Debug state, if a write to the register writes a 1 to both VECTRESET and SYSRESETREQ, the behavior is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_AIRCR_VECTRESET_when_not_in_debug

TAGS **UNPREDICTABLE** DISPLAY The effect of writing a 1 to AIRCR.VECTRESET if the processor is not halted in Debug state is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_AIRCR_incorrect_VKEY

TAGS **UNPREDICTABLE** DISPLAY The value 0x05FA must be written to AIRCR.VKEY, otherwise the register write is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_EXC_RETURN_Reserved_Bit

EXC_RETURN[23:7] are reserved with the special condition that all bits should be written as one. Values other than all 1s are **UNPREDICTABLE**. TAGS **UNPREDICTABLE**. Fields:

EXC_RETURN unsigned int

Special PC value.

SBO_MASK unsigned int

Which bits Should Be One to not UNPRED.

ArchMsg.Warning.warning_unpredictable_PMEVTYPEPn_ELO_VS_reserved_value

TAGS **UNPREDICTABLE** DISPLAY Unpredictable reserved value of PMEVTYPEP<n>_ELO.VS.

ArchMsg.Warning.warning_unpredictable_change_to_DEMCR_MON_STEP_at_insufficient_priority

TAGS **UNPREDICTABLE** DISPLAY The effect of changing DEMCR.MON_STEP at an execution priority that is lower than the priority of the DebugMonitor exception is **CONSTRAINED UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_change_to_priority_of_active_exception

TAGS **UNPREDICTABLE** DISPLAY Changing the priority of an active exception is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_clear_lspact

TAGS **UNPREDICTABLE** DISPLAY **UNPREDICTABLE** state on context switch. Fields:

DRVEXC_TAKEN unsigned int

Derived exception taken, not pended.

REACHED_LAST_STORE unsigned int

Only the last faulted.

ArchMsg.Warning.warning_unpredictable_dbg_exit_unaligned_dlr

TAGS **UNPREDICTABLE** DISPLAY Debug exit to AArch32 with DLR[0] set to 1. PC[0] can be set to 0 or to DLR[0].

ArchMsg.Warning.warning_unpredictable_exception_catch

TAGS **UNPREDICTABLE** DISPLAY Generation of an exception catch event is unpredictable.

ArchMsg.Warning.warning_unpredictable_exception_return_inconsistent_state

TAGS **UNPREDICTABLE** DISPLAY state on exception return is unpredictable.

ArchMsg.Warning.warning_unpredictable_exception_return_instruction

TAGS **UNPREDICTABLE** DISPLAY exception return instruction is unpredictable.

ArchMsg.Warning.warning_unpredictable_in_debug_state

TAGS **UNPREDICTABLE** DISPLAY Instruction is **UNPREDICTABLE** when executed in debug state.

ArchMsg.Warning.warning_unpredictable_mair_encoding

TAGS **UNPREDICTABLE** DISPLAY Memory attribute %{ATTR} has invalid MAIR encoding; this may lead to unpredictable results. Fields:

ATTR unsigned int

Memory attribute.

ArchMsg.Warning.warning_unpredictable_pmu_counter_access

TAGS **UNPREDICTABLE** DISPLAY Access to PMU counters from non-secure EL0 or EL1 is **UNPREDICTABLE** with MDCR_EL2.HPMN set to 0.

ArchMsg.Warning.warning_unpredictable_prioritization_breakpoint_match_vector_catch

TAGS **UNPREDICTABLE** DISPLAY Two events with the same priority occurred at the same instruction: Address Matching Vector Catch debug event, and Breakpoint debug event. It is constrained unpredictable which is taken.

ArchMsg.Warning.warning_unpredictable_stack_selection

TAGS **UNPREDICTABLE** DISPLAY **UNPREDICTABLE** state on context switch.

ArchMsg.Warning.warning_unpredictable_tsize_out_of_range

TAGS **UNPREDICTABLE** DISPLAY %{TCR}.T%{TTBR:d}SZ (%{TSIZE:d} bits) is out of range. Must be between 25 and %{TSIZE_MAX:d}. Fields:

TCR enum

Which TCR.

TSIZE signed int

Value of TSize.

TSIZE_MAX signed int

The maximum allowed value.

TTBR unsigned int

Which TTBR region.

ArchMsg.Warning.warning_unpredictable_unaligned_pc_as_base_register

TAGS **UNPREDICTABLE** DISPLAY The PC value must be word-aligned, otherwise the behavior of the instruction is **UNPREDICTABLE**.

ArchMsg.Warning.warning_unpredictable_unaligned_pop_stack

TAGS **UNPREDICTABLE** DISPLAY **CONSTRAINED UNPREDICTABLE** unaligned stack frame %{FRAME_ADDR:x} in exception-return. Fields:

FRAME_ADDR unsigned int
unaligned stack frame address.

ArchMsg.Warning.warning_unpredictable_unaligned_to_non_writeback
TAGS **UNPREDICTABLE** DISPLAY Unaligned Access to Non-Writeback-Cacheable Memory.

ArchMsg.Warning.warning_unpredictable_vmsa_memattrib
TAGS **UNPREDICTABLE** DISPLAY Unpredictable vmsa memory attribute with texcb value programmed to %{TEXCB} when tex remap is %{TEX_REMAP}. Fields:

TEXCB unsigned int
texcb value.

TEX_REMAP bool
use tex remap.

ArchMsg.Warning.warning_unpredictable_vtcr_t0sz_sl0
TAGS **UNPREDICTABLE** DISPLAY The combination of VTCR.TOSZ=%{TOSZ:d} and VTCR.SLO=%{SLO:d} is unpredictable. Fields:

SLO signed int
VTCR.SLO, starting level for stage 2 translation table walks.

TOSZ signed int
VTCR.TOSZ, required input address range.

ArchMsg.Warning.warning_unpredictable_write_DHCSR
TAGS **UNPREDICTABLE** DISPLAY **UNPREDICTABLE** set of bit-changes in DHCSR. Fields:

NEW unsigned int
data attempted to be written.

OLD unsigned int
previous value.

ArchMsg.Warning.warning_unpredictable_write_SHCSR
TAGS **UNPREDICTABLE** DISPLAY **UNPREDICTABLE** setting a pending bit to 1 for an exception with priority >= execution priority.

ArchMsg.Warning.warning_unsupported_access_to_memory_mapped_register
TAGS **UNPREDICTABLE** DISPLAY Access to memory mapped register is unsupported.

ArchMsg.Warning.warning_user_jmcr_access
TAGS **UNPREDICTABLE** DISPLAY User %{WRITE:(write|read)} access to JMCR is unpredictable. Fields:

WRITE unsigned int
access is write.

ArchMsg.Warning.warning_wcr_mask_and_bas
TAGS **UNPREDICTABLE** DISPLAY DBGWCR_EL1.MASK is non-zero and BAS is not 0xff.

ArchMsg.Warning.warning_wcr_mask_reserved

TAGS UNPREDICTABLE DISPLAY DBGWCRn_ELn.MASK is set to a reserved value.

ArchMsg.Warning.warning_wcr_non_configuous_bas

TAGS UNPREDICTABLE DISPLAY DBGWCRn_EL1.BAS has non-contiguous set of ones.

ArchMsg.Warning.warning_wvr_masked_bit_not_zero

TAGS UNPREDICTABLE DISPLAY DBGWVRn_EL1.VA contains a bit masked by DBGWCRn_EL1.MASK that is not zero.

ArchMsg.warning_unpred_system_register_access

DISPLAY Unpredictable %{IS_WRITE:(read|write)} access to system register offset %{OFFSET}. Fields:

IS_WRITE unsigned int

Write Not Read.

OFFSET unsigned int

Register Offset.

BRANCH_MISPREDICT

Simulating branch mispredict. Fields:

PC unsigned int

Origin address (or 0 if unavailable).

BRA_DIR

Direct branches, to immediate address. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISSET enum

The instruction set of the branch instruction.

IS_COND bool

Indicates if this is a conditional branch.

IS_HINTED bool

Whether the conditional instruction is hinted.

IS_ISB bool

Whether the instruction is ISB.

IS_LINK bool

Indicates if this is a branch with link.

PC unsigned int

The address of the branch instruction.

TARGET_ISET enum

The instruction set after the branch.

TARGET_PC unsigned int

The address the instruction branches to.

BRA_INDIR

Indirect branches, perhaps to a register. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISET enum

The instruction set of the branch instruction.

IS_COND bool

Indicates if this is a conditional branch.

IS_HINTED bool

Whether the conditional instruction is hinted.

IS_ISB bool

Whether the instruction is ISB.

IS_LINK bool

Indicates if this is a branch with link.

PC unsigned int

The address of the branch instruction.

TARGET_ISET enum

The instruction set after the branch.

TARGET_PC unsigned int

The address the instruction branches to.

BRA_INDIR_CCFAIL

Indirect branches fail, perhaps to a register.

BRA_RET_CCFAIL

Branch return fail, perhaps to a register.

CCFAIL

Conditional instruction condition check fail. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CCFAIL_UNC

Unconditional instruction condition check fail. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CCPASS

Conditional instruction condition check pass. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CCPASS_UNC

Unconditional instruction condition check pass. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CODE_CACHE_MAINT

Code cache maintenance. Fields:

END_INCL unsigned int

Inclusive end address for VA or PA requests.

NS bool

Non-secure world for PA.

START unsigned int

Start address for VA or PA requests.

TYPE enum

Request type.

COMPILE_BLOCK_END

Last instruction of basic block translated. Fields:

VADDR unsigned int

Address of next instruction after this basic block.

COMPILE_BLOCK_START

First instruction of basic block translated. Fields:

VADDR unsigned int

Address of first instruction in basic block.

COMPILE_INST

ARM instruction compiled. Fields:

DISASS string

Disassembly of the instruction.

ISSET enum

The instruction set of this instruction.

ITSTATE unsigned int

The ITSTATE current for the instruction.

OPCODE unsigned int

The opcode of the instruction.

PC unsigned int

The address of the instruction.

SIZE unsigned int

The size of the instruction in bytes.

CONTEXTIDR

Context ID Register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

NS bool

Secure or nonsecure banked register is accessed.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The new value written.

CONTEXT_SYNC

Called for every context synchronization event. Fields:

ARCH_CSE bool

Architectural Context Synchronization Event.

PC unsigned int

The address of the synchronization instruction.

SYSREG_SYNC bool

System register synchronization event.

CORE_ENDIAN

Core BE8 Big-Endian state changed. Fields:

BE8 bool

Core BE8 Big-Endian state.

CORE_INFO

Static processor attributes. Only triggered by a call to DumpState(). Fields:

ARCH_PROFILE enum

The architecture profile of the core.

CLUSTER_ID unsigned int

The cluster ID of this processor.

CORE_NUM unsigned int

The number of this core in an MP processor.

DETAILED_CORE_FEATURES enum

Detailed core features as a bitfield. Enum values are bitmasks to extract the info from the field.

FPU_VERSION enum

The VFP version implemented by the core.

MEM_ARCH enum

The memory architecture of the core.

MIDR_EL1 unsigned int

MIDR_EL1 value.

NUM_CORES unsigned int

The number of cores in this MP processor.

QUANTUM_SIZE unsigned int

The default quantum size of the core (in ticks).

SECURITY_FEATURES bool

Does the core have security features?.

CORE_LOADS

Processor load accesses. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ACQREL enum

Is this an acquire/release.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DATA unsigned int

The data read or written.

ELEMENT_SIZE unsigned int

Width of each element.

ENDIANNESS enum

Byte ordering of this access.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

LOCK enum

Normal, exclusive or locked access.

MECID unsigned int

Memory Encryption Context ID, if applicable.

MECID2 unsigned int

Memory Encryption Context ID of the second page, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PAGE_SIZE unsigned int

Page size as $\log_2(\text{size in bytes})$.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK, 2=Exclusive Failed, 3=Unsupported.

SIZE unsigned int

Width of the access in bytes. Only required if DATA is not traced.

TRANS bool

Is this a translated access.

VADDR unsigned int

The virtual address of the access.

CORE_REGS

Changes of the core registers R0 to R14. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number, 0 to 14.

OLD_VALUE unsigned int

The old value overwritten.

PHYS_ID enum

The physical register accessed.

VALUE unsigned int

The new value written to the register.

CORE_STALLED_TRANSITION

Detect when a core is prevented from progressing PChannel states. Fields:

CURRENT_STATE enum

Current power mode of core.

REASON string

The reason a transition was stalled, and what must be done to unblock it.

TARGET_STATE enum

Desired power mode of core after transition.

CORE_STORES

Processor store accesses. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ACQREL enum

Is this an acquire/release.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DATA unsigned int

The data read or written.

ELEMENT_SIZE unsigned int

Width of each element.

ENDIANNESS enum

Byte ordering of this access.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

LOCK enum

Normal, exclusive or locked access.

MECID unsigned int

Memory Encryption Context ID, if applicable.

MECID2 unsigned int

Memory Encryption Context ID of the second page, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PAGE_SIZE unsigned int

Page size as $\log_2(\text{size in bytes})$.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK, 2=Exclusive Failed, 3=Unsupported.

SIZE unsigned int

Width of the access in bytes. Only required if DATA is not traced.

TRANS bool

Is this a translated access.

VADDR unsigned int

The virtual address of the access.

CRYPTO_SPEC

Every crypto instruction speculatively executed.

DATA_CACHE_SET_ALLOC_TAGS

Set allocation tags from DC GVA instruction. Fields:

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

NSDESC unsigned int

The security state of the access.

PADDR unsigned int

The physical address of the access.

PAS enum

The physical address space of the page.

RESPONSE enum

0=Aborted, 1=OK.

SIZE unsigned int

The size of memory in bytes.

TAG unsigned int

The value of the tag being set.

TAGGING_TYPE enum

Type of Memory tagging (Physical or Virtual).

TAG_VADDR unsigned int

The virtual address of the Virtual Tag if Virtual tagging is used; otherwise 0.

VADDR unsigned int

Virtual address used for setting allocation tags.

DATA_CACHE_ZERO

Zero and invalidate cache line from DC ZVA instruction. Fields:

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

NSDESC unsigned int

The security state of the access.

PADDR unsigned int

The physical address of the access.

PADDR_TAG unsigned int

If tag checked, the physical address tag for the access.

PAS enum

The physical address space of the page.

RESPONSE enum

0=Aborted, 1=OK.

SIZE unsigned int

The size of zero'd memory in bytes.

TAG_CHECKED bool

Is this access tag checked.

VADDR unsigned int

Virtual address of the zero'd cache line.

DEBUG_EVENT

Hardware debug support event. Fields:

EVENT enum

Description of event.

VALUE unsigned int

data value.

DMI_ALLOCATE

DMI allocated. Fields:

ACCESS_DIRECTION enum

READ / WRITE.

PADDR unsigned int

The physical address of the DMI.

PTR unsigned int

DMI range pointer.

SIDE enum

DSIDE / ISIDE.

SIZE unsigned int

The size of DMI.

VADDR unsigned int

The virtual address of the DMI..

DMI_HIT

DMI hit, access used DMI (only fire for small DMI regions i.e. less than a full page). Fields:

ACCESS_DIRECTION enum

READ / WRITE.

PADDR unsigned int

The physical address of the DMI.

SIDE enum

DSIDE / ISIDE.

SIZE unsigned int

The size of the DMI.

VADDR unsigned int

The virtual address of the DMI..

DMI_REVOKE

DMI revoked. Fields:

ACCESS_DIRECTION enum

READ / WRITE.

PADDR unsigned int

The physical address of the DMI.

SIDE enum

DSIDE / ISIDE.

SIZE unsigned int

The size of DMI.

VADDR unsigned int

The virtual address of the DMI..

DWT_MATCH

DWT comparator matches. Fields:

NUM unsigned int

DWT comparator number.

TYPE enum

DWT comparator configuration.

END_COMPILE

Compilation end. Fields:

END_OF_PAGE_COUNT unsigned int

Number of basic blocks exited due to page boundary since START_COMPILE.

FETCHFAIL_COUNT unsigned int

Number of basic blocks exited due to ifetch failure START_COMPILE.

INST_COUNT unsigned int

Number of instructions compiled since START_COMPILE.

NONSEQ_COUNT unsigned int

Number of basic blocks exited due to non-sequential instructions since START_COMPILE.

PAGE_STRADDLE_COUNT unsigned int

Number of basic blocks exited due to unaligned instructions crossing page since START_COMPILE.

EXCEPTION

Exceptions that are taken. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ISSET enum

The instruction set of the processor when the exception occurred.

LR unsigned int

The value assigned to the link register.

PC unsigned int

The location where the exception occurred.

TARGET_ISET enum

The instruction set of the exception handler code.

TARGET_PC unsigned int

The address the exception branches to.

VECTOR enum

The exception vector.

EXCEPTION_END

Every exception completed.

EXCEPTION_ENTRY

Event marking the entry of an exception. All pushes of registers to the stack will follow this event. Fields:

PC unsigned int

The location where the exception occurred.

VECTOR enum

The exception that occurred.

EXCEPTION_RAISE

Every exception raised.

EXCEPTION_RETURN

Marks the end of an exception, but on an M core does not indicate a program flow change (branch). In most situations it will be followed by an indirect branch. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISET enum

The instruction set of the branch instruction.

PC unsigned int

The address of the branch instruction.

EXCEPTION_RETURN_PREBRANCH

Exception return event that must occur before PC and processor state have updated.

EXCEPTION_START

Every exception started.

EXCEPTION_VECTOR_FETCH

I-side vector fetch for M-class exception. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

DATA unsigned int

Result of fetch if OK.

RESPONSE enum

0=Aborted, 1=OK.

VADDR unsigned int

Where the vector is fetched from (including VTOR).

FP_STATE

Floating point state. Fields:

FPSR unsigned int

FPSR register value.

MASK unsigned int

The mask indicating updated cumulative bits.

FREQ_CHANGED

Core frequency changed. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The instruction count of this core.

NEW_FREQ signed int

The new frequency of this core (expressed in Hz).

OLD_FREQ signed int

The old frequency of this core (expressed in Hz).

HLT

HLT instruction occurred. Fields:

IMM16 unsigned int

The 16-bit immediate value encoded in HLT instruction.

INFO_EXCEPTION_REASON

Provide information of real cause of exception. Fields:

FaultCause enum

Which bit of which V7M/V8M-mainline FSR/HFSR would have been set.

PC unsigned int

The location where the exception occurred.

PHASE enum

What we are doing with the exception.

REASONS enum

Why we are doing it.

VECTOR enum

The exception that occurred.

INFO_STACKING

Provide information of real cause of exception. Fields:

PHASE enum

Are we starting or ending a sequence of stack writes.

REASONS enum

Why we are doing it.

INST

Every instruction executed. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DEBUG_STATE bool

The instruction is executed in debug state.

DISASS string

Disassembly of instruction.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISET enum

The current instruction set.

ITSTATE unsigned int

The current ITSTATE.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

MODE enum

The mode the core is in.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

OPCODE unsigned int

The opcode of the instruction.

PADDR unsigned int

The physical address of the instruction.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the instruction.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the page.

PC unsigned int

The address of the instruction.

SECURITY_STATE enum

The core's security state.

SIZE unsigned int

The size of the instruction in bytes.

INST_END

Every instruction completed.

INST_START

Every instruction started. Fields:

DEBUG_STATE bool

The instruction is executed in debug state.

ISSET enum

The current instruction set.

MODE enum

The mode the core is in.

NS enum

The current Secure State.

PC unsigned int

The address of the conditional instruction.

SECURITY_STATE enum

The current Security State.

INST_STRADDLE

Instruction straddles boundary. Fields:

DEBUG_STATE bool

The instruction is executed in debug state.

ISSET enum

The current instruction set.

MODE enum

The mode the core is in.

NS enum

The current Secure State.

PC unsigned int

The address of the conditional instruction.

SECURITY_STATE enum

The current Security State.

ISB

ISB instruction occurred. Fields:

PC unsigned int

The address of the ISB instruction.

LOAD_MULTIPLE_REGS_LIST

List of destination registers (not source) of a multiple registers load instruction. Fields:

REG string

Register to add into the list.

LOCAL_MONITOR

Local monitor activity. Fields:

PADDR unsigned int

Local Monitor Address.

State enum

State of the monitor (Open/Exclusive).

LOCKUP_CYCLE

This event is triggered if the core enters or stays in the lock-up state. Fields:

CAUSE enum

What causes the lockup?.

LOCKUP_ENTRY

This event is triggered if the core enters the lock-up state. Fields:

PC unsigned int

The PC when the lockup state is entered.

LOCKUP_EXIT

This event is triggered if the core leaves the lock-up state.

MODE_CHANGE

Mode change. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

MODE enum

The new mode.

OLD_MODE enum

The old mode.

MPU_TRANS

Address translation information. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

EXEC_PERM enum

Execution Permission.

INNERCACHE_RA bool

Is the inner cache allocate on read.

INNERCACHE_TYPE enum

Inner Caching scheme (NC/MB/WA).

INNERCACHE_WA bool

Is the inner cache allocate on write.

MEMTYPE enum

Memory type.

OUTERCACHE_RA bool

Is the outer cache allocate on read.

OUTERCACHE_TYPE enum

Outer Caching scheme (NC/MB/WA).

OUTERCACHE_WA bool

Is the outer cache allocate on write.

PADDR unsigned int

Address of the access.

PAGESIZE unsigned int

Page size as $\log_2(\text{size})$.

READ_PERM enum

Read Permission.

REG_NUM unsigned int

MPU register based region-number 0-15 (else 0xFF=none, 0xBB/0xDD=background/default).

SH enum

Shareability.

SIDE enum

Inst / Data.

WRITE_PERM enum

Write Permission.

PERIODIC

Called for every quantum. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The instruction count of this CPU.

PC unsigned int

The address of the next instruction to be executed on this CPU.

POTINDEX_UPDATE

Value of POT index for current pc changed. Fields:

NEW_POTINDEX unsigned int

New POTIndex.

OLD_POTINDEX unsigned int

Old POTIndex.

PPB_READ_ACCESS

Trace reads from the core to memory mapped registers in the PPB (private peripheral bus) address range. Fields:

ACCESS_SIZE unsigned int

Logarithm of the access size: 0=byte, 1=halfword, 2=word, 3=doubleword, ...

BURST_LEN unsigned int

The number of data transfers (beats) in this burst.

DATA unsigned int

The data transferred.

INNER_ATTRIBUTES unsigned int

The inner attributes: bit0=write allocate, bit1=read allocate, bit2=cachable, bit3=bufferable, bit4=shareable.

INST bool

Is an instruction fetch.

LOCK enum

Normal / Locked / Exclusive.

MANAGER_ID unsigned int

The AXI manager ID.

NS bool

Is a nonsecure access.

OUTER_ATTRIBUTES unsigned int

The outer attributes: bit0=write allocate, bit1=read allocate, bit2=cachable, bit3=bufferable, bit4=shareable.

PADDR unsigned int

Physical address of access.

PRIV bool

Is privileged access.

RESPONSE enum

Whether the transaction was successful, or an error occurred.

USER_FLAGS unsigned int

Core specific additional signals.

PPB_WRITE_ACCESS

Trace writes from the core to memory mapped registers in the PPB (private peripheral bus) address range. Fields:

ACCESS_SIZE unsigned int

Logarithm of the access size: 0=byte, 1=halfword, 2=word, 3=doubleword, ...

BURST_LEN unsigned int

The number of data transfers (beats) in this burst.

DATA unsigned int

The data transferred.

INNER_ATTRIBUTES unsigned int

The inner attributes: bit0=write allocate, bit1=read allocate, bit2=cachable, bit3=bufferable, bit4=shareable.

INST bool

Is an instruction fetch.

LOCK enum

Normal / Locked / Exclusive.

MANAGER_ID unsigned int

The AXI manager ID.

NS bool

Is a nonsecure access.

OUTER_ATTRIBUTES unsigned int

The outer attributes: bit0=write allocate, bit1=read allocate, bit2=cachable, bit3=bufferable, bit4=shareable.

PADDR unsigned int

Physical address of access.

PRIV bool

Is privileged access.

RESPONSE enum

Whether the transaction was successful, or an error occurred.

USER_FLAGS unsigned int

Core specific additional signals.

PRELOAD_DATA

Data preload from PLD instruction. Fields:

VADDR unsigned int

Virtual address of the data that should be preloaded.

PRELOAD_INST

Instruction preload from PLI instruction. Fields:

VADDR unsigned int

Virtual address of the instruction that should be preloaded.

PRE_CORE_LOAD

Trace just before a core load. Fields:

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

PRE_CORE_STORE

Trace just before a core store. Fields:

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

PSTATE_PACTIVE_OPMODE_TRANSITION

Trace pactive and Opmode mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Operating mode after the transition.

OLD_STATE enum

Operating mode before the transition.

PSTATE_PACTIVE_OPMODE_TRANSITION_DENIED

Trace the denied pactive and Opmode mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Operating mode after the transition.

OLD_STATE enum

Operating mode before the transition.

REASON string

Reason for the transition being denied.

PSTATE_PACTIVE_POWER_TRANSITION

Trace pactive and power mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Power mode after the transition.

OLD_STATE enum

Power mode before the transition.

PSTATE_PACTIVE_POWER_TRANSITION_DENIED

Trace the denied pactive and power mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Power mode after the transition.

OLD_STATE enum

Power mode before the transition.

REASON string

Reason for the transition being denied.

PSTATE_PACTIVE_THREAD_OPMODE_TRANSITION

Trace pactive and Opmode mode transitions for threads on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Operating mode after the transition.

OLD_STATE enum

Operating mode before the transition.

QUANTUM_START

Called at the start of every quantum.

RUN_STATE

Run state transition. Fields:

INST_COUNT unsigned int

Ticks count at point of transition.

NEW enum

New run state.

OLD enum

Old run state.

SEMIHOSTING_CALL

Call of a semihost function occurred. Fields:

PC unsigned int

The program counter after the semihosting call.

SEMIHOSTING_PRECALL

About to call semihost. Fields:

EL2 bool

Use EL2 translation regime.

NS bool

Is Non-Secure.

REG_WIDTH unsigned int

The current register width in bytes.

SIGNAL

External signal state change. Fields:

SIGNAL enum

Signal that changed.

STATE bool

Signal asserted state.

START_COMPILE

Compilation started. Fields:

VADDR unsigned int

Instruction where compilation begins.

STG_MULTIPLE_MICRO_OPS_ADDRESS_RANGE

Address range for MTE stores with multiple uops. Fields:

ADDR_END unsigned int

Final address.

ADDR_START unsigned int

Starting address.

STORE_MULTIPLE_REGS_ADDRESS_RANGE

Address range for stores with multiple registers. Fields:

ADDR_END unsigned int

Final address.

ADDR_START unsigned int

Starting address.

SUBOPTIMAL_LDST_RETIRED

suboptimal load/store. Fields:

ACCESS_SIZE unsigned int

Log2 of the access width in bytes used for alignment checking (1: half-word, 2: word ...).

OPTIMAL_ALIGN_SELECTOR unsigned int

Selector to decide the optimal alignment size (1: access width, 2: 4-bytes, 3: 8-bytes, ..., 12: 4-kbytes).

VADDR unsigned int

The virtual address of the access.

SYNC

Called for every synchronization, that is every quantum break. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The instruction count of this CPU.

LOCAL_QUANTUM unsigned int

The local quantum of this CPU.

LOCAL_TIME unsigned int

The local time of this CPU.

SYNC_REASON enum

Reason for sync.

SYSCALL

System call instruction executed. Fields:

IMM unsigned int

Immediate value of the system call instruction.

TYPE enum

System call type.

VADDR unsigned int

Instruction that caused the system call.

SYSREG_READ

Trace reads from system registers. Fields:

REG_NAME enum

Register number.

VALUE unsigned int

The value read.

SYSREG_UPDATE128

Triggers when the system updates a register. Fields:

REG enum

Register number.

UNKNOWN unsigned int

Bits of the register which became unknown.

VALUE unsigned int

Value written to the register.

SYSREG_UPDATE32

Triggers when the system updates a register. Fields:

REG enum

Register number.

UNKNOWN unsigned int

Bits of the register which became unknown.

VALUE unsigned int

Value written to the register.

SYSREG_UPDATE64

Triggers when the system updates a register. Fields:

REG enum

Register number.

UNKNOWN unsigned int

Bits of the register which became unknown.

VALUE unsigned int

Value written to the register.

SYSREG_WRITE

Trace write to system registers. Fields:

REG_ACCESS enum

Register access status.

REG_NAME enum

Register number.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

Value written to the register.

UNALIGNED_LDST_RETIRED

Processor unaligned load/store. Fields:

ACCESS_SIZE unsigned int

Log2 of the access width in bytes used for alignment checking (1=>half-word, 2->word ...).

NUMBER_OF_BEATS unsigned int

Number of accesses (beats) in this burst (Total size in bytes is ACCESS_SIZE * NUMBER_OF_BEATS).

VADDR unsigned int

The virtual address of the access.

WAYPOINT

Signals end of basic block execution. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ISET enum

Origin instruction set.

IS_COND bool

Indicates if this is a conditional waypoint.

PC unsigned int

Origin address (or 0 if unavailable).

TAKEN bool

Indicates if this waypoint was taken.

TARGET unsigned int

Destination address.

TARGET_ISET enum

Destination instruction set.

WFE_END

WFE ended. Fields:

INST_COUNT unsigned int

Ticks count when leaving WFE.

WFE_EVENT_REGISTER

WFE event register status: set/clear, reason. Fields:

INST_COUNT unsigned int

Ticks count.

REASON enum

Reason for set/clear. Only REASON==0 (Cleared by WFE) clears the bit, all other reasons set it.

WFE_IGNORED

WFE ignored. Fields:

EVENT bool

This WFE was ignored because the event register was set.

INST_COUNT unsigned int

Ticks count when ignoring WFE.

TRAPPED bool

This WFE was trapped.

WFE_START

WFE entered. Fields:

INST_COUNT unsigned int

Ticks count when entering WFE.

WFI_END

WFI ended. Fields:

INST_COUNT unsigned int

Ticks count when leaving WFI.

WFI_IGNORED

WFI ignored. Fields:

DISABLED bool

This WFI was ignored because WFI is disabled.

INST_COUNT unsigned int

Ticks count when ignoring WFI.

REASON enum

specifies reason why WFI trace was ignored.

TRAPPED bool

This WFI was trapped.

WFI_START

WFI entered. Fields:

INST_COUNT unsigned int

Ticks count when entering WFI.

WFI_WAKEUP

WFI wakeup. Fields:

INST_COUNT unsigned int

Ticks count when WFI wakeup occurred.

REASON enum

Reason for wakeup.

XPSR

Changes to the xPSR register. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

OLD_VALUE unsigned int

The old xPSR value.

UNKNOWN unsigned int

Bits within the register that have unknown value.

VALUE unsigned int

The new xPSR value.

2.72 ARM_SC300

This section describes the trace sources.

ASYNC_MEMORY_FAULT

Context ID Register write. Fields:

FAULT unsigned int

Fault status present in ESR.ISS encoding for Serror in ARCH64 and present in DFSR encoding for external abort format in ARCH32.

PADDR unsigned int

Physical Address (or 0 if unavailable).

ATOMIC_END_ACCESS

Bus trace access for atomics. Fields:

ACCESS_FAIL bool

Memory access failed.

ADDR unsigned int

The virtual address of the access.

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate

when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

COMPARE_VALUE unsigned int

Compare value for CAS.

LOAD_VALUE unsigned int

Loaded values.

NSDESC unsigned int

The security state of the access.

OPERAND_VALUE unsigned int

Operation's operand.

OPERATION enum

Operation type.

PADDR unsigned int

The physical address of the access.

PRIV bool

Is this a privileged access?.

ATOMIC_START_ACCESS

Bus trace access for atomics. Fields:

ADDR unsigned int

The virtual address of the access.

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

COMPARE_VALUE unsigned int

Compare value for CAS.

NSDESC unsigned int

The security state of the access.

OPERAND_VALUE unsigned int

Operation's operand.

OPERATION enum

Operation type.

PADDR unsigned int

The physical address of the access.

PRIV bool

Is this a privileged access?.

ArchMsg.Error.exit_code

DISPLAY exit code: %{EXITCODE} component kind: %{KIND} reason: %{REASON}. Fields:

EXITCODE signed int

Exit code to be returned.

KIND string

Component kind that invoked the exit code trace.

REASON string

Reason why the exit code trace was invoked (optional).

ArchMsg.Warning.dap_csw_bad_size

A write to CM3DAP CSW has an invalid size field. DISPLAY Write %{DATA} to CM3DAP CSW has invalid size field. Fields:

DATA unsigned int

bits[2:0] are size.

ArchMsg.Warning.recursive_branch

An instruction is performing a branch that targets the same instruction. If this was intended then a WFI might be more effective DISPLAY recursive branch detected at PC=%{PC}. PRIMARY KEY PC. Fields:

PC unsigned int

Address of instruction causing the branch.

ArchMsg.Warning.recursive_exception

An instruction has generated an exception that targets the same instruction. DISPLAY recursive %{TYPE} exception detected at PC=%{PC} (CPSR=%{CPSR}). PRIMARY KEY PC. Fields:

CPSR unsigned int

Processor state.

PC unsigned int

Address of instruction causing the exception.

TYPE enum

Type of exception.

ArchMsg.Warning.secure_vector_fetch_from_nonsecure

DISPLAY Secure vector table is being fetched from Non-secure memory.

ArchMsg.Warning.sg_in_it_block

Arm recommends that software does not place SG (security gate instruction) inside an IT instruction block.

ArchMsg.warning_unpred_system_register_access

DISPLAY Unpredictable %{IS_WRITE:(read|write)} access to system register offset %{OFFSET}. Fields:

IS_WRITE unsigned int

Write Not Read.

OFFSET unsigned int

Register Offset.

BRANCH_MISPREDICT

Simulating branch mispredict. Fields:

PC unsigned int

Origin address (or 0 if unavailable).

BRA_DIR

Direct branches, to immediate address. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISSET enum

The instruction set of the branch instruction.

IS_COND bool

Indicates if this is a conditional branch.

IS_HINTED bool

Whether the conditional instruction is hinted.

IS_ISB bool

Whether the instruction is ISB.

IS_LINK bool

Indicates if this is a branch with link.

PC unsigned int

The address of the branch instruction.

TARGET_ISET enum

The instruction set after the branch.

TARGET_PC unsigned int

The address the instruction branches to.

BRA_INDIR

Indirect branches, perhaps to a register. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISSET enum

The instruction set of the branch instruction.

IS_COND bool

Indicates if this is a conditional branch.

IS_HINTED bool

Whether the conditional instruction is hinted.

IS_ISB bool

Whether the instruction is ISB.

IS_LINK bool

Indicates if this is a branch with link.

PC unsigned int

The address of the branch instruction.

TARGET_ISET enum

The instruction set after the branch.

TARGET_PC unsigned int

The address the instruction branches to.

BRA_INDIR_CCFAIL

Indirect branches fail, perhaps to a register.

BRA_RET_CCFAIL

Branch return fail, perhaps to a register.

CCFAIL

Conditional instruction condition check fail. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CCFAIL_UNC

Unconditional instruction condition check fail. Fields:

COND enum

The condition of the conditional instruction.

CORE_NUM unsigned int

Core number in a multi processor.

PC unsigned int

The address of the conditional instruction.

CCPASS

Conditional instruction condition check pass. Fields:

COND *enum*

The condition of the conditional instruction.

CORE_NUM *unsigned int*

Core number in a multi processor.

PC *unsigned int*

The address of the conditional instruction.

CCPASS_UNC

Unconditional instruction condition check pass. Fields:

COND *enum*

The condition of the conditional instruction.

CORE_NUM *unsigned int*

Core number in a multi processor.

PC *unsigned int*

The address of the conditional instruction.

CODE_CACHE_MAINT

Code cache maintenance. Fields:

END_INCL *unsigned int*

Inclusive end address for VA or PA requests.

NS *bool*

Non-secure world for PA.

START *unsigned int*

Start address for VA or PA requests.

TYPE *enum*

Request type.

COMPILE_BLOCK_END

Last instruction of basic block translated. Fields:

VADDR *unsigned int*

Address of next instruction after this basic block.

COMPILE_BLOCK_START

First instruction of basic block translated. Fields:

VADDR *unsigned int*

Address of first instruction in basic block.

COMPILE_INST

ARM instruction compiled. Fields:

DISASS string

Disassembly of the instruction.

ISET enum

The instruction set of this instruction.

ITSTATE unsigned int

The ITSTATE current for the instruction.

OPCODE unsigned int

The opcode of the instruction.

PC unsigned int

The address of the instruction.

SIZE unsigned int

The size of the instruction in bytes.

CONTEXTIDR

Context ID Register write. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

NS bool

Secure or nonsecure banked register is accessed.

UNDEF bool

The register accessed is undefined.

VALUE unsigned int

The new value written.

CONTEXT_SYNC

Called for every context synchronization event. Fields:

ARCH_CSE bool

Architectural Context Synchronization Event.

PC unsigned int

The address of the synchronization instruction.

SYSREG_SYNC bool

System register synchronization event.

CORE_ENDIAN

Core BE8 Big-Endian state changed. Fields:

BE8 bool

Core BE8 Big-Endian state.

CORE_INFO

Static processor attributes. Only triggered by a call to DumpState(). Fields:

ARCH_PROFILE enum

The architecture profile of the core.

CLUSTER_ID unsigned int

The cluster ID of this processor.

CORE_NUM unsigned int

The number of this core in an MP processor.

DETAILED_CORE_FEATURES enum

Detailed core features as a bitfield. Enum values are bitmasks to extract the info from the field.

FPU_VERSION enum

The VFP version implemented by the core.

MEM_ARCH enum

The memory architecture of the core.

MIDR_EL1 unsigned int

MIDR_EL1 value.

NUM_CORES unsigned int

The number of cores in this MP processor.

QUANTUM_SIZE unsigned int

The default quantum size of the core (in ticks).

SECURITY_FEATURES bool

Does the core have security features?.

CORE_LOADS

Processor load accesses. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ACQREL enum

Is this an acquire/release.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DATA unsigned int

The data read or written.

ELEMENT_SIZE unsigned int

Width of each element.

ENDIANNESS enum

Byte ordering of this access.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

LOCK enum

Normal, exclusive or locked access.

MECID unsigned int

Memory Encryption Context ID, if applicable.

MECID2 unsigned int

Memory Encryption Context ID of the second page, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PAGE_SIZE unsigned int

Page size as $\log_2(\text{size in bytes})$.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK, 2=Exclusive Failed, 3=Unsupported.

SIZE unsigned int

Width of the access in bytes. Only required if DATA is not traced.

TRANS bool

Is this a translated access.

VADDR unsigned int

The virtual address of the access.

CORE_REGS

Changes of the core registers R0 to R14. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ID unsigned int

The register number, 0 to 14.

OLD_VALUE unsigned int

The old value overwritten.

PHYS_ID enum

The physical register accessed.

VALUE unsigned int

The new value written to the register.

CORE_STALLED_TRANSITION

Detect when a core is prevented from progressing PChannel states. Fields:

CURRENT_STATE enum

Current power mode of core.

REASON string

The reason a transition was stalled, and what must be done to unblock it.

TARGET_STATE enum

Desired power mode of core after transition.

CORE_STORES

Processor store accesses. Fields:

ACCESS_TYPE enum

The type of instruction performing the access.

ACQREL enum

Is this an acquire/release.

ATTR unsigned int

Memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

ATTR2 unsigned int

Second page memory attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

DATA unsigned int

The data read or written.

ELEMENT_SIZE unsigned int

Width of each element.

ENDIANNESS enum

Byte ordering of this access.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

LOCK enum

Normal, exclusive or locked access.

MECID unsigned int

Memory Encryption Context ID, if applicable.

MECID2 unsigned int

Memory Encryption Context ID of the second page, if applicable.

MPAM_PARTID unsigned int

MPAM Partition ID.

MPAM_PMG unsigned int

MPAM Performance Monitoring Group.

MPAM_SP unsigned int

MPAM Partition ID Space.

NSDESC unsigned int

The physical address non-secure bit.

NSDESC2 unsigned int

The second page physical address non-secure bit.

PADDR unsigned int

The physical (translated) address.

PADDR2 unsigned int

If different from PADDR, the physical address of the second page of the access.

PAGE_SIZE unsigned int

Page size as $\log_2(\text{size in bytes})$.

PAS enum

The physical address space of the page.

PAS2 enum

The physical address space of the second page.

RESPONSE enum

0=Aborted, 1=OK, 2=Exclusive Failed, 3=Unsupported.

SIZE unsigned int

Width of the access in bytes. Only required if DATA is not traced.

TRANS bool

Is this a translated access.

VADDR unsigned int

The virtual address of the access.

CRYPTO_SPEC

Every crypto instruction speculatively executed.

DATA_CACHE_SET_ALLOC_TAGS

Set allocation tags from DC GVA instruction. Fields:

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

NSDESC unsigned int

The security state of the access.

PADDR unsigned int

The physical address of the access.

PAS enum

The physical address space of the page.

RESPONSE enum

0=Aborted, 1=OK.

SIZE unsigned int

The size of memory in bytes.

TAG unsigned int

The value of the tag being set.

TAGGING_TYPE enum

Type of Memory tagging (Physical or Virtual).

TAG_VADDR unsigned int

The virtual address of the Virtual Tag if Virtual tagging is used; otherwise 0.

VADDR unsigned int

Virtual address used for setting allocation tags.

DATA_CACHE_ZERO

Zero and invalidate cache line from DC ZVA instruction. Fields:

ATTR unsigned int

Transaction Attributes: [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

NSDESC unsigned int

The security state of the access.

PADDR unsigned int

The physical address of the access.

PADDR_TAG unsigned int

If tag checked, the physical address tag for the access.

PAS enum

The physical address space of the page.

RESPONSE enum

0=Aborted, 1=OK.

SIZE unsigned int

The size of zero'd memory in bytes.

TAG_CHECKED bool

Is this access tag checked.

VADDR unsigned int

Virtual address of the zero'd cache line.

DEBUG_EVENT

Hardware debug support event. Fields:

EVENT enum

Description of event.

VALUE unsigned int

data value.

DMI_ALLOCATE

DMI allocated. Fields:

ACCESS_DIRECTION enum

READ / WRITE.

PADDR unsigned int

The physical address of the DMI.

PTR unsigned int

DMI range pointer.

SIDE enum

DSIDE / ISIDE.

SIZE unsigned int

The size of DMI.

VADDR unsigned int

The virtual address of the DMI..

DMI_HIT

DMI hit, access used DMI (only fire for small DMI regions i.e. less than a full page). Fields:

ACCESS_DIRECTION enum

READ / WRITE.

PADDR unsigned int

The physical address of the DMI.

SIDE enum

DSIDE / ISIDE.

SIZE unsigned int

The size of the DMI.

VADDR unsigned int

The virtual address of the DMI..

DMI_REVOKE

DMI revoked. Fields:

ACCESS_DIRECTION enum

READ / WRITE.

PADDR unsigned int

The physical address of the DMI.

SIDE enum

DSIDE / ISIDE.

SIZE unsigned int

The size of DMI.

VADDR unsigned int

The virtual address of the DMI..

DWT_MATCH

DWT comparator matches. Fields:

NUM unsigned int

DWT comparator number.

TYPE enum

DWT comparator configuration.

END_COMPILE

Compilation end. Fields:

END_OF_PAGE_COUNT unsigned int

Number of basic blocks exited due to page boundary since START_COMPILE.

FETCHFAIL_COUNT unsigned int

Number of basic blocks exited due to ifetch failure START_COMPILE.

INST_COUNT unsigned int

Number of instructions compiled since START_COMPILE.

NONSEQ_COUNT unsigned int

Number of basic blocks exited due to non-sequential instructions since START_COMPILE.

PAGE_STRADDLE_COUNT unsigned int

Number of basic blocks exited due to unaligned instructions crossing page since START_COMPILE.

EXCEPTION

Exceptions that are taken. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ISSET enum

The instruction set of the processor when the exception occurred.

LR unsigned int

The value assigned to the link register.

PC unsigned int

The location where the exception occurred.

TARGET_ISET enum

The instruction set of the exception handler code.

TARGET_PC unsigned int

The address the exception branches to.

VECTOR enum

The exception vector.

EXCEPTION_END

Every exception completed.

EXCEPTION_ENTRY

Event marking the entry of an exception. All pushes of registers to the stack will follow this event. Fields:

PC unsigned int

The location where the exception occurred.

VECTOR enum

The exception that occurred.

EXCEPTION_RAISE

Every exception raised.

EXCEPTION_RETURN

Marks the end of an exception, but on an M core does not indicate a program flow change (branch). In most situations it will be followed by an indirect branch. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The core's instruction counter, starting at 1 for the first instruction.

ISSET enum

The instruction set of the branch instruction.

PC unsigned int

The address of the branch instruction.

EXCEPTION_RETURN_PREBRANCH

Exception return event that must occur before PC and processor state have updated.

EXCEPTION_START

Every exception started.

EXCEPTION_VECTOR_FETCH

I-side vector fetch for M-class exception. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

DATA unsigned int

Result of fetch if OK.

RESPONSE enum

0=Aborted, 1=OK.

VADDR unsigned int

Where the vector is fetched from (including VTOR).

FP_STATE

Floating point state. Fields:

FPSR unsigned int

FPSR register value.

MASK unsigned int

The mask indicating updated cumulative bits.

FREQ_CHANGED

Core frequency changed. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The instruction count of this core.

NEW_FREQ signed int

The new frequency of this core (expressed in Hz).

OLD_FREQ signed int

The old frequency of this core (expressed in Hz).

HLT

HLT instruction occurred. Fields:

IMM16 unsigned int

The 16-bit immediate value encoded in HLT instruction.

INFO_EXCEPTION_REASON

Provide information of real cause of exception. Fields:

FaultCause enum

Which bit of which V7M/V8M-mainline FSR/HFSR would have been set.

PC unsigned int

The location where the exception occurred.

PHASE enum

What we are doing with the exception.

REASONS enum

Why we are doing it.

VECTOR enum

The exception that occurred.

INFO_STACKING

Provide information of real cause of exception. Fields:

PHASE enum

Are we starting or ending a sequence of stack writes.

REASONS enum

Why we are doing it.

INST

Every instruction executed. Fields:

CORE_NUM *unsigned int*

Core number in a multi processor.

CURRENT_TIME *unsigned int*

The core's current time, as simulated time plus local time.

DEBUG_STATE *bool*

The instruction is executed in debug state.

DISASS *string*

Disassembly of instruction.

INST_COUNT *unsigned int*

The core's instruction counter, starting at 1 for the first instruction.

ISSET *enum*

The current instruction set.

ITSTATE *unsigned int*

The current ITSTATE.

LOCAL_TIME *unsigned int*

The core's local time, relative to the current quantum.

MODE *enum*

The mode the core is in.

NSDESC *unsigned int*

The physical address non-secure bit.

NSDESC2 *unsigned int*

The second page physical address non-secure bit.

OPCODE *unsigned int*

The opcode of the instruction.

PADDR *unsigned int*

The physical address of the instruction.

PADDR2 *unsigned int*

If different from PADDR, the physical address of the second page of the instruction.

PAS *enum*

The physical address space of the page.

PAS2 *enum*

The physical address space of the page.

PC *unsigned int*

The address of the instruction.

SECURITY_STATE *enum*

The core's security state.

SIZE unsigned int

The size of the instruction in bytes.

INST_END

Every instruction completed.

INST_START

Every instruction started. Fields:

DEBUG_STATE bool

The instruction is executed in debug state.

ISET enum

The current instruction set.

MODE enum

The mode the core is in.

NS enum

The current Secure State.

PC unsigned int

The address of the conditional instruction.

SECURITY_STATE enum

The current Security State.

INST_STRADDLE

Instruction straddles boundary. Fields:

DEBUG_STATE bool

The instruction is executed in debug state.

ISET enum

The current instruction set.

MODE enum

The mode the core is in.

NS enum

The current Secure State.

PC unsigned int

The address of the conditional instruction.

SECURITY_STATE enum

The current Security State.

ISB

ISB instruction occurred. Fields:

PC unsigned int

The address of the ISB instruction.

ITM

Instrumentation Trace Macrocell. Fields:

ITM_PACKET_TYPE enum

ITM and DWT packets type.

PACKET_HEADER unsigned int

ITM Packet Header.

PACKET_PAYLOAD unsigned int

ITM Packet Payload.

LOAD_MULTIPLE_REGS_LIST

List of destination registers (not source) of a multiple registers load instruction. Fields:

REG string

Register to add into the list.

LOCAL_MONITOR

Local monitor activity. Fields:

PADDR unsigned int

Local Monitor Address.

State enum

State of the monitor (Open/Exclusive).

LOCKUP_CYCLE

This event is triggered if the core enters or stays in the lock-up state. Fields:

CAUSE enum

What causes the lockup?.

LOCKUP_ENTRY

This event is triggered if the core enters the lock-up state. Fields:

PC unsigned int

The PC when the lockup state is entered.

LOCKUP_EXIT

This event is triggered if the core leaves the lock-up state.

MODE_CHANGE

Mode change. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

MODE enum

The new mode.

OLD_MODE enum

The old mode.

MPU_TRANS

Address translation information. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

EXEC_PERM enum

Execution Permission.

INNERCACHE_RA bool

Is the inner cache allocate on read.

INNERCACHE_TYPE enum

Inner Caching scheme (NC/MB/WA).

INNERCACHE_WA bool

Is the inner cache allocate on write.

MEMTYPE enum

Memory type.

OUTERCACHE_RA bool

Is the outer cache allocate on read.

OUTERCACHE_TYPE enum

Outer Caching scheme (NC/MB/WA).

OUTERCACHE_WA bool

Is the outer cache allocate on write.

PADDR unsigned int

Address of the access.

PAGESIZE unsigned int

Page size as log2(size).

READ_PERM enum

Read Permission.

REG_NUM unsigned int

MPU register based region-number 0-15 (else 0xFF=none, 0xBB/0xDD=background/default).

SH enum

Shareability.

SIDE enum

Inst / Data.

WRITE_PERM enum

Write Permission.

PERIODIC

Called for every quantum. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The instruction count of this CPU.

PC unsigned int

The address of the next instruction to be executed on this CPU.

POTINDEX_UPDATE

Value of POT index for current pc changed. Fields:

NEW_POTINDEX unsigned int

New POTIndex.

OLD_POTINDEX unsigned int

Old POTIndex.

PPB_READ_ACCESS

Trace reads from the core to memory mapped registers in the PPB (private peripheral bus) address range. Fields:

ACCESS_SIZE unsigned int

Logarithm of the access size: 0=byte, 1=halfword, 2=word, 3=doubleword, ...

BURST_LEN unsigned int

The number of data transfers (beats) in this burst.

DATA unsigned int

The data transferred.

INNER_ATTRIBUTES unsigned int

The inner attributes: bit0=write allocate, bit1=read allocate, bit2=cachable, bit3=bufferable, bit4=shareable.

INST bool

Is an instruction fetch.

LOCK enum

Normal / Locked / Exclusive.

MANAGER_ID unsigned int

The AXI manager ID.

NS bool

Is a nonsecure access.

OUTER_ATTRIBUTES unsigned int

The outer attributes: bit0=write allocate, bit1=read allocate, bit2=cachable, bit3=bufferable, bit4=shareable.

PADDR unsigned int

Physical address of access.

PRIV bool

Is privileged access.

RESPONSE enum

Whether the transaction was successful, or an error occurred.

USER_FLAGS unsigned int

Core specific additional signals.

PPB_WRITE_ACCESS

Trace writes from the core to memory mapped registers in the PPB (private peripheral bus) address range. Fields:

ACCESS_SIZE unsigned int

Logarithm of the access size: 0=byte, 1=halfword, 2=word, 3=doubleword, ...

BURST_LEN unsigned int

The number of data transfers (beats) in this burst.

DATA unsigned int

The data transferred.

INNER_ATTRIBUTES unsigned int

The inner attributes: bit0=write allocate, bit1=read allocate, bit2=cachable, bit3=bufferable, bit4=shareable.

INST bool

Is an instruction fetch.

LOCK enum

Normal / Locked / Exclusive.

MANAGER_ID unsigned int

The AXI manager ID.

NS bool

Is a nonsecure access.

OUTER_ATTRIBUTES unsigned int

The outer attributes: bit0=write allocate, bit1=read allocate, bit2=cachable, bit3=bufferable, bit4=shareable.

PADDR unsigned int

Physical address of access.

PRIV bool

Is privileged access.

RESPONSE enum

Whether the transaction was successful, or an error occurred.

USER_FLAGS unsigned int

Core specific additional signals.

PRELOAD_DATA

Data preload from PLD instruction. Fields:

VADDR unsigned int

Virtual address of the data that should be preloaded.

PRELOAD_INST

Instruction preload from PLI instruction. Fields:

VADDR unsigned int

Virtual address of the instruction that should be preloaded.

PRE_CORE_LOAD

Trace just before a core load. Fields:

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

PRE_CORE_STORE

Trace just before a core store. Fields:

CURRENT_TIME unsigned int

The core's current time, as simulated time plus local time.

LOCAL_TIME unsigned int

The core's local time, relative to the current quantum.

PSTATE_PACTIVE_OPMODE_TRANSITION

Trace pactive and Opmode mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Operating mode after the transition.

OLD_STATE enum

Operating mode before the transition.

PSTATE_PACTIVE_OPMODE_TRANSITION_DENIED

Trace the denied pactive and Opmode mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Operating mode after the transition.

OLD_STATE enum

Operating mode before the transition.

REASON string

Reason for the transition being denied.

PSTATE_PACTIVE_POWER_TRANSITION

Trace pactive and power mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Power mode after the transition.

OLD_STATE enum

Power mode before the transition.

PSTATE_PACTIVE_POWER_TRANSITION_DENIED

Trace the denied pactive and power mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Power mode after the transition.

OLD_STATE enum

Power mode before the transition.

REASON string

Reason for the transition being denied.

PSTATE_PACTIVE_THREAD_OPMODE_TRANSITION

Trace pactive and Opmode mode transitions for threads on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Operating mode after the transition.

OLD_STATE enum

Operating mode before the transition.

QUANTUM_START

Called at the start of every quantum.

RUN_STATE

Run state transition. Fields:

INST_COUNT unsigned int

Ticks count at point of transition.

NEW enum

New run state.

OLD enum

Old run state.

SEMIHOSTING_CALL

Call of a semihost function occurred. Fields:

PC unsigned int

The program counter after the semihosting call.

SEMIHOSTING_PRECALL

About to call semihost. Fields:

EL2 bool

Use EL2 translation regime.

NS bool

Is Non-Secure.

REG_WIDTH unsigned int

The current register width in bytes.

SIGNAL

External signal state change. Fields:

SIGNAL enum

Signal that changed.

STATE bool

Signal asserted state.

START_COMPILE

Compilation started. Fields:

VADDR unsigned int

Instruction where compilation begins.

STG_MULTIPLE_MICRO_OPS_ADDRESS_RANGE

Address range for MTE stores with multiple uops. Fields:

ADDR_END unsigned int

Final address.

ADDR_START unsigned int

Starting address.

STORE_MULTIPLE_REGS_ADDRESS_RANGE

Address range for stores with multiple registers. Fields:

ADDR_END unsigned int

Final address.

ADDR_START unsigned int

Starting address.

SUBOPTIMAL_LDST_RETIRED

suboptimal load/store. Fields:

ACCESS_SIZE unsigned int

Log2 of the access width in bytes used for alignment checking (1: half-word, 2: word ...).

OPTIMAL_ALIGN_SELECTOR unsigned int

Selector to decide the optimal alignment size (1: access width, 2: 4-bytes, 3: 8-bytes, ..., 12: 4-kbytes).

VADDR unsigned int

The virtual address of the access.

SYNC

Called for every synchronization, that is every quantum break. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

INST_COUNT unsigned int

The instruction count of this CPU.

LOCAL_QUANTUM unsigned int

The local quantum of this CPU.

LOCAL_TIME unsigned int

The local time of this CPU.

SYNC_REASON enum

Reason for sync.

SYSCALL

System call instruction executed. Fields:

IMM unsigned int

Immediate value of the system call instruction.

TYPE enum

System call type.

VADDR unsigned int

Instruction that caused the system call.

SYSREG_READ

Trace reads from system registers. Fields:

REG_NAME enum

Register number.

VALUE unsigned int

The value read.

SYSREG_UPDATE128

Triggers when the system updates a register. Fields:

REG enum

Register number.

UNKNOWN unsigned int

Bits of the register which became unknown.

VALUE unsigned int

Value written to the register.

SYSREG_UPDATE32

Triggers when the system updates a register. Fields:

REG enum

Register number.

UNKNOWN unsigned int

Bits of the register which became unknown.

VALUE unsigned int

Value written to the register.

SYSREG_UPDATE64

Triggers when the system updates a register. Fields:

REG enum

Register number.

UNKNOWN unsigned int

Bits of the register which became unknown.

VALUE unsigned int

Value written to the register.

SYSREG_WRITE

Trace write to system registers. Fields:

REG_ACCESS enum

Register access status.

REG_NAME enum

Register number.

UPDATED_VALUE unsigned int

Updated value of the register now it has been written.

VALUE unsigned int

Value written to the register.

UNALIGNED_LDST_RETIRED

Processor unaligned load/store. Fields:

ACCESS_SIZE unsigned int

Log2 of the access width in bytes used for alignment checking (1=>half-word, 2->word ...).

NUMBER_OF_BEATS unsigned int

Number of accesses (beats) in this burst (Total size in bytes is ACCESS_SIZE * NUMBER_OF_BEATS).

VADDR unsigned int

The virtual address of the access.

WAYPOINT

Signals end of basic block execution. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

ISET enum

Origin instruction set.

IS_COND bool

Indicates if this is a conditional waypoint.

PC unsigned int

Origin address (or 0 if unavailable).

TAKEN bool

Indicates if this waypoint was taken.

TARGET unsigned int

Destination address.

TARGET_ISET enum

Destination instruction set.

WFE_END

WFE ended. Fields:

INST_COUNT unsigned int

Ticks count when leaving WFE.

WFE_EVENT_REGISTER

WFE event register status: set/clear, reason. Fields:

INST_COUNT unsigned int

Ticks count.

REASON enum

Reason for set/clear. Only REASON==0 (Cleared by WFE) clears the bit, all other reasons set it.

WFE_IGNORED

WFE ignored. Fields:

EVENT bool

This WFE was ignored because the event register was set.

INST_COUNT unsigned int

Ticks count when ignoring WFE.

TRAPPED bool

This WFE was trapped.

WFE_START

WFE entered. Fields:

INST_COUNT unsigned int

Ticks count when entering WFE.

WFI_END

WFI ended. Fields:

INST_COUNT unsigned int

Ticks count when leaving WFI.

WFI_IGNORED

WFI ignored. Fields:

DISABLED bool

This WFI was ignored because WFI is disabled.

INST_COUNT unsigned int

Ticks count when ignoring WFI.

REASON enum

specifies reason why WFI trace was ignored.

TRAPPED bool

This WFI was trapped.

WFI_START

WFI entered. Fields:

INST_COUNT unsigned int

Ticks count when entering WFI.

WFI_WAKEUP

WFI wakeup. Fields:

INST_COUNT unsigned int

Ticks count when WFI wakeup occurred.

REASON enum

Reason for wakeup.

xPSR

Changes to the xPSR register. Fields:

CORE_NUM unsigned int

Core number in a multi processor.

OLD_VALUE unsigned int

The old xPSR value.

UNKNOWN unsigned int

Bits within the register that have unknown value.

VALUE unsigned int

The new xPSR value.

2.73 AsyncCacheFlushUnit

This section describes the trace sources.

STATE_CHANGE

Change in state of the AsyncCacheFlushUnit state machine. Fields:

STATE enum

New state.

2.74 Base_PowerController

This section describes the trace sources.

PWRC_register_read

A register read occurred. Fields:

data unsigned int

The data returned if not an error.

error bool

An error is being returned.

ns bool

The security state of the access.

offset enum

The offset of the register access.

PWRC_register_write

A register write occurred. Fields:

data unsigned int

The data written if not an error.

error bool

An error is being returned.

ns bool

The security state of the access.

offset enum

The offset of the register access.

PWRC_signal_input

signal inputs. Fields:

index unsigned int

index into port array.

signal enum

input signal.

value bool

whether signal is asserted or not.

PWRC_signal_output

signal outputs. Fields:

index unsigned int

index into port array.

signal enum

output signal.

value bool

whether signal is asserted or not.

PWRC_sys_status_update

SYS_STATUS register is updated. Fields:

affinity unsigned int

affinity level identifier of the core which is changing.

data unsigned int

new value of SYS_STATUS.

index unsigned int

pin index to which this core is wired.

2.75 C1-DSU

This section describes the trace sources.

PSTATE_PACTIVE_OPMODE_TRANSITION

Trace pactive and Opmode mode transitions on PChannel. Fields:

Copyright © 2023–2025 Arm Limited (or its affiliates). All rights reserved.

Non-Confidential

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Operating mode after the transition.

OLD_STATE enum

Operating mode before the transition.

PSTATE_PACTIVE_OPMODE_TRANSITION_DENIED

Trace the denied pactive and Opmode mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Operating mode after the transition.

OLD_STATE enum

Operating mode before the transition.

REASON string

Reason for the transition being denied.

PSTATE_PACTIVE_POWER_TRANSITION

Trace pactive and power mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Power mode after the transition.

OLD_STATE enum

Power mode before the transition.

PSTATE_PACTIVE_POWER_TRANSITION_DENIED

Trace the denied pactive and power mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Power mode after the transition.

OLD_STATE enum

Power mode before the transition.

REASON string

Reason for the transition being denied.

dsu.set_defaulttmp

DISPLAY DEFAULTTMP port is set to %{value}. Fields:

value bool

Value driven on DEFAULTMP pin.

dsu.set_peripheral_range

DISPLAY `{isStart:AENDMP|ASTARTMP}[{index}]` is set to `{isStart:{end_address}|{start_address}}`, so that valid peripheral address range becomes `[{start_address},{end_address}]`. Fields:

end_address unsigned int

End address of this peripheral region.

index unsigned int

Index of ASTARTMP/AENDMP peripheral port. Can be between 0 and 3.

isStart bool

If ASTARTMP is set, true, false otherwise.

start_address unsigned int

Start address of this peripheral region.

dsu.utility_bus.access

DISPLAY `{is_debug:Normal|Debug} {is_read:write|read}` access has been made to UtilityBus to access `{component_accessed} {access_is_to_cluster_reg:to {target_core}core}`. Fields:

access_is_to_cluster_reg bool

true if this access is against cluster registers, false otherwise.

component_accessed string

component to which this transaction is made.

is_debug bool

true if the access is a debug one, false otherwise.

is_read bool

true if the access is a read one, false otherwise.

offset unsigned int

Offset to the utility bus base address used to identify which register is being accessed.

target_core unsigned int

is the target core if the transaction is to core register.

dsu.utility_bus.access_is_razwi

DISPLAY `{is_read:Write|Read}` access has been made to UtilityBus with offset `{offset}`, but this access results in **RAZWI** because `{msg}`. Fields:

is_read bool

true if the access is a read one, false otherwise.

msg string

the reason why this access is **RAZWI**.

offset unsigned int

Offset to the base address of utility bus used to identify which register is being accessed.

2.76 CCI400

This section describes the trace sources.

CCI_REGISTER_READ

A valid read from the register file. (Access violations are not traced.). Fields:

Offset unsigned int

Offset into the register block.

Value unsigned int

Value read.

CCI_REGISTER_WRITE

A valid write to the register file. (Access violations are not traced.). Fields:

Offset unsigned int

Offset into the register block.

Value unsigned int

Value written.

2.77 CCI500

This section describes the trace sources.

ArchMsg.Error.reset_state_of_upstream_is_in_reset_but_trying_to_enable_snoops_andor_dvm

The upstream system is known to be in reset but we are trying to enable snoops/DVM to the port. In a real system, this could deadlock as snoop transactions could get routed to the port unless there were absolutely no shared transactions in flight or made. In any event, it is highly unlikely that the programmer intended this. Fields:

is_dvm_enable bool

Is DVM enable (rather than snoop enable).

upstream_port_index unsigned int

The upstream port index of the reset signal.

ArchMsg.Error.sw_reset

An error occurred in managing the reset of upstream systems whose entrance and exit of the coherency domain is managed by software. Fields:

why string

Why the problem occurred.

ArchMsg.Warning.acchannelensx_enables_an_upstream_port_with_no_snoop_channel

ACCHANNELENSx should tell us which upstream ports have a connected snoop and DVM channel. The model has detected that an enabled port does not have a snoop/DVM channel. This is silently ignored in the model, but might have deadlocked the hardware. `DISPLAY ACCHANNELENS({acchannelensx:d})` enables snoops to upstream port `{upstream_port_index}` which has no snoop channel, this might deadlock HW. Fields:

acchannelensx unsigned int

The value of ACCHANNELENSx that is being used.

upstream_port_index unsigned int

Upstream port index of offending port.

ArchMsg.Warning.odd_programming_trying_to_enable_upstream_port

DESCRIPTION: The software tried to enable `{which}` messages on port `{upstream_port_index}` that is not capable of receiving these messages, for example by being disallowed by `ACCHANNELENS{upstream_port_index}` or entirely disconnected. Fields:

acchannelensx unsigned int

The ACCHANNELENSx bitmap of upstream ports.

bitmap_of_connected_upstream_ports unsigned int

This is the bitmap of upstream ports that are actually connected.

bitmap_of_dvm_upstream_ports unsigned int

The bitmap of DVM-capable upstream ports.

bitmap_of_full_ace_upstream_ports unsigned int

The bitmap of snoop-capable upstream ports.

upstream_port_index unsigned int

The upstream port index of the port.

which enum

Which type of messages we were trying to enable.

ArchMsg.Warning.reset_state_of_upstream_port_driven_after_first_transaction_seen

All signals should be driven at simulation reset so that both sides of the connection agree on the value of the wire. This trace source means that the driver of the signal did not do this. `DISPLAY reset_state_of_upstream_port[{upstream_port_index}]` first driven after first transaction. Fields:

upstream_port_index unsigned int

The upstream port index of the reset signal.

value bool

The value driven.

ArchMsg.Warning.reset_state_of_upstream_system_changed_whilest_a_snoop_change_is_pending

If snoop change request is pending, and the upstream system is reset during that process then the system could deadlock if there is any possibility of a shared transaction being in flight at the time. The upstream port might have an outstanding snoop request when it is reset and the interconnect would wait indefinitely for the response. DISPLAY Upstream system %{upstream_port_index} reset whilst a snoop request may be upstream due to a snoop change pending. Fields:

upstream_port_index unsigned int

The upstream port index of the reset signal.

value bool

The value driven.

why enum

Why this message is emitted.

ArchMsg.Warning.shared_transaction_received_from_upstream_port_where_acchannelensx_is_disabled

ACCHANNELENSx tells the interconnect which upstream ports do not have connected snoop channels. This warns when a shared transaction is received from an upstream ACE port that doesn't have its snoop channel enabled by ACCHANNELENSx and there are other ports. This may be OK if the upstream system does not accept Snoop/DVM and so is purely an ACE-Lite system, or the upstream systems are truly separate shareability domains. However, this might also mean that you have completely forgotten to program the interconnect or use the parameter force_on_from_start. This will only be warned *once* between resets of the interconnect. DISPLAY shared transaction received from upstream %{upstream_port_index} and ACCHANNELENS%{upstream_port_index} for that port is disabled. Fields:

upstream_port_index unsigned int

Upstream port index of offending port.

DVM_Message

DISPLAY DVM_msg (%{DVM_MESSAGE}). Fields:

DVM_MESSAGE string

The DVM Message.

allowed_register_access

An allowed register access to the register file. Fields:

data unsigned int

The data read/written. If this is a read, it is only valid if result is ok.

is_debug bool

Is the transaction a debug transaction?.

is_read bool

Is the transaction a read transaction?.

ns bool

The transaction is non-secure.

register enum

The register of the transaction.

change_dvm_disabled_ports

This reports a bitmap of which ports are disabled to receive DVM messages. Fields:

new_bitmap unsigned int

The new bitmap of disabled ports.

new_bitmap_list unsigned int

A list of port numbers that were disabled.

old_bitmap unsigned int

The old bitmap of disabled ports.

old_bitmap_list unsigned int

A list of port numbers that were disabled.

change_upstream_disabled_ports

This reports a bitmap of which upstream ports of the cache are *disabled* and so no snoop requests are sent to them. Fields:

new_bitmap unsigned int

The bitmap of the first 32 upstream ports that are disabled.

new_bitmap_list unsigned int

A list of port numbers that were disabled.

old_bitmap unsigned int

The bitmap of the first 32 upstream ports that are disabled.

old_bitmap_list unsigned int

A list of port numbers that were disabled.

configuration_pins

One of the configuration pins changed. In the RTL these are sampled at reset, but in the model, we sample it at first transaction. If it occurs after this point then it is traced but the field 'ignored' will be true and the value reported will be what it wanted to change to rather than the value it actually is. Fields:

ignored bool

The value was ignored because it occurred after seeing a transaction.

value unsigned int

The value it wants to change to.

which enum

Which configuration pin changed.

debug_enables

The signals/parameters that cause whether events are counted or not. Fields:

dbgen bool

DBGEN value.

niden bool

NIDEN value.

non_secure_traceable bool

Non-secure events are traceable.

secure_traceable bool

Secure events are traceable.

spiden bool

SPIDEN value.

spniden bool

SPNIDEN value.

errirq

This is the error interrupt signal that is generated when there are unacknowledged imprecise errors. Fields:

value bool

The value it is driven to.

evntcntoverflow

The event counter overflow interrupt lines. Fields:

counter_id unsigned int

The ID of the counter that this relates to.

value bool

The value of the interrupt line.

force_on_from_start

A commentary on what force_on_from_start is doing. Fields:

acchannelen_dvm unsigned int

The bitmap of which upstream ports are capable of receiving DVM messages. If the appropriate bit is clear in this register then we won't enable it.

acchannelen_snoop unsigned int

The bitmap of which upstream ports are capable of receiving snoop messages. If the appropriate bit is clear in this register then we won't enable it.

enable bool

Whether we are trying to enable the upstream port or not.

reset_state_of_upstream_system enum

What the interconnect knows about the reset state of the upstream system.

upstream_port_index unsigned int

The upstream port index.

why enum

Why the value changed.

ignored_register_access

The register access is ignored for some reason. Fields:

address enum

The address.

is_debug bool

Is the access a debug access?.

is_read bool

Is the access a read?.

ns bool

The security state of the transaction.

why enum

Why the access was ignored.

imprecise_error_received

An imprecise error was received for the specified transaction. Fields:

address unsigned int

Address of transaction.

downstream_port_index unsigned int

The downstream port index that the transaction went out.

ns bool

Is the transaction non-secure.

pmu_overflow

The performance monitor counter overflowed. Fields:

counter_id unsigned int

The ID of the counter that this relates to.

event_counting unsigned int

The event that the counter is counting.

new_value bool

The new value of the overflowed flag.

old_value bool

The old value of the overflowed flag.

pmu_programmed

A programmed change to the performance monitor unit counter. Fields:

counter_id unsigned int

The ID of the counter that this relates to.

current_count unsigned int

The current value of the counter.

enabled bool

Is the counter enabled?.

event_counting unsigned int

The event that the counter is counting.

why enum

why.

register_access

Read/Write of the register file. Fields:

address unsigned int

The address of the transaction.

data unsigned int

The data read/written. If this is a read, it is only valid if result is ok.

is_debug bool

Is the transaction a debug transaction?.

is_read bool

Is the transaction a read transaction?.

ns bool

Is the transaction non-secure?.

number_of_beats unsigned int

The number of beats in the transaction.

result enum

The result of the transaction.

width_in_bytes unsigned int

Width of the transaction in bytes.

reset_state_of_upstream_port

The pins `reset_state_of_upstream_port[N]` are intended to be use as checks by the interconnect that the programmer has disabled snoops and DVMs to upstream ports whose systems are in reset. If you connect these ports to a signal that indicates if the upstream port is in reset then the interconnect can perform this check. This trace source is the trace of those reset state changes. Fields:

upstream_port_index unsigned int

The upstream port index of the reset signal.

value bool

The value driven.

upstream_port_with_an_unsynced_snoop_change_pending

When a change to whether to send snoop requests or DVM messages to a port is setup then it takes a finite period of time to take effect and the SW should poll for completion. This tells you when an upstream port has a snoop change pending on it that hasn't been polled for completion. Fields:

comment enum

Comment.

pending bool

Whether the change is pending or not.

upstream_port_index unsigned int

Upstream port index.

2.78 CCI550

This section describes the trace sources.

ArchMsg.Error.reset_state_of_upstream_is_in_reset_but_trying_to_enable_snoops_andor_dvm

The upstream system is known to be in reset but we are trying to enable snoops/DVM to the port. In a real system, this could deadlock as snoop transactions could get routed to the port unless there were absolutely no shared transactions in flight or made. In any event, it is highly unlikely that the programmer intended this. Fields:

is_dvm_enable bool

Is DVM enable (rather than snoop enable).

upstream_port_index unsigned int

The upstream port index of the reset signal.

ArchMsg.Error.sw_reset

An error occurred in managing the reset of upstream systems whose entrance and exit of the coherency domain is managed by software. Fields:

why string

Why the problem occurred.

ArchMsg.Warning.acchannelensx_enables_an_upstream_port_with_no_snoop_channel

ACCHANNELENSx should tell us which upstream ports have a connected snoop and DVM channel. The model has detected that an enabled port does not have a snoop/DVM channel. This is silently ignored in the model, but might have deadlocked the hardware. DISPLAY ACCHANNELENS({acchannelensx:d}) enables snoops to upstream port {upstream_port_index} which has no snoop channel, this might deadlock HW. Fields:

acchannelensx unsigned int

The value of ACCHANNELENSx that is being used.

upstream_port_index unsigned int

Upstream port index of offending port.

ArchMsg.Warning.odd_programming_trying_to_enable_upstream_port

DESCRIPTION: The software tried to enable %{which} messages on port %{upstream_port_index} that is not capable of receiving these messages, for example by being disallowed by ACCHANNELENSX%{upstream_port_index} or entirely disconnected. Fields:

acchannelensx unsigned int

The ACCHANNELENSx bitmap of upstream ports.

bitmap_of_connected_upstream_ports unsigned int

This is the bitmap of upstream ports that are actually connected.

bitmap_of_dvm_upstream_ports unsigned int

The bitmap of DVM-capable upstream ports.

bitmap_of_full_ace_upstream_ports unsigned int

The bitmap of snoop-capable upstream ports.

upstream_port_index unsigned int

The upstream port index of the port.

which enum

Which type of messages we were trying to enable.

ArchMsg.Warning.reset_state_of_upstream_port_driven_after_first_transaction_seen

All signals should be driven at simulation reset so that both sides of the connection agree on the value of the wire. This trace source means that the driver of the signal did not do this. DISPLAY reset_state_of_upstream_port[%{upstream_port_index}] first driven after first transaction. Fields:

upstream_port_index unsigned int

The upstream port index of the reset signal.

value bool

The value driven.

ArchMsg.Warning.reset_state_of_upstream_system_changedWhilst_a_snoop_change_is_pending

If snoop change request is pending, and the upstream system is reset during that process then the system could deadlock if there is any possibility of a shared transaction being in flight at the time. The upstream port might have an outstanding snoop request when it is reset and the interconnect would wait indefinitely for the response. DISPLAY Upstream system %{upstream_port_index} reset whilst a snoop request may be upstream due to a snoop change pending. Fields:

upstream_port_index unsigned int

The upstream port index of the reset signal.

value bool

The value driven.

why enum

Why this message is emitted.

ArchMsg.Warning.shared_transaction_received_from_upstream_port_where_acchannelensx_is_disabled

ACCHANNELENSx tells the interconnect which upstream ports do not have connected snoop channels. This warns when a shared transaction is received from an upstream ACE port that doesn't have its snoop channel enabled by ACCHANNELENSx and there are other ports. This may be OK if the upstream system does not accept Snoop/DVM and so is purely an ACE-Lite system, or the upstream systems are truly separate shareability domains. However, this might also mean that you have completely forgotten to program the interconnect or use the parameter `force_on_from_start`. This will only be warned *once* between resets of the interconnect. DISPLAY shared transaction received from upstream `%{upstream_port_index}` and ACCHANNELENS`%{upstream_port_index}` for that port is disabled. Fields:

upstream_port_index unsigned int

Upstream port index of offending port.

DVM_Message

DISPLAY DVM_msg (`%{DVM_MESSAGE}`). Fields:

DVM_MESSAGE string

The DVM Message.

allowed_register_access

An allowed register access to the register file. Fields:

data unsigned int

The data read/written. If this is a read, it is only valid if result is ok.

is_debug bool

Is the transaction a debug transaction?.

is_read bool

Is the transaction a read transaction?.

ns bool

The transaction is non-secure.

register enum

The register of the transaction.

change_dvm_disabled_ports

This reports a bitmap of which ports are disabled to receive DVM messages. Fields:

new_bitmap unsigned int

The new bitmap of disabled ports.

new_bitmap_list unsigned int

A list of port numbers that were disabled.

old_bitmap unsigned int

The old bitmap of disabled ports.

old_bitmap_list unsigned int

A list of port numbers that were disabled.

change_upstream_disabled_ports

This reports a bitmap of which upstream ports of the cache are *disabled* and so no snoop requests are sent to them. Fields:

new_bitmap unsigned int

The bitmap of the first 32 upstream ports that are disabled.

new_bitmap_list unsigned int

A list of port numbers that were disabled.

old_bitmap unsigned int

The bitmap of the first 32 upstream ports that are disabled.

old_bitmap_list unsigned int

A list of port numbers that were disabled.

configuration_pins

One of the configuration pins changed. In the RTL these are sampled at reset, but in the model, we sample it at first transaction. If it occurs after this point then it is traced but the field 'ignored' will be true and the value reported will be what it wanted to change to rather than the value it actually is. Fields:

ignored bool

The value was ignored because it occurred after seeing a transaction.

value unsigned int

The value it wants to change to.

which enum

Which configuration pin changed.

debug_enables

The signals/parameters that cause whether events are counted or not. Fields:

dbgen bool

DBGGEN value.

niden bool

NIDEN value.

non_secure_traceable bool

Non-secure events are traceable.

secure_traceable bool

Secure events are traceable.

spiden bool

SPIDEN value.

spniden bool

SPNIDEN value.

errirq

This is the error interrupt signal that is generated when there are unacknowledged imprecise errors. Fields:

value bool

The value it is driven to.

evntcntoverflow

The event counter overflow interrupt lines. Fields:

counter_id unsigned int

The ID of the counter that this relates to.

value bool

The value of the interrupt line.

force_on_from_start

A commentary on what force_on_from_start is doing. Fields:

acchannelen_dvm unsigned int

The bitmap of which upstream ports are capable of receiving DVM messages. If the appropriate bit is clear in this register then we won't enable it.

acchannelen_snoop unsigned int

The bitmap of which upstream ports are capable of receiving snoop messages. If the appropriate bit is clear in this register then we won't enable it.

enable bool

Whether we are trying to enable the upstream port or not.

reset_state_of_upstream_system enum

What the interconnect knows about the reset state of the upstream system.

upstream_port_index unsigned int

The upstream port index.

why enum

Why the value changed.

ignored_register_access

The register access is ignored for some reason. Fields:

address enum

The address.

is_debug bool

Is the access a debug access?.

is_read bool

Is the access a read?.

ns bool

The security state of the transaction.

why enum

Why the access was ignored.

imprecise_error_received

An imprecise error was received for the specified transaction. Fields:

address unsigned int

Address of transaction.

downstream_port_index unsigned int

The downstream port index that the transaction went out.

ns bool

Is the transaction non-secure.

pmu_overflow

The performance monitor counter overflowed. Fields:

counter_id unsigned int

The ID of the counter that this relates to.

event_counting unsigned int

The event that the counter is counting.

new_value bool

The new value of the overflowed flag.

old_value bool

The old value of the overflowed flag.

pmu_programmed

A programmed change to the performance monitor unit counter. Fields:

counter_id unsigned int

The ID of the counter that this relates to.

current_count unsigned int

The current value of the counter.

enabled bool

Is the counter enabled?.

event_counting unsigned int

The event that the counter is counting.

why enum

why.

register_access

Read/Write of the register file. Fields:

address unsigned int

The address of the transaction.

data unsigned int

The data read/written. If this is a read, it is only valid if result is ok.

is_debug bool

Is the transaction a debug transaction?.

is_read bool

Is the transaction a read transaction?.

ns bool

Is the transaction non-secure?.

number_of_beats unsigned int

The number of beats in the transaction.

result enum

The result of the transaction.

width_in_bytes unsigned int

Width of the transaction in bytes.

reset_state_of_upstream_port

The pins reset_state_of_upstream_port[N] are intended to be use as checks by the interconnect that the programmer has disabled snoops and DVMs to upstream ports whose systems are in reset. If you connect these ports to a signal that indicates if the upstream port is in reset then the interconnect can perform this check. This trace source is the trace of those reset state changes. Fields:

upstream_port_index unsigned int

The upstream port index of the reset signal.

value bool

The value driven.

upstream_port_with_an_unsynced_snoop_change_pending

When a change to whether to send snoop requests or DVM messages to a port is setup then it takes a finite period of time to take effect and the SW should poll for completion. This tells you when an upstream port has a snoop change pending on it that hasn't been polled for completion. Fields:

comment enum

Comment.

pending bool

Whether the change is pending or not.

upstream_port_index unsigned int

Upstream port index.

2.79 CCNCache

This section describes the trace sources.

ALLOC_LINEFILL

The system allocated a complete line fill into the RAMs. This happens when the cache has to read the data for a line fill from potentially upstream or downstream. Fields:

ENTRY_INDEX unsigned int

The entry index the line has been loaded into.

MANAGER_ID unsigned int

The manager ID for the associated transaction.

MEMORY_ATTRIBUTES unsigned int

The memory attributes used for the allocation. bits[3:0] are the inner ACACHE, bits[7:4] are the outer ACACHE, bits[9:8] are ADOMAIN, bits[12:10] are APROT, the rest are undocumented.

NS_ADDR unsigned int

The security world and address of the first byte of the line, the MSB encodes the NS bit.

PAS enum

The Physical Address Space.

ALLOC_WRITE

The system allocated a complete line write into the RAMs. This happens when the cache receives a complete cache line write. It need not read from anywhere. Fields:

ENTRY_INDEX unsigned int

The entry index the line has been written into.

MEMORY_ATTRIBUTES unsigned int

The memory attributes used for the allocation. bits[3:0] are the inner ACACHE, bits[7:4] are the outer ACACHE, bits[9:8] are ADOMAIN, bits[12:10] are APROT, the rest are undocumented.

NS_ADDR unsigned int

The security world and address of the first byte of the line, the MSB encodes the NS bit.

PAS enum

The Physical Address Space.

ATOMIC_SLAVE_ACCESS

Information on the atomic operation at slave side. Fields:

WT-cacheable|WB-cacheable))%{TXATTR[7:4.14:13]:(nGnRnE|nGRnE?|GnRnE?|GRnE?
nGnRE|nGRE|GnRE?|GRE))%{TXATTR[16]:(|-transient)} shareability: %
{TXATTR[9:8]:(nsh|ish|osh|sys))%{TXATTR[21.24]:(|TranslatedAccess|NonStallable|TransFaultFlow)%
{TXATTR[22]:(|NonAddressBasedRouted))%{TXATTR[25]:(|HasMetaData)%
{TXATTR[26]:(|ReservedImpDef))%{TXATTR[27]:(|-StreamTransaction)%
{TXATTR[28]:(|-nse))%{TXATTR[29]:(|-nse2))%{TXATTR[31]:(|-HasMutatingMetaDataOp)}. PRIMARY KEY NS_ADDR. Fields:

LINEATTR unsigned int

Attributes of the cache allocated lines.

LINES unsigned int

Number of cache lines already allocated.

NS_ADDR unsigned int

The security world and address of the first byte of the page, the MSB encodes the NS bit.

PAS enum

The Physical Address Space.

TXATTR unsigned int

Attributes of the transaction.

ArchMsg.Error.cached_attributes_mismatch#lineinfo

Information about allocated lines in page. DISPLAY %
{TAG[1]:(u|U))%{TAG[2]:(d|D)}-
{TAG[63:61]:(s|rt|ns|rl|sa|na6|nsp|na7)}-%{PADDR}-%
{TAG[59:58]:(nsh|ish|osh|sys))%{TAG[56]:(|-iHittable))%
{TAG[57]:(|-oHittable)}. Fields:

PADDR unsigned int

address of this cache line.

TAG unsigned int

tag data of this cache line.

ArchMsg.Warning.ns_s_dirty_hit

Cache hit at S and NS versions of the same physical address DISPLAY At PAddr %
{PADDR} Line index %
{INDEX0} is for %
{TAG0[1]:(u|U))%{TAG0[2]:(d|D)}-%
{TAG0[63:61]:(s|rt|ns|rl|sa|na6|nsp|na7)}-%
{PADDR}-%
{TAG0[59:58]:(nsh|ish|osh|sys))%
{TAG0[56]:(|-iHittable))%
{TAG0[57]:(|-oHittable)} Line index %
{INDEX1} is for %
{TAG1[1]:(u|U))%{TAG1[2]:(d|D)}-%
{TAG1[63:61]:(s|rt|ns|rl|sa|na6|nsp|na7)}-%
{PADDR}-%
{TAG1[59:58]:(nsh|ish|osh|sys))%
{TAG1[56]:(|-iHittable))%
{TAG1[57]:(|-oHittable)} i.e. same address but different security regimes and one or more are dirty This is not an error if different memory is backing the lines, but otherwise can cause unpredictable cache incoherency. PRIMARY KEY PADDR. Fields:

INDEX0 unsigned int

The cache line index of the affected cache line.

INDEX1 unsigned int

The cache line index of the other regime cache line.

PADDR unsigned int

The address of the affected cache line.

TAG0 unsigned int

The tag data of the affected cache line.

TAG1 unsigned int

The tag data of the other regime cache line.

CACHE_ACCESS

Number of cache lookups.

CACHE_MISS

cache access cache miss.

CACHE_READ_HIT

Read access cache hit. Note that this trace source causes a large slowdown in the simulation.
Fields:

IS_PRELOAD enum

Is the access a preload.

IS_SHARED enum

Is the access shared.

LATENCY unsigned int

Cache latency in ticks per access.

MANAGER_ID unsigned int

Indicates the manager ID associated with this monitor.

CACHE_READ_MISS

Read access cache miss and cache miss latency. Fields:

IS_PRELOAD enum

Is the access a preload.

IS_SHARED enum

Is the access shared.

LATENCY unsigned int

Cache miss latency in ticks per access.

MANAGER_ID unsigned int

Indicates the manager ID associated with this monitor.

CACHE_SNOOP BROADCAST

cache snoop accesses.

CACHE_WRITE_HIT

Write access cache hit. Note that this trace source causes a large slowdown in the simulation.
Fields:

IS_PRELOAD enum

Is the access a preload.

IS_SHARED enum

Is the access shared.

LATENCY unsigned int

Cache latency in ticks per access.

MANAGER_ID unsigned int

Indicates the manager ID associated with this monitor.

CACHE_WRITE_MISS

Write access cache miss and cache miss latency. Fields:

IS_PRELOAD enum

Is the access a preload.

IS_SHARED enum

Is the access shared.

LATENCY unsigned int

Cache miss latency in ticks per access.

MANAGER_ID unsigned int

Indicates the manager ID associated with this monitor.

DVM_Message

DISPLAY DVM_msg (%{DVM_MESSAGE}). Fields:

DVM_MESSAGE string

The DVM Message.

ENTRY_BECOMES_INVALID

An entry is now invalid. The cause could be any reason. Fields:

ENTRY_INDEX unsigned int

The entry index that has become invalid.

ENTRY_SET_DIRTY

Entry change, clean/dirty. Fields:

ENTRY_INDEX unsigned int

The entry index that has become clean or dirty.

STATUS unsigned int

True if the line was made dirty.

ERROR_MIXED_ATTRIBUTES_LINE

Mismatched line and transaction attributes. Fields:

ENTRY_INDEX unsigned int

The entry index that has collided.

NS_ADDR unsigned int

The security world and address of the first byte of the line, the MSB encodes the NS bit.

PAS enum

The Physical Address Space.

ERROR_MIXED_ATTRIBUTES_PAGE

Attributes differ between page and lines in cache from the same page. Fields:

NS_ADDR unsigned int

The security world and address of the first byte of the page, the MSB encodes the NS bit.

PAS enum

The Physical Address Space.

change_dvm_disabled_ports

This reports a bitmap of which ports are disabled to receive DVM messages. Fields:

new_bitmap unsigned int

The new bitmap of disabled ports.

new_bitmap_list unsigned int

A list of port numbers that were disabled.

old_bitmap unsigned int

The old bitmap of disabled ports.

old_bitmap_list unsigned int

A list of port numbers that were disabled.

change_upstream_disabled_ports

This reports a bitmap of which upstream ports of the cache are *disabled* and so no snoop requests are sent to them. Fields:

new_bitmap unsigned int

The bitmap of the first 32 upstream ports that are disabled.

new_bitmap_list unsigned int

A list of port numbers that were disabled.

old_bitmap unsigned int

The bitmap of the first 32 upstream ports that are disabled.

old_bitmap_list unsigned int

A list of port numbers that were disabled.

entry_after_far_atomic

A cache line has had a far atomic applied to it. Fields:

atomic_operation enum

The operation that operated on this line.

be bool

The far atomic's big-endian flag.

data unsigned int

The data in the cache line.

entry_begin_address unsigned int

The start address of the entry.

entry_index unsigned int

The entry index in the cache.

incoming_data unsigned int

The incoming data from the far atomic. For CAS operations then this is compare-value and then the replacement-value.

manager_id unsigned int

The manager ID for the associated transaction.

ns enum

The security state of the line.

pas enum

The Physical Address Space.

prior_data unsigned int

The data that will be returned as the 'prior' data of the far atomic.

trans_begin_address unsigned int

The start address of the transaction.

entry_after_read

A cache line has been read. Fields:

ace_operation enum

The ACE operation that operated on this line.

data unsigned int

The data in the cache line.

dirty bool

Is the line ACE-dirty.

entry_begin_address unsigned int

The start address of the entry.

entry_index unsigned int

The entry index in the cache.

ns enum

The security state of the line.

pas enum

The Physical Address Space.

snoop bool

This transaction is a snoop transaction.

trans_begin_address unsigned int

The start address of the transaction.

unique bool

Is the line ACE-unique.

entry_after_write

A cache line has been written. Fields:

ace_operation enum

The ACE operation that operated on this line.

data unsigned int

The data in the cache line.

entry_begin_address unsigned int

The start address of the entry.

entry_index unsigned int

The entry index in the cache.

ns enum

The security state of the line.

pas enum

The Physical Address Space.

trans_begin_address unsigned int

The start address of the transaction.

2.80 CCNRegisterSet

This section describes the trace sources.

ArchMsg.Warning.CCNRegister.MemoryMapped_ReadReserved

DISPLAY CCN reserved location attempting to be read. Fields:

OFFSET unsigned int

Offset of address within the CCN module.

ArchMsg.Warning.CCNRegister.MemoryMapped_ReadWriteOnlyReg

DISPLAY CCN write-only register %{REG_NAME} attempting to be read. Fields:

OFFSET unsigned int

Offset of address within the CCN module.

REG_NAME string

Name of the register being read.

ArchMsg.Warning.CCNRegister.MemoryMapped_WriteReadOnlyReg

DISPLAY CCN read-only register %{REG_NAME} attempting to be written with value %{VALUE}. Fields:

OFFSET unsigned int

Offset of address within the CCN module.

REG_NAME string

Name of the register being written.

VALUE unsigned int

Value written.

ArchMsg.Warning.CCNRegister.MemoryMapped_WriteReserved

DISPLAY CCN reserved location attempting to be written with value %{VALUE}. Fields:

OFFSET unsigned int

Offset of address within the CCN module.

VALUE unsigned int

Value written.

CCNRegister.MemoryMapped_Read

Trace read from a CCN memory-mapped register. Fields:

NS bool

Access is non-secure.

OFFSET unsigned int

Offset of address within the CCN module.

REG_NAME string

Name of the register being read.

VALUE unsigned int

Value returned.

CCNRegister.MemoryMapped_Write

Trace write to a CCN memory-mapped register. Fields:

NS bool

Access is non-secure.

OFFSET unsigned int

Offset of address within the CCN module.

REG_NAME string

Name of the register being written.

UPDATED_VALUE unsigned int

New value read back from the register; for write-only registers this will be equal to VALUE.

VALUE unsigned int

Value written.

2.81 CI700

This section describes the trace sources.

ASSIGNED_LCN_LDID

DISPLAY In real HW, access to memory location %{ADDR} routed through HN-S (ID = %{HNSID}) LCN with LDID = %{LCN_LDID}. Fields:

ADDR unsigned int

Address of memory location being accessed.

HNSID unsigned int

NodeID of HNS node transaction is routed to.

LCN_LDID unsigned int

Programmed LDID of LCN node transaction is routed to.

ArchMsg.Error.MemoryMapped_AddressInvalid

DISPLAY Register access attempted to %{ADDRESS} does not fall into %{REG_SPACE_BASE} - %{REG_SPACE_END}. This range is where registers may be found. Fields:

ADDRESS unsigned int

Address of Access that entered RegisterInterface to the Register Space in the Interconnect.

REG_SPACE_BASE unsigned int

Register Space Base Address.

REG_SPACE_END unsigned int

Register Space End Address.

ArchMsg.Warning.MemoryMapped_ReadDisallowed

DISPLAY Read of %{REG_NAME}{%{Reg_Ownership}} not allowed from PASpace= %{PASpace}; will be read-as-zero. Fields:

OFFSET unsigned int

Offset of address within the Interconnect.

PASpace string

Physical Address Space.

REG_NAME string

Name of the register being read.

Reg_Ownership string

Register Ownership.

ArchMsg.Warning.MemoryMapped_ReadReserved

DISPLAY Reserved offset %{OFFSET} attempting to be read. Fields:

OFFSET unsigned int

Offset of address within the Interconnect.

ArchMsg.Warning.MemoryMapped_ReadWriteOnlyReg

DISPLAY Write-only register %{REG_NAME} attempting to be read; will read-as-zero. Fields:

OFFSET unsigned int

Offset of address within the Interconnect.

REG_NAME string

Name of the register being read.

ArchMsg.Warning.MemoryMapped_UnalignedReadAccess

DISPLAY Read attempted to %{OFFSET} and not aligned to 32 or 64-bit. Fields:

OFFSET unsigned int

Offset of address within the Interconnect.

ArchMsg.Warning.MemoryMapped_UnalignedWriteAccess

DISPLAY Write attempted to %{OFFSET} and not aligned to 32 or 64-bit. Fields:

OFFSET unsigned int

Offset of address within the Interconnect.

ArchMsg.Warning.MemoryMapped_WriteDisallowed

DISPLAY Write to %{REG_NAME}(%{Reg_Ownership})=%{VALUE} not allowed from PASpace=%{PASpace}; will be ignored. Fields:

OFFSET unsigned int

Offset of address within the Interconnect.

PASpace string

Physical Address Space.

REG_NAME string

Name of the register being read.

Reg_Ownership string

Register Ownership.

VALUE unsigned int

Value.

ArchMsg.Warning.MemoryMapped_WriteReadOnlyReg

DISPLAY Read-only register %{REG_NAME} attempting to be written with value %{VALUE}. Fields:

OFFSET unsigned int

Offset of address within the Interconnect.

REG_NAME string

Name of the register being written.

VALUE unsigned int

Value written.

ArchMsg.Warning.MemoryMapped_WriteReserved

DISPLAY Reserved offset %{OFFSET} attempting to be written with value %{VALUE}. Fields:

OFFSET unsigned int

Offset of address within the Interconnect.

VALUE unsigned int

Value written.

CMN_AXU_Routing

DISPLAY Access to memory location %{ADDR} is mapped to %{AXUTYPE}_AXU %{AXU_IDX} which is located on %{NODETYPE} %{NODE_IDX}. Fields:

ADDR unsigned int

Address of memory location being accessed.

AXUTYPE string

AXU Port Type (DSU, DMC, etc.).

AXU_IDX unsigned int

Index of the AXU Port.

NODETYPE string

Node type the AXU port is associated with (RNF, SNF, etc.).

NODE_IDX unsigned int

Index of the Node.

CMN_Cache_Downstream_Routing

DISPLAY Access to memory location %{ADDR} is routed to %{TYPE} port %{TO_PORT}. Fields:

ADDR unsigned int

Address of memory location being accessed.

FROM_PORT unsigned int

Upstream port.

TO_PORT unsigned int

Downstream port number to which access is routed.

TYPE string

Downstream port type to which access is routed.

CMN_Cache_Routing

DISPLAY Access to memory location %{ADDR} from port %{FROM_PORT} is routed to System Cache. Fields:

ADDR unsigned int

Address of memory location being accessed.

FROM_PORT unsigned int

Upstream port.

CMN_Invalid_OCM_Region

DISPLAY OCM region %{OCM_REGION_START}-{OCM_REGION_END} does not belong to Cache region %{CACHE_REGION_START}-{CACHE_REGION_END}. Fields:

CACHE_REGION_END unsigned int

End Address of memory location that belongs to the Cache.

CACHE_REGION_START unsigned int

Start Address of memory location that belongs to the Cache.

OCM_REGION_END unsigned int

End Address of OCM region.

OCM_REGION_START unsigned int

Start Address of OCM region.

CMN_OCM_Routing

DISPLAY Access to memory location %{ADDR} is routed to OCM offset %{OFFSET}. Fields:

ADDR unsigned int

Address of memory location being accessed.

OFFSET unsigned int

OFFSET in OCM memory location to which access is routed.

CMN_OCM_Size_Exceeded

DISPLAY Total secure & non-secure accesses exceed the OCM size which is %{OCM_SIZE}. Fields:

OCM_SIZE unsigned int

Size of the OCM.

CMN_OCM_unimplemented

DISPLAY details for unimplemented conditions encountered. Fields:

OCM_UNIMPLEMENTED string

Detail for unimplemented condition reached.

CPAG_ACCESS

DISPLAY In real HW, access to memory location %{ADDR} targets CPAG %{CPAG} and txn will be routed through CCG %{CCG_NODEID}. Fields:

ADDR unsigned int

Address of memory location being accessed.

CCG_NODEID unsigned int

NodeID of the CCG that will be used to route the txn in real HW.

CPAG unsigned int

CPAG number that the txn targets.

HNFSAM_Memory_Map

DISPLAY Memory Map of HNF%{HNF_INDEX} %{MMAP}. Fields:

HNF_INDEX unsigned int

Index of HNF.

MMAP string

memory map of cmn600 hnfsam.

HNF_ACCESS

DISPLAY In real HW, access to memory location %{ADDR} routed through HN-F/HN-S with node ID = %{HNID}. Fields:

ADDR unsigned int

Address of memory location being accessed.

HNFID unsigned int

NodeID of home cache node transaction is routed to.

HNSAM_Target_SNF

DISPLAY The target SNF for address (%{ADDRESS}) is (%{SNF_NUM}). Fields:

ADDRESS unsigned int

Transaction Address.

SNF_NUM unsigned int

SNF Node Index which is targeted from HNFSAM.

Interconnect_ABF_policy_reg

DISPLAY ABF policy register write details. Fields:

ABF_PR_WRITE string

Detail for policy register write.

Interconnect_do_abf_call

DISPLAY Cache state modelled %{CACHE_STATE_MODELLED} ABF commanded for addr %{ABF_REGION_START}-{ABF_REGION_END} with mode %{ABF_MODE} and . Fields:

ABF_MODE unsigned int

0 = CLEAN_INVALID, 1 = MAKE_INVALID, 2 = CLEAN_SHARED, 3 = RESERVED.

ABF_REGION_END unsigned int

End Address of ABF region.

ABF_REGION_START unsigned int

Start Address of ABF region.

CACHE_STATE_MODELLED bool

cache_state_modelled, true = yes, false = no.

LCN_ACCESS

DISPLAY In real HW, access to memory location %{ADDR} routed through LCN with node ID = %{LCN_NODEID}. Fields:

ADDR unsigned int

Address of memory location being accessed.

LCN_NODEID unsigned int

Node Id of LCN node transaction is routed to.

MTU_address_generation

DISPLAY MTU Address generation: Mode %{MODE}, original address %{ORIGINAL}, translated: %{TRANSLATION}, and shuttered: %{SHUTTER}. Fields:

MODE enum

Translation mode configured.

ORIGINAL unsigned int

Original transaction address.

SHUTTER unsigned int

Address after shuttering.

TRANSLATION unsigned int

Address after translation.

MTU_shuttering_offset

DISPLAY Shuttering bit %{INDEX} by %{OFFSET}. Fields:

INDEX unsigned int

Address bit index.

OFFSET unsigned int

Shifted by this amount.

MTU_translation_mode

DISPLAY MTU Address generation mode set to %{MODE}. Fields:

MODE enum

Translation mode being configured.

MemoryMapped_Read

Trace read from a memory-mapped register. Fields:

OFFSET unsigned int

Offset of address within the Interconnect.

PASpace string

Physical Address Space.

REG_NAME string

Name of the register being read.

VALUE unsigned int

Value returned.

MemoryMapped_Write

Trace write to a memory-mapped register. Fields:

OFFSET unsigned int

Offset of address within the Interconnect.

PASpace string

Physical Address Space.

REG_NAME string

Name of the register being written.

UPDATED_VALUE unsigned int

New value read back from the register; for write-only registers this will be equal to VALUE.

VALUE unsigned int

Value written.

OCM_Regions

DISPLAY Regions configured as OCM %{REGIONS}. Fields:

REGIONS string

memory regions allocated to OCM.

RNSAM_Memory_Map

DISPLAY Memory Map of RNSAM %{MMAP}. Fields:

MMAP string

memory map of cmn600 rnsam.

RNSAM_Target_HNF

DISPLAY The target HNF for address (%{ADDRESS}) is (%{HNF_NUM}). Fields:

ADDRESS unsigned int

Transaction Address.

HNF_NUM unsigned int

HNF Node Index to which the transaction is routed.

SNF_HASHING_Target_SNF

DISPLAY The target SNF for address (%{ADDRESS}) in default hashed region is (%{HASHED_SNF}). But the model is routing the transaction to SNF indicated by the HNSAM_Target_SNF trace. Fields:

ADDRESS unsigned int

Transaction Address.

SNF_NUM unsigned int

SNF Node Index which is targeted from HNSAM hashing for default region.

Warning.Hashing_enabled_for_cache_group

DISPLAY The SCG group (which contains HNF%{INDEX}) has different SNF default ports in member HNFs which is not supported. Fields:

HNF_INDEX unsigned int

HNF Node index which is part of system cache group.

Warning.MTU_MODE_Not_Unique

DISPLAY MTU node configuration should be same for all devices. The MTU node %{INDEX} in mode %{MODE} is different the current mode %{CURRENT_MODE}. Fields:

CURRENT_MODE enum

Execution mode for another device already configured.

INDEX unsigned int

MTU index being configured.

MODE enum

Execution mode being configured.

Warning.MTU_Shuttering_Reg_Not_Unique

DISPLAY MTU shuttering register value should be same for all devices. Register %{REG_INDEX} in MTU %{MTU_INDEX} with value %{VALUE} is different the current value %{CURRENT_VALUE}. Fields:

CURRENT_VALUE unsigned int

Shuttering register index already configured.

MTU_INDEX unsigned int

MTU index being configured.

REG_INDEX unsigned int

Register index being configured.

VALUE unsigned int

Shuttering register index being configured.

Warning.RNSAM_Not_Unique

DISPLAY RNSAM configuration should be same for all masters. The RNSAM for master %{INDEX} is different. Fields:

INDEX unsigned int

Index of RN component. Index starts with RNFs and then the RNIs.

Warning.event_cpu_events_disabled

DISPLAY Event not propagated because CPU Events are disabled for CCIX link%{INDEX}.
Fields:

INDEX unsigned int
CCIX link index.

Warning.event_smp_mode_disabled

DISPLAY Event not propagated because SMP mode is not enabled.

Warning.rnsam_and_cxrasam_mismatch

DISPLAY RNSAM and CXRASAM address regions don't match. You may need to review its configuration.

event_propagation_downstream

DISPLAY Event changed from downstream (from other chips) from port: %{FROM_PORT} to state: %{STATE}. Fields:

FROM_PORT unsigned int
Activated port.

STATE unsigned int
State being transmitted.

event_propagation_downstream_routing

DISPLAY Event propagated from downstream (from other systems) from port: %{FROM_PORT} to link: %{LINK} state: %{STATE} sent to local clusters. Fields:

FROM_PORT unsigned int
Activated port.

LINK unsigned int
Link.

STATE unsigned int
State being transmitted.

event_propagation_upstream

DISPLAY Event changed from upstream (clusters) sent to another chip via port %{TO_PORT} on link %{LINK} state: %{STATE}. Fields:

LINK unsigned int
Link number.

STATE unsigned int
State being transmitted.

TO_PORT unsigned int
Target port.

trace_add_cxra_memory_region

DISPLAY CXRA%{INDEX} SAM memory region: [%{START}-%{END}] => HAID=%{HAID}.

Fields:

END unsigned int

End address of the memory region.

HAID unsigned int

Memory region target HAID.

INDEX unsigned int

CXRA/CCG_RA index.

START unsigned int

Start address of the memory region.

trace_add_memory_region

DISPLAY Add new memory region type %{TYPE}, range start : %{START}, and end: %{END}.

Fields:

END unsigned int

End address of the memory region.

START unsigned int

Start address of the memory region.

TYPE enum

Node type.

trace_hnd

DISPLAY The HND index for XP%{INDEX} is %{HND_INDEX}. Fields:

HND_INDEX unsigned int

HND index for the XP.

INDEX unsigned int

XP Index.

2.82 CMN600

This section describes the trace sources.

ASSIGNED_LCN_LDID

DISPLAY In real HW, access to memory location %{ADDR} routed through HN-S (ID = %{HNSID}) LCN with LDID = %{LCN_LDID}. Fields:

ADDR unsigned int

Address of memory location being accessed.

HNSID unsigned int

NodeID of HNS node transaction is routed to.

LCN_LDID unsigned int

Programmed LDID of LCN node transaction is routed to.

ArchMsg.Error.MemoryMapped_AddressInvalid

DISPLAY Register access attempted to %{ADDRESS} does not fall into %{REG_SPACE_BASE} - %{REG_SPACE_END}. This range is where registers may be found. Fields:

ADDRESS unsigned int

Address of Access that entered RegisterInterface to the Register Space in the Interconnect.

REG_SPACE_BASE unsigned int

Register Space Base Address.

REG_SPACE_END unsigned int

Register Space End Address.

ArchMsg.Warning.MemoryMapped_ReadDisallowed

DISPLAY Read of %{REG_NAME}(%{Reg_Ownership}) not allowed from PASpace= %{PASpace}; will be read-as-zero. Fields:

OFFSET unsigned int

Offset of address within the Interconnect.

PASpace string

Physical Address Space.

REG_NAME string

Name of the register being read.

Reg_Ownership string

Register Ownership.

ArchMsg.Warning.MemoryMapped_ReadReserved

DISPLAY Reserved offset %{OFFSET} attempting to be read. Fields:

OFFSET unsigned int

Offset of address within the Interconnect.

ArchMsg.Warning.MemoryMapped_ReadWriteOnlyReg

DISPLAY Write-only register %{REG_NAME} attempting to be read; will read-as-zero. Fields:

OFFSET unsigned int

Offset of address within the Interconnect.

REG_NAME string

Name of the register being read.

ArchMsg.Warning.MemoryMapped_UnalignedReadAccess

DISPLAY Read attempted to %{OFFSET} and not aligned to 32 or 64-bit. Fields:

OFFSET unsigned int

Offset of address within the Interconnect.

ArchMsg.Warning.MemoryMapped_UnalignedWriteAccess

DISPLAY Write attempted to %{OFFSET} and not aligned to 32 or 64-bit. Fields:

OFFSET unsigned int

Offset of address within the Interconnect.

ArchMsg.Warning.MemoryMapped_WriteDisallowed

DISPLAY Write to %{REG_NAME}{%{Reg_Ownership}}=%{VALUE} not allowed from PASpace=%{PASpace}; will be ignored. Fields:

OFFSET unsigned int

Offset of address within the Interconnect.

PASpace string

Physical Address Space.

REG_NAME string

Name of the register being read.

Reg_Ownership string

Register Ownership.

VALUE unsigned int

Value.

ArchMsg.Warning.MemoryMapped_WriteReadOnlyReg

DISPLAY Read-only register %{REG_NAME} attempting to be written with value %{VALUE}. Fields:

OFFSET unsigned int

Offset of address within the Interconnect.

REG_NAME string

Name of the register being written.

VALUE unsigned int

Value written.

ArchMsg.Warning.MemoryMapped_WriteReserved

DISPLAY Reserved offset %{OFFSET} attempting to be written with value %{VALUE}. Fields:

OFFSET unsigned int

Offset of address within the Interconnect.

VALUE unsigned int

Value written.

CMN_AXU_Routing

DISPLAY Access to memory location %{ADDR} is mapped to %{AXUTYPE}_AXU %{AXU_IDX} which is located on %{NODETYPE} %{NODE_IDX}. Fields:

ADDR unsigned int

Address of memory location being accessed.

AXUTYPE string

AXU Port Type (DSU, DMC, etc.).

AXU_IDX unsigned int

Index of the AXU Port.

NODETYPE string

Node type the AXU port is associated with (RNF, SNF, etc.).

NODE_IDX unsigned int

Index of the Node.

CMN_Cache_Downstream_Routing

DISPLAY Access to memory location %{ADDR} is routed to %{TYPE} port %{TO_PORT}. Fields:

ADDR unsigned int

Address of memory location being accessed.

FROM_PORT unsigned int

Upstream port.

TO_PORT unsigned int

Downstream port number to which access is routed.

TYPE string

Downstream port type to which access is routed.

CMN_Cache_Routing

DISPLAY Access to memory location %{ADDR} from port %{FROM_PORT} is routed to System Cache. Fields:

ADDR unsigned int

Address of memory location being accessed.

FROM_PORT unsigned int

Upstream port.

CMN_Invalid_OCM_Region

DISPLAY OCM region %{OCM_REGION_START}-{OCM_REGION_END} does not belong to Cache region %{CACHE_REGION_START}-{CACHE_REGION_END}. Fields:

CACHE_REGION_END unsigned int

End Address of memory location that belongs to the Cache.

CACHE_REGION_START unsigned int

Start Address of memory location that belongs to the Cache.

OCM_REGION_END unsigned int

End Address of OCM region.

OCM_REGION_START unsigned int

Start Address of OCM region.

CMN_OCM_Routing

DISPLAY Access to memory location %{ADDR} is routed to OCM offset %{OFFSET}. Fields:

ADDR unsigned int

Address of memory location being accessed.

OFFSET unsigned int

OFFSET in OCM memory location to which access is routed.

CMN_OCM_Size_Exceeded

DISPLAY Total secure & non-secure accesses exceed the OCM size which is %{OCM_SIZE}. Fields:

OCM_SIZE unsigned int

Size of the OCM.

CMN_OCM_unimplemented

DISPLAY details for unimplemented conditions encountered. Fields:

OCM_UNIMPLEMENTED string

Detail for unimplemented condition reached.

CPAG_ACCESS

DISPLAY In real HW, access to memory location %{ADDR} targets CPAG %{CPAG} and txn will be routed through CCG %{CCG_NODEID}. Fields:

ADDR unsigned int

Address of memory location being accessed.

CCG_NODEID unsigned int

NodeID of the CCG that will be used to route the txn in real HW.

CPAG unsigned int

CPAG number that the txn targets.

HNFSAM_Memory_Map

DISPLAY Memory Map of HNF %{HNF_INDEX} %{MMAP}. Fields:

HNF_INDEX unsigned int

Index of HNF.

MMAP string

memory map of cmn600 hnfsam.

HNH_ACCESS

DISPLAY In real HW, access to memory location %{ADDR} routed through HN-F/HN-S with node ID = %{HNID}. Fields:

ADDR unsigned int

Address of memory location being accessed.

HNHID unsigned int

Nodeid of home cache node transaction is routed to.

HNSAM_Target_SNF

DISPLAY The target SNF for address (%{ADDRESS}) is (%{SNF_NUM}). Fields:

ADDRESS unsigned int

Transaction Address.

SNF_NUM unsigned int

SNF Node Index which is targeted from HNSAM.

Interconnect_ABFPOLICY_REG

DISPLAY ABF policy register write details. Fields:

ABF_PR_WRITE string

Detail for policy register write.

Interconnect_do_abf_call

DISPLAY Cache state modelled %{CACHE_STATE_MODELLED} ABF commanded for address %{ABF_REGION_START}-%{ABF_REGION_END} with mode %{ABF_MODE} and . Fields:

ABF_MODE unsigned int

0 = CLEAN_INVALID, 1 = MAKE_INVALID, 2 = CLEAN_SHARED, 3 = RESERVED.

ABF_REGION_END unsigned int

End Address of ABF region.

ABF_REGION_START unsigned int

Start Address of ABF region.

CACHE_STATE_MODELLED bool

cache_state_modelled, true = yes, false = no.

LCN_ACCESS

DISPLAY In real HW, access to memory location %{ADDR} routed through LCN with node ID = %{LCN_NODEID}. Fields:

ADDR unsigned int

Address of memory location being accessed.

LCN_NODEID unsigned int

Node Id of LCN node transaction is routed to.

MemoryMapped_Read

Trace read from a memory-mapped register. Fields:

OFFSET unsigned int

Offset of address within the Interconnect.

PASpace string

Physical Address Space.

REG_NAME string

Name of the register being read.

VALUE unsigned int

Value returned.

MemoryMapped_Write

Trace write to a memory-mapped register. Fields:

OFFSET unsigned int

Offset of address within the Interconnect.

PASpace string

Physical Address Space.

REG_NAME string

Name of the register being written.

UPDATED_VALUE unsigned int

New value read back from the register; for write-only registers this will be equal to VALUE.

VALUE unsigned int

Value written.

OCM_Regions

DISPLAY Regions configured as OCM %{REGIONS}. Fields:

REGIONS string

memory regions allocated to OCM.

RNSAM_Memory_Map

DISPLAY Memory Map of RNSAM %{MMAP}. Fields:

MMAP string

memory map of cmn600 rnsam.

RNSAM_Target_HNF

DISPLAY The target HNF for address (%{ADDRESS}) is (%{HNF_NUM}). Fields:

ADDRESS unsigned int

Transaction Address.

HNF_NUM unsigned int

HNF Node Index to which the transaction is routed.

SNF_HASHING_Target_SNF

DISPLAY The target SNF for address (%{ADDRESS}) in default hashed region is (%{HASHED_SNF}). But the model is routing the transaction to SNF indicated by the HNSAM_Target_SNF trace. Fields:

ADDRESS unsigned int

Transaction Address.

SNF_NUM unsigned int

SNF Node Index which is targeted from HNSAM hashing for default region.

Warning.Hashing_enabled_for_cache_group

DISPLAY The SCG group (which contains HNF%{INDEX}) has different SNF default ports in member HNFs which is not supported. Fields:

HNF_INDEX unsigned int

HNF Node index which is part of system cache group.

Warning.RNSAM_Not_Unique

DISPLAY RNSAM configuration should be same for all masters. The RNSAM for master %{INDEX} is different. Fields:

INDEX unsigned int

Index of RN component. Index starts with RNFs and then the RNIs.

Warning.event_cpu_events_disabled

DISPLAY Event not propagated because CPU Events are disabled for CCIX link%{INDEX}. Fields:

INDEX unsigned int

CCIX link index.

Warning.event_smp_mode_disabled

DISPLAY Event not propagated because SMP mode is not enabled.

Warning.rnsam_and_cxrasam_mismatch

DISPLAY RNSAM and CXRASAM address regions don't match. You may need to review its configuration.

event_propagation_downstream

DISPLAY Event changed from downstream (from other chips) from port: %{FROM_PORT} to state: %{STATE}. Fields:

FROM_PORT unsigned int

Activated port.

STATE unsigned int

State being transmitted.

event_propagation_downstream_routing

DISPLAY Event propagated from downstream (from other systems) from port: %{FROM_PORT} to link: %{LINK} state: %{STATE} sent to local clusters. Fields:

FROM_PORT unsigned int

Activated port.

LINK unsigned int

Link.

STATE unsigned int

State being transmitted.

event_propagation_upstream

DISPLAY Event changed from upstream (clusters) sent to another chip via port %{TO_PORT} on link %{LINK} state: %{STATE}. Fields:

LINK unsigned int

Link number.

STATE unsigned int

State being transmitted.

TO_PORT unsigned int

Target port.

trace_add_cxra_memory_region

DISPLAY CXRA%{INDEX} SAM memory region: [%{START}-%{END}] => HAID=%{HAID}. Fields:

END unsigned int

End address of the memory region.

HAID unsigned int

Memory region target HAID.

INDEX unsigned int

CXRA/CCG_RA index.

START unsigned int

Start address of the memory region.

trace_add_memory_region

DISPLAY Add new memory region type %{TYPE}, range start : %{START}, and end: %{END}. Fields:

END unsigned int

End address of the memory region.

START unsigned int

Start address of the memory region.

TYPE enum

Node type.

trace_hnd

DISPLAY The HND index for XP%{INDEX} is %HND_INDEX}. Fields:

HND_INDEX unsigned int

HND index for the XP.

INDEX unsigned int

XP Index.

2.83 CMN600AE

This section describes the trace sources.

ASSIGNED_LCN_LDID

DISPLAY In real HW, access to memory location %ADDR} routed through HN-S (ID = %HNSID}) LCN with LDID = %LCN_LDID}. Fields:

ADDR unsigned int

Address of memory location being accessed.

HNSID unsigned int

NodeID of HNS node transaction is routed to.

LCN_LDID unsigned int

Programmed LDID of LCN node transaction is routed to.

ArchMsg.Error.MemoryMapped_AddressInvalid

DISPLAY Register access attempted to %ADDRESS} does not fall into %REG_SPACE_BASE} - %REG_SPACE_END}. This range is where registers may be found. Fields:

ADDRESS unsigned int

Address of Access that entered RegisterInterface to the Register Space in the Interconnect.

REG_SPACE_BASE unsigned int

Register Space Base Address.

REG_SPACE_END unsigned int

Register Space End Address.

ArchMsg.Warning.MemoryMapped_NotAllowed

Trace when a memory-mapped register access is not allowed. This trace doesn't cover previous traces. Fields:

ERROR_MSG string

Description of the error.

IS_READ bool

Whether it is a read or write access.

OFFSET unsigned int

Offset of address within the CMN600.

REG_NAME string

Name of the register being written.

ArchMsg.Warning.MemoryMapped_ReadDisallowed

DISPLAY Read of %{REG_NAME}{%{Reg_Ownership}} not allowed from PASpace=%{PASpace}; will be read-as-zero. Fields:

OFFSET unsigned int

Offset of address within the Interconnect.

PASpace string

Physical Address Space.

REG_NAME string

Name of the register being read.

Reg_Ownership string

Register Ownership.

ArchMsg.Warning.MemoryMapped_ReadReserved

DISPLAY Reserved offset %{OFFSET} attempting to be read. Fields:

OFFSET unsigned int

Offset of address within the Interconnect.

ArchMsg.Warning.MemoryMapped_ReadWriteOnlyReg

DISPLAY Write-only register %{REG_NAME} attempting to be read; will read-as-zero. Fields:

OFFSET unsigned int

Offset of address within the Interconnect.

REG_NAME string

Name of the register being read.

ArchMsg.Warning.MemoryMapped_UnalignedReadAccess

DISPLAY Read attempted to %{OFFSET} and not aligned to 32 or 64-bit. Fields:

OFFSET unsigned int

Offset of address within the Interconnect.

ArchMsg.Warning.MemoryMapped_UnalignedWriteAccess

DISPLAY Write attempted to %{OFFSET} and not aligned to 32 or 64-bit. Fields:

OFFSET unsigned int

Offset of address within the Interconnect.

ArchMsg.Warning.MemoryMapped_WriteDisallowed

DISPLAY Write to %{REG_NAME}{%{Reg_Ownership}}=%{VALUE} not allowed from PASpace=%{PASpace}; will be ignored. Fields:

OFFSET unsigned int

Offset of address within the Interconnect.

PASpace string

Physical Address Space.

REG_NAME string

Name of the register being read.

Reg_Ownership string

Register Ownership.

VALUE unsigned int

Value.

ArchMsg.Warning.MemoryMapped_WriteReadOnlyReg

DISPLAY Read-only register %{REG_NAME} attempting to be written with value %{VALUE}. Fields:

OFFSET unsigned int

Offset of address within the Interconnect.

REG_NAME string

Name of the register being written.

VALUE unsigned int

Value written.

ArchMsg.Warning.MemoryMapped_WriteReserved

DISPLAY Reserved offset %{OFFSET} attempting to be written with value %{VALUE}. Fields:

OFFSET unsigned int

Offset of address within the Interconnect.

VALUE unsigned int

Value written.

CMN_AXU_Routing

DISPLAY Access to memory location %{ADDR} is mapped to %{AXUTYPE}_AXU %{AXU_IDX} which is located on %{NODETYPE} %{NODE_IDX}. Fields:

ADDR unsigned int

Address of memory location being accessed.

AXUTYPE string

AXU Port Type (DSU, DMC, etc.).

AXU_IDX unsigned int

Index of the AXU Port.

NODETYPE string

Node type the AXU port is associated with (RNF, SNF, etc.).

NODE_IDX unsigned int

Index of the Node.

CMN_Cache_Downstream_Routing

DISPLAY Access to memory location %{ADDR} is routed to %{TYPE} port %{TO_PORT}. Fields:

ADDR unsigned int

Address of memory location being accessed.

FROM_PORT unsigned int

Upstream port.

TO_PORT unsigned int

Downstream port number to which access is routed.

TYPE string

Downstream port type to which access is routed.

CMN_Cache_Routing

DISPLAY Access to memory location %{ADDR} from port %{FROM_PORT} is routed to System Cache. Fields:

ADDR unsigned int

Address of memory location being accessed.

FROM_PORT unsigned int

Upstream port.

CMN_Invalid_OCM_Region

DISPLAY OCM region %{OCM_REGION_START}-{OCM_REGION_END} does not belong to Cache region %{CACHE_REGION_START}-{CACHE_REGION_END}. Fields:

CACHE_REGION_END unsigned int

End Address of memory location that belongs to the Cache.

CACHE_REGION_START unsigned int

Start Address of memory location that belongs to the Cache.

OCM_REGION_END unsigned int

End Address of OCM region.

OCM_REGION_START unsigned int

Start Address of OCM region.

CMN_OCM_Routing

DISPLAY Access to memory location %{ADDR} is routed to OCM offset %{OFFSET}. Fields:

ADDR unsigned int

Address of memory location being accessed.

OFFSET unsigned int

OFFSET in OCM memory location to which access is routed.

CMN_OCM_Size_Exceeded

DISPLAY Total secure & non-secure accesses exceed the OCM size which is %{OCM_SIZE}. Fields:

OCM_SIZE unsigned int

Size of the OCM.

CMN_OCM_unimplemented

DISPLAY details for unimplemented conditions encountered. Fields:

OCM_UNIMPLEMENTED string

Detail for unimplemented condition reached.

CPAG_ACCESS

DISPLAY In real HW, access to memory location %{ADDR} targets CPAG %{CPAG} and txn will be routed through CCG %{CCG_NODEID}. Fields:

ADDR unsigned int

Address of memory location being accessed.

CCG_NODEID unsigned int

NodeID of the CCG that will be used to route the txn in real HW.

CPAG unsigned int

CPAG number that the txn targets.

HNFSAM_Memory_Map

DISPLAY Memory Map of HNF%{HNF_INDEX} %{MMAP}. Fields:

HNF_INDEX unsigned int

Index of HNF.

MMAP string

memory map of cmn600 hnfsam.

HNFS_ACCESS

DISPLAY In real HW, access to memory location %{ADDR} routed through HN-F/HN-S with node ID = %{HNID}. Fields:

ADDR unsigned int

Address of memory location being accessed.

HNFSID unsigned int

NodeID of home cache node transaction is routed to.

HNSAM_Target_SNF

DISPLAY The target SNF for address (%{ADDRESS}) is (%{SNF_NUM}). Fields:

ADDRESS unsigned int

Transaction Address.

SNF_NUM unsigned int

SNF Node Index which is targeted from HNFSAM.

Interconnect_ABF_policy_reg

DISPLAY ABF policy register write details. Fields:

ABF_PR_WRITE string

Detail for policy register write.

Interconnect_do_abf_call

DISPLAY Cache state modelled %{CACHE_STATE_MODELLED} ABF commanded for addr %{ABF_REGION_START}-{ABF_REGION_END} with mode %{ABF_MODE} and . Fields:

ABF_MODE unsigned int

0 = CLEAN_INVALID, 1 = MAKE_INVALID, 2 = CLEAN_SHARED, 3 = RESERVED.

ABF_REGION_END unsigned int

End Address of ABF region.

ABF_REGION_START unsigned int

Start Address of ABF region.

CACHE_STATE_MODELLED bool

cache_state_modelled, true = yes, false = no.

Invalid_parameter_warning

Traces an invalid user parameter. Fields:

MESSAGE string

Describes the situation.

NAME string

User parameter name.

LCN_ACCESS

DISPLAY In real HW, access to memory location %{ADDR} routed through LCN with node ID = %{LCN_NODEID}}. Fields:

ADDR unsigned int

Address of memory location being accessed.

LCN_NODEID unsigned int

Node Id of LCN node transation is routed to.

MPU_access_allowed_warning

Traces an access allowed by MPU without checking the region configuration. Fields:

BASE_ADDR unsigned int

Base address of the region.

INTERFACE string

Bus interface associated with the transaction.

IS_READ bool

Access type. True: read, False: write.

LIMIT_ADDR unsigned int

Limit address of the region.

MASTER unsigned int

Master associated with the transaction. Possible values are 0 to 7.

MASTER_ENABLED bool

Is master enabled?.

MESSAGE string

Describes the situation.

MPU unsigned int

MPU logical index.

NS_READ bool

Non-secure read permission.

NS_WRITE bool

Non-Secure write permission.

NUMBER_PROGRAMMABLE_REGIONS unsigned int

Number of programmable regions, set by user parameter.

OFFSET unsigned int

Offset of allowed address access within the CMN600.

REGION_ENABLED bool

Is region enabled?.

REGION_NUMBER unsigned int

Region number.

S_READ bool

Secure read permission.

S_WRITE bool

Secure write permission.

MPU_access_denied

Traces an access denied by MPU. Fields:

ACTION string

Action on error.

BASE_ADDR unsigned int

Base address of the region.

INTERFACE string

Bus interface associated with the transaction.

IS_READ bool

Access type. True: read, False: write.

LIMIT_ADDR unsigned int

Limit address of the region.

MASTER unsigned int

Master associated with the transaction. Possible values are 0 to 7.

MASTER_ENABLED bool

Is master enabled?.

MPU unsigned int

MPU logical index.

NS_READ bool

Non-secure read permission.

NS_WRITE bool

Non-Secure write permission.

NUMBER_PROGRAMMABLE_REGIONS unsigned int

Number of programmable regions, set by user parameter.

OFFSET unsigned int

Offset of denied address access within the CMN600.

REGION_ENABLED bool

Is region enabled?.

REGION_NUMBER unsigned int

Region number.

S_READ bool

Secure read permission.

S_WRITE bool

Secure write permission.

MPU_configuration_error

Traces error while configuring MPU. Fields:

ACTION string

Action on error.

BASE_ADDR unsigned int

Base address of the region.

ERROR_MESSAGE string

Error description.

LIMIT_ADDR unsigned int

Limit address of the region.

MASTER_REGISTER unsigned int

Master registers being configured, possible values are 0 to 7.

MPU unsigned int

MPU logical index.

NS_READ bool

Non-secure read permission.

NS_WRITE bool

Non-Secure write permission.

REGION_NUMBER unsigned int

Region number.

S_READ bool

Secure read permission.

S_WRITE bool

Secure write permission.

MemoryMapped_Read

Trace read from a memory-mapped register. Fields:

OFFSET unsigned int

Offset of address within the Interconnect.

PASpace string

Physical Address Space.

REG_NAME string

Name of the register being read.

VALUE unsigned int

Value returned.

MemoryMapped_Write

Trace write to a memory-mapped register. Fields:

OFFSET unsigned int

Offset of address within the Interconnect.

PASpace string

Physical Address Space.

REG_NAME string

Name of the register being written.

UPDATED_VALUE unsigned int

New value read back from the register; for write-only registers this will be equal to VALUE.

VALUE unsigned int

Value written.

OCM_Regions

DISPLAY Regions configured as OCM %{REGIONS}. Fields:

REGIONS string

memory regions allocated to OCM.

RNSAM_Memory_Map

DISPLAY Memory Map of RNSAM %{MMAP}. Fields:

MMAP string

memory map of cmn600 rnsam.

RNSAM_Target_HNF

DISPLAY The target HNF for address (%{ADDRESS}) is (%{HNF_NUM}). Fields:

ADDRESS unsigned int

Transaction Address.

HNF_NUM unsigned int

HNF Node Index to which the transaction is routed.

SNF_HASHING_Target_SNF

DISPLAY The target SNF for address (%{ADDRESS}) in default hashed region is (%{HASHED_SNF}). But the model is routing the transaction to SNF indicated by the HNSAM_Target_SNF trace. Fields:

ADDRESS unsigned int

Transaction Address.

SNF_NUM unsigned int

SNF Node Index which is targeted from HNSAM hashing for default region.

Warning.Hashing_enabled_for_cache_group

DISPLAY The SCG group (which contains HNF%{INDEX}) has different SNF default ports in member HNFs which is not supported. Fields:

HNF_INDEX unsigned int

HNF Node index which is part of system cache group.

Warning.RNSAM_Not_Unique

DISPLAY RNSAM configuration should be same for all masters. The RNSAM for master %{INDEX} is different. Fields:

INDEX unsigned int

Index of RN component. Index starts with RNFs and then the RNIs.

Warning.event_cpu_events_disabled

DISPLAY Event not propagated because CPU Events are disabled for CCIX link%{INDEX}.
Fields:

INDEX unsigned int
CCIX link index.

Warning.event_smp_mode_disabled

DISPLAY Event not propagated because SMP mode is not enabled.

Warning.rnsam_and_cxrasam_mismatch

DISPLAY RNSAM and CXRASAM address regions don't match. You may need to review its configuration.

event_propagation_downstream

DISPLAY Event changed from downstream (from other chips) from port: %{FROM_PORT} to state: %{STATE}. Fields:

FROM_PORT unsigned int
Activated port.

STATE unsigned int
State being transmitted.

event_propagation_downstream_routing

DISPLAY Event propagated from downstream (from other systems) from port: %{FROM_PORT} to link: %{LINK} state: %{STATE} sent to local clusters. Fields:

FROM_PORT unsigned int
Activated port.

LINK unsigned int
Link.

STATE unsigned int
State being transmitted.

event_propagation_upstream

DISPLAY Event changed from upstream (clusters) sent to another chip via port %{TO_PORT} on link %{LINK} state: %{STATE}. Fields:

LINK unsigned int
Link number.

STATE unsigned int
State being transmitted.

TO_PORT unsigned int
Target port.

trace_add_cxra_memory_region

DISPLAY CXRA%{INDEX} SAM memory region: [%{START}-%{END}] => HAID=%{HAID}.

Fields:

END unsigned int

End address of the memory region.

HAID unsigned int

Memory region target HAID.

INDEX unsigned int

CXRA/CCG_RA index.

START unsigned int

Start address of the memory region.

trace_add_memory_region

DISPLAY Add new memory region type %{TYPE}, range start : %{START}, and end: %{END}.

Fields:

END unsigned int

End address of the memory region.

START unsigned int

Start address of the memory region.

TYPE enum

Node type.

trace_hnd

DISPLAY The HND index for XP%{INDEX} is %{HND_INDEX}. Fields:

HND_INDEX unsigned int

HND index for the XP.

INDEX unsigned int

XP Index.

2.84 CMN600CML

This section describes the trace sources.

cross_chip_addressing_inconsistency

DISPLAY Cross-chip addressing inconsistency: [%{START1}-%{END1}] => HAID=%{HAID1} (port=%{PORT1}) [%{START2}-%{END2}] => HAID=%{HAID2}(port=%{PORT2}). Fields:

END1 unsigned int

End address of the memory region 1.

END2 unsigned int

End address of the memory region 2.

HAID1 unsigned int

Target HAID 1.

HAID2 unsigned int

Target HAID 2.

PORT1 unsigned int

Target port 1.

PORT2 unsigned int

Target port 2.

START1 unsigned int

Start address of the memory region 1.

START2 unsigned int

Start address of the memory region 2.

event_propagation_hub_routing

DISPLAY Event broadcast to all clients. Initiated by client: %{INITIATING_CLIENT} state: %{STATE}. Fields:

INITIATING_CLIENT unsigned int

Event initiator.

STATE enum

State being transmitted.

trace_add_port_address_map

DISPLAY Map new memory region for HAID %{HAID} to target port %{TARGET}, with range starting at %{START}, and ending at %{END}. Fields:

END unsigned int

End address of the memory region.

HAID unsigned int

HAID.

START unsigned int

Start address of the memory region.

TARGET_PORT unsigned int

Target port.

trace_address_decode

DISPLAY Initiator port %{CLIENT} at address %{ADDRESS} mapped to port %{TARGET}. If target port %{TARGET} invalid (0xFF) the transaction is remapped to port %{CLIENT}. Fields:

ADDRESS unsigned int

Target Address.

CLIENT unsigned int

Client port.

TARGET unsigned int

Target port.

2.85 CMN600Cache

This section describes the trace sources.

ALLOC_LINEFILL

The system allocated a complete line fill into the RAMs. This happens when the cache has to read the data for a line fill from potentially upstream or downstream. Fields:

ENTRY_INDEX unsigned int

The entry index the line has been loaded into.

MANAGER_ID unsigned int

The manager ID for the associated transaction.

MEMORY_ATTRIBUTES unsigned int

The memory attributes used for the allocation. bits[3:0] are the inner ACACHE, bits[7:4] are the outer ACACHE, bits[9:8] are ADOMAIN, bits[12:10] are APROT, the rest are undocumented.

NS_ADDR unsigned int

The security world and address of the first byte of the line, the MSB encodes the NS bit.

PAS enum

The Physical Address Space.

ALLOC_WRITE

The system allocated a complete line write into the RAMs. This happens when the cache receives a complete cache line write. It need not read from anywhere. Fields:

ENTRY_INDEX unsigned int

The entry index the line has been written into.

MEMORY_ATTRIBUTES unsigned int

The memory attributes used for the allocation. bits[3:0] are the inner ACACHE, bits[7:4] are the outer ACACHE, bits[9:8] are ADOMAIN, bits[12:10] are APROT, the rest are undocumented.

NS_ADDR unsigned int

The security world and address of the first byte of the line, the MSB encodes the NS bit.

PAS enum

The Physical Address Space.

ATOMIC_SLAVE_ACCESS

Information on the atomic operation at slave side. Fields:

WT-cacheable|WB-cacheable))%{TXATTR[7:4.14:13]:(nGnRnE|nGRnE?|GnRnE?|GRnE?
nGnRE|nGRE|GnRE?|GRE))%{TXATTR[16]:(|-transient)} shareability: %
{TXATTR[9:8]:(nsh|ish|osh|sys))%{TXATTR[21.24]:(|TranslatedAccess|NonStallable|TransFaultFlow)%
{TXATTR[22]:(|NonAddressBasedRouted))%{TXATTR[25]:(|HasMetaData)%
{TXATTR[26]:(|ReservedImpDef))%{TXATTR[27]:(|-StreamTransaction)%
{TXATTR[28]:(|-nse))%{TXATTR[29]:(|-nse2))%{TXATTR[31]:(|-HasMutatingMeta
DataOp)}. PRIMARY KEY NS_ADDR. Fields:

LINEATTR unsigned int

Attributes of the cache allocated lines.

LINES unsigned int

Number of cache lines already allocated.

NS_ADDR unsigned int

The security world and address of the first byte of the page, the MSB encodes the NS bit.

PAS enum

The Physical Address Space.

TXATTR unsigned int

Attributes of the transaction.

ArchMsg.Error.cached_attributes_mismatch#lineinfo

Information about allocated lines in page. DISPLAY %
{TAG[1]:(u|U))%{TAG[2]:(d|D)}-
{TAG[63:61]:(s|rt|ns|rl|sa|na6|nsp|na7)}-%{PADDR}-%
{TAG[59:58]:(nsh|ish|osh|sys))%{TAG[56]:(|-iHittable))%
{TAG[57]:(|-oHittable)}. Fields:

PADDR unsigned int

address of this cache line.

TAG unsigned int

tag data of this cache line.

ArchMsg.Error.sci_reset

An error occurred in the System Coherency Protocol between two ports. Fields:

why string

Why the problem occurred.

ArchMsg.Warning.ns_s_dirty_hit

Cache hit at S and NS versions of the same physical address DISPLAY At PAddr %
{PADDR}
Line index %
{INDEX0} is for %
{TAG0[1]:(u|U))%
{TAG0[2]:(d|D)}-%
{TAG0[63:61]:(s|rt|ns|rl|sa|na6|nsp|na7)}-%
{PADDR}-%
{TAG0[59:58]:(nsh|ish|osh|sys))%
{TAG0[56]:(|-iHittable))%
{TAG0[57]:(|-oHittable)} Line index %
{INDEX1} is for %
{TAG1[1]:(u|U))%
{TAG1[2]:(d|D)}-%
{TAG1[63:61]:(s|rt|ns|rl|sa|na6|nsp|na7)}-%
{PADDR}-%
{TAG1[59:58]:(nsh|ish|osh|sys))%
{TAG1[56]:(|-iHittable))%
{TAG1[57]:(|-oHittable)} i.e. same address but different security regimes and one or more are dirty This is not an error if different memory is backing the lines, but otherwise can cause unpredictable cache incoherency. PRIMARY KEY PADDR. Fields:

INDEX0 unsigned int

The cache line index of the affected cache line.

INDEX1 unsigned int

The cache line index of the other regime cache line.

PADDR unsigned int

The address of the affected cache line.

TAG0 unsigned int

The tag data of the affected cache line.

TAG1 unsigned int

The tag data of the other regime cache line.

CACHE_READ_HIT

Read access cache hit. Note that this trace source causes a large slowdown in the simulation.
Fields:

IS_PRELOAD enum

Is the access a preload.

IS_SHARED enum

Is the access shared.

LATENCY unsigned int

Cache latency in ticks per access.

MANAGER_ID unsigned int

Indicates the manager ID associated with this monitor.

CACHE_READ_MISS

Read access cache miss and cache miss latency. Fields:

IS_PRELOAD enum

Is the access a preload.

IS_SHARED enum

Is the access shared.

LATENCY unsigned int

Cache miss latency in ticks per access.

MANAGER_ID unsigned int

Indicates the manager ID associated with this monitor.

CACHE_WRITE_HIT

Write access cache hit. Note that this trace source causes a large slowdown in the simulation.
Fields:

IS_PRELOAD enum

Is the access a preload.

IS_SHARED enum

Is the access shared.

LATENCY unsigned int

Cache latency in ticks per access.

MANAGER_ID unsigned int

Indicates the manager ID associated with this monitor.

CACHE_WRITE_MISS

Write access cache miss and cache miss latency. Fields:

IS_PRELOAD enum

Is the access a preload.

IS_SHARED enum

Is the access shared.

LATENCY unsigned int

Cache miss latency in ticks per access.

MANAGER_ID unsigned int

Indicates the manager ID associated with this monitor.

DVM_Message

DISPLAY DVM_msg (%{DVM_MESSAGE}). Fields:

DVM_MESSAGE string

The DVM Message.

ENTRY_BECOMES_INVALID

An entry is now invalid. The cause could be any reason. Fields:

ENTRY_INDEX unsigned int

The entry index that has become invalid.

ENTRY_SET_DIRTY

Entry change, clean/dirty. Fields:

ENTRY_INDEX unsigned int

The entry index that has become clean or dirty.

STATUS unsigned int

True if the line was made dirty.

ERROR_MIXED_ATTRIBUTES_LINE

Mismatched line and transaction attributes. Fields:

ENTRY_INDEX unsigned int

The entry index that has collided.

NS_ADDR unsigned int

The security world and address of the first byte of the line, the MSB encodes the NS bit.

PAS enum

The Physical Address Space.

ERROR_MIXED_ATTRIBUTES_PAGE

Attributes differ between page and lines in cache from the same page. Fields:

NS_ADDR unsigned int

The security world and address of the first byte of the page, the MSB encodes the NS bit.

PAS enum

The Physical Address Space.

change_dvm_disabled_ports

This reports a bitmap of which ports are disabled to receive DVM messages. Fields:

new_bitmap unsigned int

The new bitmap of disabled ports.

new_bitmap_list unsigned int

A list of port numbers that were disabled.

old_bitmap unsigned int

The old bitmap of disabled ports.

old_bitmap_list unsigned int

A list of port numbers that were disabled.

change_upstream_disabled_ports

This reports a bitmap of which upstream ports of the cache are *disabled* and so no snoop requests are sent to them. Fields:

new_bitmap unsigned int

The bitmap of the first 32 upstream ports that are disabled.

new_bitmap_list unsigned int

A list of port numbers that were disabled.

old_bitmap unsigned int

The bitmap of the first 32 upstream ports that are disabled.

old_bitmap_list unsigned int

A list of port numbers that were disabled.

entry_after_far_atomic

A cache line has had a far atomic applied to it. Fields:

atomic_operation enum

The operation that operated on this line.

be bool

The far atomic's big-endian flag.

data unsigned int

The data in the cache line.

entry_begin_address unsigned int

The start address of the entry.

entry_index unsigned int

The entry index in the cache.

incoming_data unsigned int

The incoming data from the far atomic. For CAS operations then this is compare-value and then the replacement-value.

manager_id unsigned int

The manager ID for the associated transaction.

ns enum

The security state of the line.

pas enum

The Physical Address Space.

prior_data unsigned int

The data that will be returned as the 'prior' data of the far atomic.

trans_begin_address unsigned int

The start address of the transaction.

entry_after_read

A cache line has been read. Fields:

ace_operation enum

The ACE operation that operated on this line.

data unsigned int

The data in the cache line.

dirty bool

Is the line ACE-dirty.

entry_begin_address unsigned int

The start address of the entry.

entry_index unsigned int

The entry index in the cache.

ns enum

The security state of the line.

pas enum

The Physical Address Space.

snoop bool

This transaction is a snoop transaction.

trans_begin_address unsigned int

The start address of the transaction.

unique bool

Is the line ACE-unique.

entry_after_write

A cache line has been written. Fields:

ace_operation enum

The ACE operation that operated on this line.

data unsigned int

The data in the cache line.

entry_begin_address unsigned int

The start address of the entry.

entry_index unsigned int

The entry index in the cache.

ns enum

The security state of the line.

pas enum

The Physical Address Space.

trans_begin_address unsigned int

The start address of the transaction.

sci_downstream_action

A System Coherency Action was received on the specified port. Fields:

action enum

The action to perform.

upstream_port_index unsigned int

The upstream port index that received the SCI action.

sci_downstream_action.origin

This is a description of the origin and why the action was sent if available. Fields:

description string

The description of what the originator was doing at the time.

originator string

The originator's hierarchical name.

sci_link_state

The state of the System Coherency Interface link between a master and a slave has been re-evaluated due to an event. Fields:

effect_on_dvm enum

The effect of the change on DVM messages being sent to the port.

effect_on_snoops enum

The effect of the change on snoops being sent to the port.

new_link_state enum

The new state of the link.

old_link_state enum

The state of the link.

upstream_port_index unsigned int

The upstream port index.

sci_reset_signal

The reset signals of the master and slave. Fields:

master_in_reset bool

The master is in reset.

slave_in_reset bool

The slave is in reset.

upstream_port_index unsigned int

The upstream port index.

why enum

Why this trace event was generated.

sci_upstream_action

An upstream action has been sent. Fields:

action enum

The action.

end_link_state enum

The link state that this will become.

upstream_port_index unsigned int

The upstream port index.

2.86 CMN650

This section describes the trace sources.

ASSIGNED_LCN_LDID

DISPLAY In real HW, access to memory location %{ADDR} routed through HN-S (ID = %{HNSID}) LCN with LDID = %{LCN_LDID}. Fields:

ADDR unsigned int

Address of memory location being accessed.

HNSID unsigned int

NodId of HNS node transaction is routed to.

LCN_LDID unsigned int

Programmed LDID of LCN node transaction is routed to.

ArchMsg.Error.MemoryMapped_AddressInvalid

DISPLAY Register access attempted to %{ADDRESS} does not fall into %{REG_SPACE_BASE} - %{REG_SPACE_END}. This range is where registers may be found. Fields:

ADDRESS unsigned int

Address of Access that entered RegisterInterface to the Register Space in the Interconnect.

REG_SPACE_BASE unsigned int

Register Space Base Address.

REG_SPACE_END unsigned int

Register Space End Address.

ArchMsg.Warning.MemoryMapped_ReadDisallowed

DISPLAY Read of %{REG_NAME}{%{Reg_Ownership}} not allowed from PASpace=%{PASpace}; will be read-as-zero. Fields:

OFFSET unsigned int

Offset of address within the Interconnect.

PASpace string

Physical Address Space.

REG_NAME string

Name of the register being read.

Reg_Ownership string

Register Ownership.

ArchMsg.Warning.MemoryMapped_ReadReserved

DISPLAY Reserved offset %{OFFSET} attempting to be read. Fields:

OFFSET unsigned int

Offset of address within the Interconnect.

ArchMsg.Warning.MemoryMapped_ReadWriteOnlyReg

DISPLAY Write-only register %{REG_NAME} attempting to be read; will read-as-zero. Fields:

OFFSET unsigned int

Offset of address within the Interconnect.

REG_NAME string

Name of the register being read.

ArchMsg.Warning.MemoryMapped_UnalignedReadAccess

DISPLAY Read attempted to %{OFFSET} and not aligned to 32 or 64-bit. Fields:

OFFSET unsigned int

Offset of address within the Interconnect.

ArchMsg.Warning.MemoryMapped_UnalignedWriteAccess

DISPLAY Write attempted to %{OFFSET} and not aligned to 32 or 64-bit. Fields:

OFFSET unsigned int

Offset of address within the Interconnect.

ArchMsg.Warning.MemoryMapped_WriteDisallowed

DISPLAY Write to %{REG_NAME}(%{Reg_Ownership})=%{VALUE} not allowed from PASpace=%{PASpace}; will be ignored. Fields:

OFFSET unsigned int

Offset of address within the Interconnect.

PASpace string

Physical Address Space.

REG_NAME string

Name of the register being read.

Reg_Ownership string

Register Ownership.

VALUE unsigned int

Value.

ArchMsg.Warning.MemoryMapped_WriteReadOnlyReg

DISPLAY Read-only register %{REG_NAME} attempting to be written with value %{VALUE}. Fields:

OFFSET unsigned int

Offset of address within the Interconnect.

REG_NAME string

Name of the register being written.

VALUE unsigned int

Value written.

ArchMsg.Warning.MemoryMapped_WriteReserved

DISPLAY Reserved offset %{OFFSET} attempting to be written with value %{VALUE}. Fields:

OFFSET unsigned int

Offset of address within the Interconnect.

VALUE unsigned int

Value written.

CMN_AXU_Routing

DISPLAY Access to memory location %{ADDR} is mapped to %{AXUTYPE}_AXU %{AXU_IDX} which is located on %{NODETYPE} %{NODE_IDX}. Fields:

ADDR unsigned int

Address of memory location being accessed.

AXUTYPE string

AXU Port Type (DSU, DMC, etc.).

AXU_IDX unsigned int

Index of the AXU Port.

NODETYPE string

Node type the AXU port is associated with (RNF, SNF, etc.).

NODE_IDX unsigned int

Index of the Node.

CMN_Cache_Downstream_Routing

DISPLAY Access to memory location %{ADDR} is routed to %{TYPE} port %{TO_PORT}. Fields:

ADDR unsigned int

Address of memory location being accessed.

FROM_PORT unsigned int

Upstream port.

TO_PORT unsigned int

Downstream port number to which access is routed.

TYPE string

Downstream port type to which access is routed.

CMN_Cache_Routing

DISPLAY Access to memory location %{ADDR} from port %{FROM_PORT} is routed to System Cache. Fields:

ADDR unsigned int

Address of memory location being accessed.

FROM_PORT unsigned int

Upstream port.

CMN_Invalid_OCM_Region

DISPLAY OCM region %{OCM_REGION_START}-{OCM_REGION_END} does not belong to Cache region %{CACHE_REGION_START}-{CACHE_REGION_END}. Fields:

CACHE_REGION_END unsigned int

End Address of memory location that belongs to the Cache.

CACHE_REGION_START unsigned int

Start Address of memory location that belongs to the Cache.

OCM_REGION_END unsigned int

End Address of OCM region.

OCM_REGION_START unsigned int

Start Address of OCM region.

CMN_OCM_Routing

DISPLAY Access to memory location %{ADDR} is routed to OCM offset %{OFFSET}. Fields:

ADDR unsigned int

Address of memory location being accessed.

OFFSET unsigned int

OFFSET in OCM memory location to which access is routed.

CMN_OCM_Size_Exceeded

DISPLAY Total secure & non-secure accesses exceed the OCM size which is %{OCM_SIZE}. Fields:

OCM_SIZE unsigned int

Size of the OCM.

CMN_OCM_unimplemented

DISPLAY details for unimplemented conditions encountered. Fields:

OCM_UNIMPLEMENTED string

Detail for unimplemented condition reached.

CPAG_ACCESS

DISPLAY In real HW, access to memory location %{ADDR} targets CPAG %{CPAG} and txn will be routed through CCG %{CCG_NODEID}. Fields:

ADDR unsigned int

Address of memory location being accessed.

CCG_NODEID unsigned int

NodeID of the CCG that will be used to route the txn in real HW.

CPAG unsigned int

CPAG number that the txn targets.

HNFSAM_Memory_Map

DISPLAY Memory Map of HNF%{HNF_INDEX} %{\MMAP}. Fields:

HNF_INDEX unsigned int

Index of HNF.

MMAP string

memory map of cmn600 hnfsam.

HNF_ACCESS

DISPLAY In real HW, access to memory location %{\ADDR} routed through HN-F/HN-S with node ID = %{\HNID}. Fields:

ADDR unsigned int

Address of memory location being accessed.

HNFID unsigned int

Nodeid of home cache node transation is routed to.

HNSAM_Target_SNF

DISPLAY The target SNF for address (%{\ADDRESS}) is (%{\SNF_NUM}). Fields:

ADDRESS unsigned int

Transaction Address.

SNF_NUM unsigned int

SNF Node Index which is targeted from HNFSAM.

Interconnect_ABF_policy_reg

DISPLAY ABF policy register write details. Fields:

ABF_PR_WRITE string

Detail for policy register write.

Interconnect_do_abf_call

DISPLAY Cache state modelled %{\CACHE_STATE_MODELLED} ABF commanded for addr %{\ABF_REGION_START}-%{\ABF_REGION_END} with mode %{\ABF_MODE} and . Fields:

ABF_MODE unsigned int

0 = CLEAN_INVALID, 1 = MAKE_INVALID, 2 = CLEAN_SHARED, 3 = RESERVED.

ABF_REGION_END unsigned int

End Address of ABF region.

ABF_REGION_START unsigned int

Start Address of ABF region.

CACHE_STATE_MODELLED bool

cache_state_modelled, true = yes, false = no.

LCN_ACCESS

DISPLAY In real HW, access to memory location %{ADDR} routed through LCN with node ID = %{LCN_NODEID}}. Fields:

ADDR unsigned int

Address of memory location being accessed.

LCN_NODEID unsigned int

Node Id of LCN node transaction is routed to.

MemoryMapped_Read

Trace read from a memory-mapped register. Fields:

OFFSET unsigned int

Offset of address within the Interconnect.

PASpace string

Physical Address Space.

REG_NAME string

Name of the register being read.

VALUE unsigned int

Value returned.

MemoryMapped_Write

Trace write to a memory-mapped register. Fields:

OFFSET unsigned int

Offset of address within the Interconnect.

PASpace string

Physical Address Space.

REG_NAME string

Name of the register being written.

UPDATED_VALUE unsigned int

New value read back from the register; for write-only registers this will be equal to VALUE.

VALUE unsigned int

Value written.

OCM_Regions

DISPLAY Regions configured as OCM %{REGIONS}. Fields:

REGIONS string

memory regions allocated to OCM.

RNSAM_Memory_Map

DISPLAY Memory Map of RNSAM %{MMAP}. Fields:

MMAP string

memory map of cmn600 rnsam.

RNSAM_Target_HNF

DISPLAY The target HNF for address ({ADDRESS}) is ({HNF_NUM}). Fields:

ADDRESS unsigned int

Transaction Address.

HNF_NUM unsigned int

HNF Node Index to which the transaction is routed.

SNF_HASHING_Target_SNF

DISPLAY The target SNF for address ({ADDRESS}) in default hashed region is ({HASHED_SNF}). But the model is routing the transaction to SNF indicated by the HNSAM_Target_SNF trace. Fields:

ADDRESS unsigned int

Transaction Address.

SNF_NUM unsigned int

SNF Node Index which is targeted from HNSAM hashing for default region.

Warning.Hashing_enabled_for_cache_group

DISPLAY The SCG group (which contains HNF{INDEX}) has different SNF default ports in member HNFs which is not supported. Fields:

HNF_INDEX unsigned int

HNF Node index which is part of system cache group.

Warning.RNSAM_Not_Unique

DISPLAY RNSAM configuration should be same for all masters. The RNSAM for master {INDEX} is different. Fields:

INDEX unsigned int

Index of RN component. Index starts with RNFs and then the RNIs.

Warning.event_cpu_events_disabled

DISPLAY Event not propagated because CPU Events are disabled for CCIX link{INDEX}. Fields:

INDEX unsigned int

CCIX link index.

Warning.event_smp_mode_disabled

DISPLAY Event not propagated because SMP mode is not enabled.

Warning.rnsam_and_cxrasam_mismatch

DISPLAY RNSAM and CXRASAM address regions don't match. You may need to review its configuration.

event_propagation_downstream

DISPLAY Event changed from downstream (from other chips) from port: %{FROM_PORT} to state: %{STATE}. Fields:

FROM_PORT unsigned int

Activated port.

STATE unsigned int

State being transmitted.

event_propagation_downstream_routing

DISPLAY Event propagated from downstream (from other systems) from port: %{FROM_PORT} to link: %{LINK} state: %{STATE} sent to local clusters. Fields:

FROM_PORT unsigned int

Activated port.

LINK unsigned int

Link.

STATE unsigned int

State being transmitted.

event_propagation_upstream

DISPLAY Event changed from upstream (clusters) sent to another chip via port %{TO_PORT} on link %{LINK} state: %{STATE}. Fields:

LINK unsigned int

Link number.

STATE unsigned int

State being transmitted.

TO_PORT unsigned int

Target port.

trace_add_cxra_memory_region

DISPLAY CXRA%{INDEX} SAM memory region: [%{START}-%{END}] => HAID=%{HAID}. Fields:

END unsigned int

End address of the memory region.

HAID unsigned int

Memory region target HAID.

INDEX unsigned int

CXRA/CCG_RA index.

START unsigned int

Start address of the memory region.

trace_add_memory_region

DISPLAY Add new memory region type %{TYPE}, range start : %{START}, and end: %{END}.

Fields:

END unsigned int

End address of the memory region.

START unsigned int

Start address of the memory region.

TYPE enum

Node type.

trace_hnd

DISPLAY The HND index for XP%{INDEX} is %{HND_INDEX}. Fields:

HND_INDEX unsigned int

HND index for the XP.

INDEX unsigned int

XP Index.

2.87 CMN650R2

This section describes the trace sources.

ASSIGNED_LCN_LDID

DISPLAY In real HW, access to memory location %{ADDR} routed through HN-S (ID = %{HNSID}) LCN with LDID = %{LCN_LDID}. Fields:

ADDR unsigned int

Address of memory location being accessed.

HNSID unsigned int

NodeID of HNS node transaction is routed to.

LCN_LDID unsigned int

Programmed LDID of LCN node transaction is routed to.

ArchMsg.Error.MemoryMapped_AddressInvalid

DISPLAY Register access attempted to %{ADDRESS} does not fall into %{REG_SPACE_BASE} - %{REG_SPACE_END}. This range is where registers may be found. Fields:

ADDRESS unsigned int

Address of Access that entered RegisterInterface to the Register Space in the Interconnect.

REG_SPACE_BASE unsigned int

Register Space Base Address.

REG_SPACE_END unsigned int

Register Space End Address.

ArchMsg.Warning.MemoryMapped_ReadDisallowed

DISPLAY Read of %{REG_NAME}{%{Reg_Ownership}} not allowed from PASpace=%{PASpace}; will be read-as-zero. Fields:

OFFSET unsigned int

Offset of address within the Interconnect.

PASpace string

Physical Address Space.

REG_NAME string

Name of the register being read.

Reg_Ownership string

Register Ownership.

ArchMsg.Warning.MemoryMapped_ReadReserved

DISPLAY Reserved offset %{OFFSET} attempting to be read. Fields:

OFFSET unsigned int

Offset of address within the Interconnect.

ArchMsg.Warning.MemoryMapped_ReadWriteOnlyReg

DISPLAY Write-only register %{REG_NAME} attempting to be read; will read-as-zero. Fields:

OFFSET unsigned int

Offset of address within the Interconnect.

REG_NAME string

Name of the register being read.

ArchMsg.Warning.MemoryMapped_UnalignedReadAccess

DISPLAY Read attempted to %{OFFSET} and not aligned to 32 or 64-bit. Fields:

OFFSET unsigned int

Offset of address within the Interconnect.

ArchMsg.Warning.MemoryMapped_UnalignedWriteAccess

DISPLAY Write attempted to %{OFFSET} and not aligned to 32 or 64-bit. Fields:

OFFSET unsigned int

Offset of address within the Interconnect.

ArchMsg.Warning.MemoryMapped_WriteDisallowed

DISPLAY Write to %{REG_NAME}{%{Reg_Ownership}}=%{VALUE} not allowed from PAspace=%{PASpace}; will be ignored. Fields:

OFFSET unsigned int

Offset of address within the Interconnect.

PASpace string

Physical Address Space.

REG_NAME string

Name of the register being read.

Reg_Ownership string

Register Ownership.

VALUE unsigned int

Value.

ArchMsg.Warning.MemoryMapped_WriteReadOnlyReg

DISPLAY Read-only register %{REG_NAME} attempting to be written with value %{VALUE}.

Fields:

OFFSET unsigned int

Offset of address within the Interconnect.

REG_NAME string

Name of the register being written.

VALUE unsigned int

Value written.

ArchMsg.Warning.MemoryMapped_WriteReserved

DISPLAY Reserved offset %{OFFSET} attempting to be written with value %{VALUE}. Fields:

OFFSET unsigned int

Offset of address within the Interconnect.

VALUE unsigned int

Value written.

CMN_AXU_Routing

DISPLAY Access to memory location %{ADDR} is mapped to %{AXUTYPE}_AXU %{AXU_IDX} which is located on %{NODETYPE} %{NODE_IDX}. Fields:

ADDR unsigned int

Address of memory location being accessed.

AXUTYPE string

AXU Port Type (DSU, DMC, etc.).

AXU_IDX unsigned int

Index of the AXU Port.

NODETYPE string

Node type the AXU port is associated with (RNF, SNF, etc.).

NODE_IDX unsigned int

Index of the Node.

CMN_Cache_Downstream_Routing

DISPLAY Access to memory location %{ADDR} is routed to %{TYPE} port %{TO_PORT}. Fields:

ADDR unsigned int

Address of memory location being accessed.

FROM_PORT unsigned int

Upstream port.

TO_PORT unsigned int

Downstream port number to which access is routed.

TYPE string

Downstream port type to which access is routed.

CMN_Cache_Routing

DISPLAY Access to memory location %{ADDR} from port %{FROM_PORT} is routed to System Cache. Fields:

ADDR unsigned int

Address of memory location being accessed.

FROM_PORT unsigned int

Upstream port.

CMN_Invalid_OCM_Region

DISPLAY OCM region %{OCM_REGION_START}-{OCM_REGION_END} does not belong to Cache region %{CACHE_REGION_START}-{CACHE_REGION_END}. Fields:

CACHE_REGION_END unsigned int

End Address of memory location that belongs to the Cache.

CACHE_REGION_START unsigned int

Start Address of memory location that belongs to the Cache.

OCM_REGION_END unsigned int

End Address of OCM region.

OCM_REGION_START unsigned int

Start Address of OCM region.

CMN_OCM_Routing

DISPLAY Access to memory location %{ADDR} is routed to OCM offset %{OFFSET}. Fields:

ADDR unsigned int

Address of memory location being accessed.

OFFSET unsigned int

OFFSET in OCM memory location to which access is routed.

CMN_OCM_Size_Exceeded

DISPLAY Total secure & non-secure accesses exceed the OCM size which is %{OCM_SIZE}. Fields:

OCM_SIZE unsigned int

Size of the OCM.

CMN_OCM_unimplemented

DISPLAY details for unimplemented conditions encountered. Fields:

OCM_UNIMPLEMENTED string

Detail for unimplemented condition reached.

CPAG_ACCESS

DISPLAY In real HW, access to memory location %{ADDR} targets CPAG %{CPAG} and txn will be routed through CCG %{CCG_NODEID}. Fields:

ADDR unsigned int

Address of memory location being accessed.

CCG_NODEID unsigned int

NodeID of the CCG that will be used to route the txn in real HW.

CPAG unsigned int

CPAG number that the txn targets.

HNFSAM_Memory_Map

DISPLAY Memory Map of HNF%{HNF_INDEX} %{MMAP}. Fields:

HNF_INDEX unsigned int

Index of HNF.

MMAP string

memory map of cmn600 hnfsam.

HNFS_ACCESS

DISPLAY In real HW, access to memory location %{ADDR} routed through HN-F/HN-S with node ID = %{HNID}. Fields:

ADDR unsigned int

Address of memory location being accessed.

HNFSID unsigned int

NodeID of home cache node transaction is routed to.

HNSAM_Target_SNF

DISPLAY The target SNF for address (%{ADDRESS}) is (%{SNF_NUM}). Fields:

ADDRESS unsigned int

Transaction Address.

SNF_NUM unsigned int

SNF Node Index which is targeted from HNFSAM.

Interconnect_ABF_policy_reg

DISPLAY ABF policy register write details. Fields:

ABF_PR_WRITE string

Detail for policy register write.

Interconnect_do_abf_call

DISPLAY Cache state modelled %{CACHE_STATE_MODELLED} ABF commanded for addr %{ABF_REGION_START}-{ABF_REGION_END} with mode %{ABF_MODE} and . Fields:

ABF_MODE unsigned int

0 = CLEAN_INVALID, 1 = MAKE_INVALID, 2 = CLEAN_SHARED, 3 = RESERVED.

ABF_REGION_END unsigned int

End Address of ABF region.

ABF_REGION_START unsigned int

Start Address of ABF region.

CACHE_STATE_MODELLED bool

cache_state_modelled, true = yes, false = no.

LCN_ACCESS

DISPLAY In real HW, access to memory location %{ADDR} routed through LCN with node ID = %{LCN_NODEID}}. Fields:

ADDR unsigned int

Address of memory location being accessed.

LCN_NODEID unsigned int

Node Id of LCN node transation is routed to.

MemoryMapped_Read

Trace read from a memory-mapped register. Fields:

OFFSET unsigned int

Offset of address within the Interconnect.

PASpace string

Physical Address Space.

REG_NAME string

Name of the register being read.

VALUE unsigned int

Value returned.

MemoryMapped_Write

Trace write to a memory-mapped register. Fields:

OFFSET unsigned int

Offset of address within the Interconnect.

PASpace string

Physical Address Space.

REG_NAME string

Name of the register being written.

UPDATED_VALUE unsigned int

New value read back from the register; for write-only registers this will be equal to VALUE.

VALUE unsigned int

Value written.

OCM_Regions

DISPLAY Regions configured as OCM %{REGIONS}. Fields:

REGIONS string

memory regions allocated to OCM.

RNSAM_Memory_Map

DISPLAY Memory Map of RNSAM %{MMAP}. Fields:

MMAP string

memory map of cmn600 rnsam.

RNSAM_Target_HNF

DISPLAY The target HNF for address (%{ADDRESS}) is (%{HNF_NUM}). Fields:

ADDRESS unsigned int

Transaction Address.

HNF_NUM unsigned int

HNF Node Index to which the transaction is routed.

SNF_HASHING_Target_SNF

DISPLAY The target SNF for address (%{ADDRESS}) in default hashed region is (%{HASHED_SNF}). But the model is routing the transaction to SNF indicated by the HNSAM_Target_SNF trace. Fields:

ADDRESS unsigned int

Transaction Address.

SNF_NUM unsigned int

SNF Node Index which is targeted from HNSAM hashing for default region.

Warning.Hashing_enabled_for_cache_group

DISPLAY The SCG group (which contains HNF%{INDEX}) has different SNF default ports in member HNFs which is not supported. Fields:

HNf_INDEX unsigned int

HNf Node index which is part of system cache group.

Warning.RNSAM_Not_Unique

DISPLAY RNSAM configuration should be same for all masters. The RNSAM for master %{INDEX} is different. Fields:

INDEX unsigned int

Index of RN component. Index starts with RNFs and then the RNIs.

Warning.event_cpu_events_disabled

DISPLAY Event not propagated because CPU Events are disabled for CCIX link %{INDEX}. Fields:

INDEX unsigned int

CCIX link index.

Warning.event_smp_mode_disabled

DISPLAY Event not propagated because SMP mode is not enabled.

Warning.rnsam_and_cxrasam_mismatch

DISPLAY RNSAM and CXRASAM address regions don't match. You may need to review its configuration.

event_propagation_downstream

DISPLAY Event changed from downstream (from other chips) from port: %{FROM_PORT} to state: %{STATE}. Fields:

FROM_PORT unsigned int

Activated port.

STATE unsigned int

State being transmitted.

event_propagation_downstream_routing

DISPLAY Event propagated from downstream (from other systems) from port: %{FROM_PORT} to link: %{LINK} state: %{STATE} sent to local clusters. Fields:

FROM_PORT unsigned int

Activated port.

LINK unsigned int

Link.

STATE unsigned int

State being transmitted.

event_propagation_upstream

DISPLAY Event changed from upstream (clusters) sent to another chip via port %{TO_PORT} on link %{LINK} state: %{STATE}. Fields:

LINK unsigned int

Link number.

STATE unsigned int

State being transmitted.

TO_PORT unsigned int

Target port.

trace_add_cxra_memory_region

DISPLAY CXRA%{INDEX} SAM memory region: [%{START}-%{END}] => HAID=%{HAID}.

Fields:

END unsigned int

End address of the memory region.

HAID unsigned int

Memory region target HAID.

INDEX unsigned int

CXRA/CCG_RA index.

START unsigned int

Start address of the memory region.

trace_add_memory_region

DISPLAY Add new memory region type %{TYPE}, range start : %{START}, and end: %{END}.

Fields:

END unsigned int

End address of the memory region.

START unsigned int

Start address of the memory region.

TYPE enum

Node type.

trace_hnd

DISPLAY The HND index for XP%{INDEX} is %{HND_INDEX}. Fields:

HND_INDEX unsigned int

HND index for the XP.

INDEX unsigned int

XP Index.

2.88 CMN700

This section describes the trace sources.

A4S_Routing

DISPLAY A4S access from GICD to A4S_TDEST %{A4S_TDEST} is being routed to downstream port tx_cxs_a4s[%{PORT}]. Fields:

A4S_TDEST unsigned int

A4S_TDEST of the incoming transaction.

PORT unsigned int

PORT in a4srouter to which access is routed.

A4S_Routing_Abort

DISPLAY A4S access from GICD to A4S_TDEST %{A4S_TDEST} is aborted, unknown A4S_TDEST. Fields:

A4S_TDEST unsigned int

A4S_TDEST of the incoming transaction.

ASSIGNED_LCN_LDID

DISPLAY In real HW, access to memory location %{ADDR} routed through HN-S (ID = %{HNSID}) LCN with LDID = %{LCN_LDID}. Fields:

ADDR unsigned int

Address of memory location being accessed.

HNSID unsigned int

NodeID of HNS node transaction is routed to.

LCN_LDID unsigned int

Programmed LDID of LCN node transaction is routed to.

ArchMsg.Error.MemoryMapped_AddressInvalid

DISPLAY Register access attempted to %{ADDRESS} does not fall into %{REG_SPACE_BASE} - %{REG_SPACE_END}. This range is where registers may be found. Fields:

ADDRESS unsigned int

Address of Access that entered RegisterInterface to the Register Space in the Interconnect.

REG_SPACE_BASE unsigned int

Register Space Base Address.

REG_SPACE_END unsigned int

Register Space End Address.

ArchMsg.Warning.MemoryMapped_ReadDisallowed

DISPLAY Read of %{REG_NAME}(%{Reg_Ownership}) not allowed from PASpace= %{PASpace}; will be read-as-zero. Fields:

OFFSET unsigned int

Offset of address within the Interconnect.

PASpace string

Physical Address Space.

REG_NAME string

Name of the register being read.

Reg_Ownership string

Register Ownership.

ArchMsg.Warning.MemoryMapped_ReadReserved

DISPLAY Reserved offset %{OFFSET} attempting to be read. Fields:

OFFSET unsigned int

Offset of address within the Interconnect.

ArchMsg.Warning.MemoryMapped_ReadWriteOnlyReg

DISPLAY Write-only register %{REG_NAME} attempting to be read; will read-as-zero. Fields:

OFFSET unsigned int

Offset of address within the Interconnect.

REG_NAME string

Name of the register being read.

ArchMsg.Warning.MemoryMapped_UnalignedReadAccess

DISPLAY Read attempted to %{OFFSET} and not aligned to 32 or 64-bit. Fields:

OFFSET unsigned int

Offset of address within the Interconnect.

ArchMsg.Warning.MemoryMapped_UnalignedWriteAccess

DISPLAY Write attempted to %{OFFSET} and not aligned to 32 or 64-bit. Fields:

OFFSET unsigned int

Offset of address within the Interconnect.

ArchMsg.Warning.MemoryMapped_WriteDisallowed

DISPLAY Write to %{REG_NAME}{%{Reg_Ownership}}=%{VALUE} not allowed from PASpace=%{PASpace}; will be ignored. Fields:

OFFSET unsigned int

Offset of address within the Interconnect.

PASpace string

Physical Address Space.

REG_NAME string

Name of the register being read.

Reg_Ownership string

Register Ownership.

VALUE unsigned int

Value.

ArchMsg.Warning.MemoryMapped_WriteReadOnlyReg

DISPLAY Read-only register %{REG_NAME} attempting to be written with value %{VALUE}. Fields:

OFFSET unsigned int

Offset of address within the Interconnect.

REG_NAME string

Name of the register being written.

VALUE unsigned int

Value written.

ArchMsg.Warning.MemoryMapped_WriteReserved

DISPLAY Reserved offset %{OFFSET} attempting to be written with value %{VALUE}. Fields:

OFFSET unsigned int

Offset of address within the Interconnect.

VALUE unsigned int

Value written.

CMN_AXU_Routing

DISPLAY Access to memory location %{ADDR} is mapped to %{AXUTYPE}_AXU %{AXU_IDX} which is located on %{NODETYPE} %{NODE_IDX}. Fields:

ADDR unsigned int

Address of memory location being accessed.

AXUTYPE string

AXU Port Type (DSU, DMC, etc.).

AXU_IDX unsigned int

Index of the AXU Port.

NODETYPE string

Node type the AXU port is associated with (RNF, SNF, etc.).

NODE_IDX unsigned int

Index of the Node.

CMN_Cache_Downstream_Routing

DISPLAY Access to memory location %{ADDR} is routed to %{TYPE} port %{TO_PORT}. Fields:

ADDR unsigned int

Address of memory location being accessed.

FROM_PORT unsigned int

Upstream port.

TO_PORT unsigned int

Downstream port number to which access is routed.

TYPE string

Downstream port type to which access is routed.

CMN_Cache_Routing

DISPLAY Access to memory location %{ADDR} from port %{FROM_PORT} is routed to System Cache. Fields:

ADDR unsigned int

Address of memory location being accessed.

FROM_PORT unsigned int

Upstream port.

CMN_Invalid_OCM_Region

DISPLAY OCM region %{OCM_REGION_START}-{OCM_REGION_END} does not belong to Cache region %{CACHE_REGION_START}-{CACHE_REGION_END}. Fields:

CACHE_REGION_END unsigned int

End Address of memory location that belongs to the Cache.

CACHE_REGION_START unsigned int

Start Address of memory location that belongs to the Cache.

OCM_REGION_END unsigned int

End Address of OCM region.

OCM_REGION_START unsigned int

Start Address of OCM region.

CMN_OCM_Routing

DISPLAY Access to memory location %{ADDR} is routed to OCM offset %{OFFSET}. Fields:

ADDR unsigned int

Address of memory location being accessed.

OFFSET unsigned int

OFFSET in OCM memory location to which access is routed.

CMN_OCM_Size_Exceeded

DISPLAY Total secure & non-secure accesses exceed the OCM size which is %{OCM_SIZE}. Fields:

OCM_SIZE unsigned int

Size of the OCM.

CMN_OCM_unimplemented

DISPLAY details for unimplemented conditions encountered. Fields:

OCM_UNIMPLEMENTED string

Detail for unimplemented condition reached.

CPAG_ACCESS

DISPLAY In real HW, access to memory location %{ADDR} targets CPAG %{CPAG} and txn will be routed through CCG %{CCG_NODEID}. Fields:

ADDR unsigned int

Address of memory location being accessed.

CCG_NODEID unsigned int

NodeID of the CCG that will be used to route the txn in real HW.

CPAG unsigned int

CPAG number that the txn targets.

HNFSAM_Memory_Map

DISPLAY Memory Map of HNF%{HNF_INDEX} %{MMAP}. Fields:

HNF_INDEX unsigned int

Index of HNF.

MMAP string

memory map of cmn600 hnfsam.

HNF_ACCESS

DISPLAY In real HW, access to memory location %{ADDR} routed through HN-F/HN-S with node ID = %{HNID}. Fields:

ADDR unsigned int

Address of memory location being accessed.

HNID unsigned int

NodeID of home cache node transaction is routed to.

HNSAM_Target_SNF

DISPLAY The target SNF for address (%{ADDRESS}) is (%{SNF_NUM}). Fields:

ADDRESS unsigned int

Transaction Address.

SNF_NUM unsigned int

SNF Node Index which is targeted from HNFSAM.

Interconnect_ABF_policy_reg

DISPLAY ABF policy register write details. Fields:

ABF_PR_WRITE string

Detail for policy register write.

Interconnect_do_abf_call

DISPLAY Cache state modelled %{CACHE_STATE_MODELLED} ABF commanded for addr %{ABF_REGION_START}-{ABF_REGION_END} with mode %{ABF_MODE} and . Fields:

ABF_MODE unsigned int

0 = CLEAN_INVALID, 1 = MAKE_INVALID, 2 = CLEAN_SHARED, 3 = RESERVED.

ABF_REGION_END unsigned int

End Address of ABF region.

ABF_REGION_START unsigned int

Start Address of ABF region.

CACHE_STATE_MODELLED bool

cache_state_modelled, true = yes, false = no.

LCN_ACCESS

DISPLAY In real HW, access to memory location %{ADDR} routed through LCN with node ID = %{LCN_NODEID}}. Fields:

ADDR unsigned int

Address of memory location being accessed.

LCN_NODEID unsigned int

Node Id of LCN node transaction is routed to.

MemoryMapped_Read

Trace read from a memory-mapped register. Fields:

OFFSET unsigned int

Offset of address within the Interconnect.

PASpace string

Physical Address Space.

REG_NAME string

Name of the register being read.

VALUE unsigned int

Value returned.

MemoryMapped_Write

Trace write to a memory-mapped register. Fields:

OFFSET unsigned int

Offset of address within the Interconnect.

PASpace string

Physical Address Space.

REG_NAME string

Name of the register being written.

UPDATED_VALUE unsigned int

New value read back from the register; for write-only registers this will be equal to VALUE.

VALUE unsigned int

Value written.

OCM_Regions

DISPLAY Regions configured as OCM %{REGIONS}. Fields:

REGIONS string

memory regions allocated to OCM.

RAS_BEHAVIOR_TRACE

Trace to capture the RAS behavior. Fields:

RAS_INFO string

Gives information about the error detected.

RNSAM_Memory_Map

DISPLAY Memory Map of RNSAM %{MMAP}. Fields:

MMAP string

memory map of cmn600 rnsam.

RNSAM_Target_HNF

DISPLAY The target HNF for address (%{ADDRESS}) is (%{HNF_NUM}). Fields:

ADDRESS unsigned int

Transaction Address.

HNF_NUM unsigned int

HNF Node Index to which the transaction is routed.

SNF_HASHING_Target_SNF

DISPLAY The target SNF for address (%{ADDRESS}) in default hashed region is (%{HASHED_SNF}). But the model is routing the transaction to SNF indicated by the HNSAM_Target_SNF trace. Fields:

ADDRESS unsigned int

Transaction Address.

SNF_NUM unsigned int

SNF Node Index which is targeted from HNSAM hashing for default region.

Warning.Hashing_enabled_for_cache_group

DISPLAY The SCG group (which contains HNF%{INDEX}) has different SNF default ports in member HNFs which is not supported. Fields:

HNF_INDEX unsigned int

HNF Node index which is part of system cache group.

Warning.MTU_MODE_Not_Unique

DISPLAY MTU node configuration should be same for all devices. The MTU node %{INDEX} in mode %{MODE} is different the current mode %{CURRENT_MODE}. Fields:

CURRENT_MODE enum

Execution mode for another device already configured.

INDEX unsigned int

MTU index being configured.

MODE enum

Execution mode being configured.

Warning.MTU_Shuttering_Reg_Not_Unique

DISPLAY MTU shuttering register value should be same for all devices. Register %{REG_INDEX} in MTU %{MTU_INDEX} with value %{VALUE} is different the current value %{CURRENT_VALUE}. Fields:

CURRENT_VALUE unsigned int

Shuttering register index already configured.

MTU_INDEX unsigned int

MTU index being configured.

REG_INDEX unsigned int

Register index being configured.

VALUE unsigned int

Shuttering register index being configured.

Warning.RNSAM_Not_Unique

DISPLAY RNSAM configuration should be same for all masters. The RNSAM for master %{INDEX} is different. Fields:

INDEX unsigned int

Index of RN component. Index starts with RNFs and then the RNIs.

Warning.event_cpu_events_disabled

DISPLAY Event not propagated because CPU Events are disabled for CCIX link %{INDEX}. Fields:

INDEX unsigned int

CCIX link index.

Warning.event_smp_mode_disabled

DISPLAY Event not propagated because SMP mode is not enabled.

Warning.rnsam_and_cxrasam_mismatch

DISPLAY RNSAM and CXRASAM address regions don't match. You may need to review its configuration.

event_propagation_downstream

DISPLAY Event changed from downstream (from other chips) from port: %{FROM_PORT} to state: %{STATE}. Fields:

FROM_PORT unsigned int

Activated port.

STATE unsigned int

State being transmitted.

event_propagation_downstream_routing

DISPLAY Event propagated from downstream (from other systems) from port: %{FROM_PORT} to link: %{LINK} state: %{STATE} sent to local clusters. Fields:

FROM_PORT unsigned int

Activated port.

LINK unsigned int

Link.

STATE unsigned int

State being transmitted.

event_propagation_upstream

DISPLAY Event changed from upstream (clusters) sent to another chip via port %{TO_PORT} on link %{LINK} state: %{STATE}. Fields:

LINK unsigned int

Link number.

STATE unsigned int

State being transmitted.

TO_PORT unsigned int

Target port.

trace_add_cxra_memory_region

DISPLAY CXRA%{INDEX} SAM memory region: [%{START}-%{END}] => HAID=%{HAID}. Fields:

END unsigned int

End address of the memory region.

HAID unsigned int

Memory region target HAID.

INDEX unsigned int

CXRA/CCG_RA index.

START unsigned int

Start address of the memory region.

trace_add_memory_region

DISPLAY Add new memory region type %{TYPE}, range start : %{START}, and end: %{END}.

Fields:

END unsigned int

End address of the memory region.

START unsigned int

Start address of the memory region.

TYPE enum

Node type.

trace_hnd

DISPLAY The HND index for XP%{INDEX} is %{HND_INDEX}. Fields:

HND_INDEX unsigned int

HND index for the XP.

INDEX unsigned int

XP Index.

2.89 CMNTAGCACHECADI

This section describes the trace sources.

MTU_address_generation

DISPLAY MTU Address generation: Mode %{MODE}, original address %{ORIGINAL}, translated: %{TRANSLATION}, and shuttered: %{SHUTTER}. Fields:

MODE enum

Translation mode configured.

ORIGINAL unsigned int

Original transaction address.

SHUTTER unsigned int

Address after shuttering.

TRANSLATION unsigned int

Address after translation.

MTU_shuttering_offset

DISPLAY Shuttering bit %{INDEX} by %{OFFSET}. Fields:

INDEX unsigned int

Address bit index.

OFFSET unsigned int

Shifted by this amount.

MTU_translation_mode

DISPLAY MTU Address generation mode set to %{MODE}. Fields:

MODE enum

Translation mode being configured.

2.90 CMN_S3

This section describes the trace sources.

A4S_Routing

DISPLAY A4S access from GICD to A4S_TDEST %{A4S_TDEST} is being routed to downstream port tx_cxs_a4s[%{PORT}]. Fields:

A4S_TDEST unsigned int

A4S_TDEST of the incoming transaction.

PORT unsigned int

PORT in a4srouter to which access is routed.

A4S_Routing_Abort

DISPLAY A4S access from GICD to A4S_TDEST %{A4S_TDEST} is aborted, unknown A4S_TDEST. Fields:

A4S_TDEST unsigned int

A4S_TDEST of the incoming transaction.

ASSIGNED_LCN_LDID

DISPLAY In real HW, access to memory location %{ADDR} routed through HN-S (ID = %{HNSID}) LCN with LDID = %{LCN_LDID}. Fields:

ADDR unsigned int

Address of memory location being accessed.

HNSID unsigned int

NodeID of HNS node transaction is routed to.

LCN_LDID unsigned int

Programmed LDID of LCN node transaction is routed to.

ArchMsg.Error.MemoryMapped_AddressInvalid

DISPLAY Register access attempted to %{ADDRESS} does not fall into %{REG_SPACE_BASE} - %{REG_SPACE_END}. This range is where registers may be found. Fields:

ADDRESS unsigned int

Address of Access that entered RegisterInterface to the Register Space in the Interconnect.

REG_SPACE_BASE unsigned int

Register Space Base Address.

REG_SPACE_END unsigned int

Register Space End Address.

ArchMsg.Warning.MemoryMapped_ReadDisallowed

DISPLAY Read of %{REG_NAME}{%{Reg_Ownership}} not allowed from PASpace=%{PASpace}; will be read-as-zero. Fields:

OFFSET unsigned int

Offset of address within the Interconnect.

PASpace string

Physical Address Space.

REG_NAME string

Name of the register being read.

Reg_Ownership string

Register Ownership.

ArchMsg.Warning.MemoryMapped_ReadReserved

DISPLAY Reserved offset %{OFFSET} attempting to be read. Fields:

OFFSET unsigned int

Offset of address within the Interconnect.

ArchMsg.Warning.MemoryMapped_ReadWriteOnlyReg

DISPLAY Write-only register %{REG_NAME} attempting to be read; will read-as-zero. Fields:

OFFSET unsigned int

Offset of address within the Interconnect.

REG_NAME string

Name of the register being read.

ArchMsg.Warning.MemoryMapped_UnalignedReadAccess

DISPLAY Read attempted to %{OFFSET} and not aligned to 32 or 64-bit. Fields:

OFFSET unsigned int

Offset of address within the Interconnect.

ArchMsg.Warning.MemoryMapped_UnalignedWriteAccess

DISPLAY Write attempted to %{OFFSET} and not aligned to 32 or 64-bit. Fields:

OFFSET unsigned int

Offset of address within the Interconnect.

ArchMsg.Warning.MemoryMapped_WriteDisallowed

DISPLAY Write to %{REG_NAME}{%{Reg_Ownership}}=%{VALUE} not allowed from PAspace=%{PASpace}; will be ignored. Fields:

OFFSET unsigned int

Offset of address within the Interconnect.

PASpace string

Physical Address Space.

REG_NAME string

Name of the register being read.

Reg_Ownership string

Register Ownership.

VALUE unsigned int

Value.

ArchMsg.Warning.MemoryMapped_WriteReadOnlyReg

DISPLAY Read-only register %{REG_NAME} attempting to be written with value %{VALUE}.

Fields:

OFFSET unsigned int

Offset of address within the Interconnect.

REG_NAME string

Name of the register being written.

VALUE unsigned int

Value written.

ArchMsg.Warning.MemoryMapped_WriteReserved

DISPLAY Reserved offset %{OFFSET} attempting to be written with value %{VALUE}. Fields:

OFFSET unsigned int

Offset of address within the Interconnect.

VALUE unsigned int

Value written.

CMN_AXU_Routing

DISPLAY Access to memory location %{ADDR} is mapped to %{AXUTYPE}_AXU %{AXU_IDX} which is located on %{NODETYPE} %{NODE_IDX}. Fields:

ADDR unsigned int

Address of memory location being accessed.

AXUTYPE string

AXU Port Type (DSU, DMC, etc.).

AXU_IDX unsigned int

Index of the AXU Port.

NODETYPE string

Node type the AXU port is associated with (RNF, SNF, etc.).

NODE_IDX unsigned int

Index of the Node.

CMN_Cache_Downstream_Routing

DISPLAY Access to memory location %{ADDR} is routed to %{TYPE} port %{TO_PORT}.
Fields:

ADDR unsigned int

Address of memory location being accessed.

FROM_PORT unsigned int

Upstream port.

TO_PORT unsigned int

Downstream port number to which access is routed.

TYPE string

Downstream port type to which access is routed.

CMN_Cache_Routing

DISPLAY Access to memory location %{ADDR} from port %{FROM_PORT} is routed to System Cache. Fields:

ADDR unsigned int

Address of memory location being accessed.

FROM_PORT unsigned int

Upstream port.

CMN_Invalid_OCM_Region

DISPLAY OCM region %{OCM_REGION_START}-{OCM_REGION_END} does not belong to Cache region %{CACHE_REGION_START}-{CACHE_REGION_END}. Fields:

CACHE_REGION_END unsigned int

End Address of memory location that belongs to the Cache.

CACHE_REGION_START unsigned int

Start Address of memory location that belongs to the Cache.

OCM_REGION_END unsigned int

End Address of OCM region.

OCM_REGION_START unsigned int

Start Address of OCM region.

CMN_OCM_Routing

DISPLAY Access to memory location %{ADDR} is routed to OCM offset %{OFFSET}. Fields:

ADDR unsigned int

Address of memory location being accessed.

OFFSET unsigned int

OFFSET in OCM memory location to which access is routed.

CMN_OCM_Size_Exceeded

DISPLAY Total secure & non-secure accesses exceed the OCM size which is %{OCM_SIZE}. Fields:

OCM_SIZE unsigned int

Size of the OCM.

CMN_OCM_unimplemented

DISPLAY details for unimplemented conditions encountered. Fields:

OCM_UNIMPLEMENTED string

Detail for unimplemented condition reached.

CPAG_ACCESS

DISPLAY In real HW, access to memory location %{ADDR} targets CPAG %{CPAG} and txn will be routed through CCG %{CCG_NODEID}. Fields:

ADDR unsigned int

Address of memory location being accessed.

CCG_NODEID unsigned int

NodeID of the CCG that will be used to route the txn in real HW.

CPAG unsigned int

CPAG number that the txn targets.

HNFSAM_Memory_Map

DISPLAY Memory Map of HNF%{HNF_INDEX} %{MMAP}. Fields:

HNF_INDEX unsigned int

Index of HNF.

MMAP string

memory map of cmn600 hnfsam.

HNF_ACCESS

DISPLAY In real HW, access to memory location %{ADDR} routed through HN-F/HN-S with node ID = %{HNID}. Fields:

ADDR unsigned int

Address of memory location being accessed.

HNFID unsigned int

NodeID of home cache node transaction is routed to.

HNSAM_Target_SNF

DISPLAY The target SNF for address (%{ADDRESS}) is (%{SNF_NUM}). Fields:

ADDRESS unsigned int

Transaction Address.

SNF_NUM unsigned int

SNF Node Index which is targeted from HNFSAM.

Interconnect_ABF_policy_reg

DISPLAY ABF policy register write details. Fields:

ABF_PR_WRITE string

Detail for policy register write.

Interconnect_do_abf_call

DISPLAY Cache state modelled %{CACHE_STATE_MODELLED} ABF commanded for addr %{ABF_REGION_START}-{ABF_REGION_END} with mode %{ABF_MODE} and . Fields:

ABF_MODE unsigned int

0 = CLEAN_INVALID, 1 = MAKE_INVALID, 2 = CLEAN_SHARED, 3 = RESERVED.

ABF_REGION_END unsigned int

End Address of ABF region.

ABF_REGION_START unsigned int

Start Address of ABF region.

CACHE_STATE_MODELLED bool

cache_state_modelled, true = yes, false = no.

LCN_ACCESS

DISPLAY In real HW, access to memory location %{ADDR} routed through LCN with node ID = %{LCN_NODEID}}. Fields:

ADDR unsigned int

Address of memory location being accessed.

LCN_NODEID unsigned int

Node Id of LCN node transation is routed to.

MemoryMapped_Read

Trace read from a memory-mapped register. Fields:

OFFSET unsigned int

Offset of address within the Interconnect.

PASpace string

Physical Address Space.

REG_NAME string

Name of the register being read.

VALUE unsigned int

Value returned.

MemoryMapped_Write

Trace write to a memory-mapped register. Fields:

OFFSET unsigned int

Offset of address within the Interconnect.

PASpace string

Physical Address Space.

REG_NAME string

Name of the register being written.

UPDATED_VALUE unsigned int

New value read back from the register; for write-only registers this will be equal to VALUE.

VALUE unsigned int

Value written.

OCM_Regions

DISPLAY Regions configured as OCM %{REGIONS}. Fields:

REGIONS string

memory regions allocated to OCM.

RAS_BEHAVIOR_TRACE

Trace to capture the RAS behavior. Fields:

RAS_INFO string

Gives information about the error detected.

RNSAM_Memory_Map

DISPLAY Memory Map of RNSAM %{MMAP}. Fields:

MMAP string

memory map of cmn600 rnsam.

RNSAM_Region_HTG_TgtID_Table

DISPLAY %{RNSAM} Region %{REGION_NUM} selected HTG TgtID Table %{TABLE_NAME}. Fields:

REGION_NUM unsigned int

Region Number in RNSAM.

RNSAM string

Name of the RNSAM registers where the region is configured.

TABLE_NAME string

Table storing Target ID's.

RNSAM_Target_HNF

DISPLAY The target HNF for address (%{ADDRESS}) is (%{HNF_NUM}). Fields:

ADDRESS unsigned int

Transaction Address.

HNF_NUM unsigned int

HNF Node Index to which the transaction is routed.

SNF_HASHING_Target_SNF

DISPLAY The target SNF for address (%{ADDRESS}) in default hashed region is (%{HASHED_SNF}). But the model is routing the transaction to SNF indicated by the HNSAM_Target_SNF trace. Fields:

ADDRESS unsigned int

Transaction Address.

SNF_NUM unsigned int

SNF Node Index which is targeted from HNSAM hashing for default region.

Warning.Hashing_enabled_for_cache_group

DISPLAY The SCG group (which contains HNF%{INDEX}) has different SNF default ports in member HNFs which is not supported. Fields:

HNF_INDEX unsigned int

HNF Node index which is part of system cache group.

Warning.MTU_MODE_Not_Unique

DISPLAY MTU node configuration should be same for all devices. The MTU node %{INDEX} in mode %{MODE} is different the current mode %{CURRENT_MODE}. Fields:

CURRENT_MODE enum

Execution mode for another device already configured.

INDEX unsigned int

MTU index being configured.

MODE enum

Execution mode being configured.

Warning.MTU_Shuttering_Reg_Not_Unique

DISPLAY MTU shuttering register value should be same for all devices. Register %{REG_INDEX} in MTU %{MTU_INDEX} with value %{VALUE} is different the current value %{CURRENT_VALUE}. Fields:

CURRENT_VALUE unsigned int

Shuttering register index already configured.

MTU_INDEX unsigned int

MTU index being configured.

REG_INDEX unsigned int

Register index being configured.

VALUE unsigned int

Shuttering register index being configured.

Warning.RNSAM_Not_Unique

DISPLAY RNSAM configuration should be same for all masters. The RNSAM for master %{INDEX} is different. Fields:

INDEX unsigned int

Index of RN component. Index starts with RNFs and then the RNIs.

Warning.event_cpu_events_disabled

DISPLAY Event not propagated because CPU Events are disabled for CCIX link %{INDEX}. Fields:

INDEX unsigned int

CCIX link index.

Warning.event_smp_mode_disabled

DISPLAY Event not propagated because SMP mode is not enabled.

Warning.rnsam_and_cxrasam_mismatch

DISPLAY RNSAM and CXRASAM address regions don't match. You may need to review its configuration.

event_propagation_downstream

DISPLAY Event changed from downstream (from other chips) from port: %{FROM_PORT} to state: %{STATE}. Fields:

FROM_PORT unsigned int

Activated port.

STATE unsigned int

State being transmitted.

event_propagation_downstream_routing

DISPLAY Event propagated from downstream (from other systems) from port: %{FROM_PORT} to link: %{LINK} state: %{STATE} sent to local clusters. Fields:

FROM_PORT unsigned int

Activated port.

LINK unsigned int

Link.

STATE unsigned int

State being transmitted.

event_propagation_upstream

DISPLAY Event changed from upstream (clusters) sent to another chip via port %{TO_PORT} on link %{LINK} state: %{STATE}. Fields:

LINK unsigned int

Link number.

STATE unsigned int

State being transmitted.

TO_PORT unsigned int

Target port.

trace_add_cxra_memory_region

DISPLAY CXRA%{INDEX} SAM memory region: [%{START}-%{END}] => HAID=%{HAID}.
Fields:

END unsigned int

End address of the memory region.

HAID unsigned int

Memory region target HAID.

INDEX unsigned int

CXRA/CCG_RA index.

START unsigned int

Start address of the memory region.

trace_add_memory_region

DISPLAY Add new memory region type %{TYPE}, range start : %{START}, and end: %{END}.
Fields:

END unsigned int

End address of the memory region.

START unsigned int

Start address of the memory region.

TYPE enum

Node type.

trace_hnd

DISPLAY The HND index for XP%{INDEX} is %{HND_INDEX}. Fields:

HND_INDEX unsigned int

HND index for the XP.

INDEX unsigned int

XP Index.

2.91 ClockDivider

This section describes the trace sources.

frequency

Notify when changes of frequency occur. Fields:

current_time_in_simulator_ticks unsigned int

The current time in simulator ticks (and not clock ticks).

input_frequency_Hz unsigned int

The input frequency for this clock source (truncated to an integer).

known_invalid_frequency bool

The input frequency was known to be invalid and so the output frequency is too.

Not all frequency sources will provide information on the validity of the frequency it advertises.

output_frequency_Hz unsigned int

The output frequency of this clock source (truncated to an integer).

rate

Notify when the rate of a ClockDivider has been changed. Fields:

current_time_in_simulator_ticks unsigned int

The current time in simulator ticks (and not clock ticks).

div unsigned int

Frequency Divider.

mul unsigned int

Frequency Multiplier.

2.92 DMA350

This section describes the trace sources.

2D

%{CHANNEL} {IS_READ:(Y|N)} %{ADDRESS} %{XSIZE} %{LINES}. Fields:

CHANNEL unsigned int

channel.

IS_READ bool

Read transfer.

NEXT ADDRESS unsigned int

address.

REMAINING LINES unsigned int

Number of remaining lines.

UPDATED XSIZE unsigned int

xsize.

AutoReload

Autoreload **%{CHANNEL} %{MESSAGE}**. Fields:

CHANNEL unsigned int
channel.

MESSAGE string
message.

BootAddress

Boot address %{ADDRESS} %{IS_ENABLED} %{MEM_ATTR} %{SHARE_ATTR}. Fields:

ADDRESS unsigned int
address.

IS_ENABLED bool
is enabled.

MEM_ATTR unsigned int
Memory attribute.

SHARE_ATTR unsigned int
shareability attribute.

CHANNEL_CONFIG

Channel %{CHANNEL} Configuration: XType = %{XTYPE} YType = %{YTYPE} Transfer size = %{TRANSIZE} Source address = %{SRCADDR} Source Xsize = %{SRCXSIZE} SourceX increment = %{SRCXADDRINC} Source Ysize = %{SRCYSIZE} Source Y address stride = %{SRCYADDRSTRIDE} Destination address = %{DESADDR} Destination Xsize = %{DESXSIZE} DestinationX increment = %{DESXADDRINC} Destination Ysize = %{DESYSIZE} Destination Y address stride = %{DESYADDRSTRIDE} Stream = %{USESTREAM} GPO = %{USEGPO} Regreloadtype = %{REGRELOADTYPE} Donetype = %{DONETYPE} Donepauseen = %{DONEPAUSEEN} Fill value = %{FILLVAL}. Fields:

CHANNEL unsigned int
Index of the channel configured by the following parameters.

DESADDR unsigned int
Destination transaction start address.

DESXADDRINC signed int
 $\text{Desaddr_next} = \text{DESADDR} + 2^{\text{TRANSIZE}} * \text{DESXADDRINC}.$

DESXSIZE unsigned int
Destination X transactions to be performed.

DESYADDRSTRIDE signed int
 $\text{DESADDR_next_line_base} = \text{DESADDR_line_base} + 2^{\text{TRANSIZE}} * \text{DESYADDRSTRIDE}.$

DESYSIZE unsigned int
Number of DESXSIZE lines to read.

DONEPAUSEEN bool
Whether or not the pause-on-done feature is enabled.

DONETYPE enum

When (if at all) to assert done.

FILLVAL unsigned int

Fill value, when the wrap type is configured to fill and the destination transaction number is bigger than the source.

REGRELOADTYPE enum

The type of the autoreload feature.

SRCADDR unsigned int

Source transaction start address.

SRCXADDRINC signed int

$\text{Srcaddr_next} = \text{SRCADDR} + 2^{\text{TRANSIZE}} * \text{SRCXADDRINC}$.

SRCXSIZE unsigned int

Source X transactions to be performed.

SRCYADDRSTRIDE signed int

$\text{SRCADDR_next_line_base} = \text{SRCADDR_line_base} + 2^{\text{TRANSIZE}} * \text{SRCYADDRSTRIDE}$.

SRCYSIZE unsigned int

Number of SRCXSIZE lines to read.

TRANSIZE unsigned int

Size of the transfer performed by the Channel in bytes.

USEGPO bool

Whether or not the GPO feature is used for this transfer.

USESTREAM bool

Whether or not the stream feature is used for this transfer.

XTYPE enum

Transfer type in the X dimension.

YTYPE enum

Transfer type in the Y dimension.

CHANNEL_ERROR_INFO

Channel error info. $\{\text{CHANNEL}\} \{\text{BUSERR}\} \{\text{CFGERR}\} \{\text{SRCTRIGINSELERR}\} \{\text{DESTRIGINSELERR}\} \{\text{TRIGOUTSELERR}\} \{\text{AXIRDRESPERR}\} \{\text{AXIWRRESPERR}\} \{\text{AXIRDPOISERR}\} \{\text{STRINTSTRBERR}\} \{\text{STRINOVERRUN}\} \{\text{STRINEARLYTERM}\} \{\text{LINKHDRERR}\} \{\text{REGVALERR}\} \{\text{CFGCONFLERR}\}$. Fields:

AXIRDPOISERR bool

AXI read poison error.

AXIRDRESPERR bool

AXI read error.

AXIWRRESPERR bool

AXI write error.

BUSERR bool

Bus error.

CFGCONFLERR bool

Configuration conflict error.

CFGERR bool

Configuration error.

CHANNEL unsigned int

Index of the channel with the following error info statuses.

DESTRIGINSELERR bool

DST Trigger In selection error.

LINKHDRERR bool

Command link header error.

REGVALERR bool

Configuration register error.

SRCTRIGINSELERR bool

SRC Trigger In selection error.

STRINEARLYTERM bool

Stream in early termination error.

STRINOVERRUN bool

Stream in overflow error.

STRINTSTRBERR bool

Stream in strobe error.

TRIGOUTSELERR bool

Trigger Out selection error.

CHANNEL_IRQ_CONFIG

Interface trace for Signal IRQ_CHANNEL %{CHANNEL} %{INT_DONE_ENABLED:(SET|CLEAR)} %{INT_ERROR_ENABLED:(SET|CLEAR)} %{INT_DISABLED_ENABLED:(SET|CLEAR)} %{INT_STOPPED_ENABLED:(SET|CLEAR)} %{INT_WAIT_SRC_TRIG_ENABLED:(SET|CLEAR)} %{INT_WAIT_DST_TRIG_ENABLED:(SET|CLEAR)} %{INT_WAIT_OUT_TRIG_ACK_ENABLED:(SET|CLEAR)}. Fields:

CHANNEL unsigned int

Index of the channel with the following interrupt statuses.

INT_DISABLED_ENABLED bool

Disabled status interrupt enable.

INT_DONE_ENABLED bool

Done status interrupt enable.

INT_ERROR_ENABLED bool

Error status interrupt enable.

INT_STOPPED_ENABLED bool

Stopped status interrupt enable.

INT_WAIT_DST_TRIG_ENABLED bool

Wait destination trigger status interrupt enable.

INT_WAIT_OUT_TRIG_ACK_ENABLED bool

Wait output trigger acknowledge status interrupt enable.

INT_WAIT_SRC_TRIG_ENABLED bool

Wait source trigger status interrupt enable.

CHANNEL_IRQ_STATUS

Interface trace for Signal IRQ_CHANNEL %{CHANNEL} %{STAT_DONE:(SET|CLEAR)}
 %{STAT_ERR:(SET|CLEAR)} %{STAT_DISABLED:(SET|CLEAR)} %{STAT_STOPPED:
 (SET|CLEAR)} %{STAT_PAUSED:(SET|CLEAR)} %{STAT_RESUMEWAIT:(SET|CLEAR)}
 %{STAT_SRCTRIGINWAIT:(SET|CLEAR)} %{STAT_DESTRIGINWAIT:(SET|CLEAR)}
 %{STAT_TRIGOUTACKWAIT:(SET|CLEAR)}. Fields:

CHANNEL unsigned int

Index of the channel with the following interrupt statuses.

STAT_DESTRIGINWAIT bool

Wait destination trigger status.

STAT_DISABLED bool

Disabled status.

STAT_DONE bool

Done status.

STAT_ERR bool

Error status.

STAT_PAUSED bool

Paused status.

STAT_RESUMEWAIT bool

Resume wait status.

STAT_SRCTRIGINWAIT bool

Wait source trigger status.

STAT_STOPPED bool

Stopped status.

STAT_TRIGOUTACKWAIT bool

Wait output trigger acknowledge status.

CLK

Interface trace for CLK.

ChannelCommand

Channel Command. %{CHANNEL} %{COMMAND}. Fields:

CHANNEL unsigned int

channel.

COMMAND string

Command.

ChannelCommandLink

Channel Command Link. %{CHANNEL} %{DATA} %{MESSAGE}. Fields:

CHANNEL unsigned int

channel.

DATA unsigned int

data.

MESSAGE string

message.

ChannelThreadState

Channel Thread State. %{CHANNEL} %{THREAD_STATE} {IS_READ:(Y|N)} %{MESSAGE}. Fields:

CHANNEL unsigned int

channel.

IS_READ bool

Was called from src fiber.

MESSAGE string

a message in text.

THREAD_STATE enum

Thread state of the current channel.

ChannelTransfers

Channel Transfers. %{CHANNEL} %{ADDRESS} {IS_READ} {DATA} {READS_RECEIVED} {WRITES_SENT}. Fields:

ADDRESS unsigned int

Address.

CHANNEL unsigned int

channel.

DATA unsigned int

Data.

IS_READ bool

Read transfer.

READS_RECEIVED unsigned int

Number of reads received so far in the transfer.

WRITES_SENT unsigned int

Number of writes sent so far in the transfer.

GPO

Interface trace for GPO Signals GPO %{CHANNEL_INDEX} %{GPO_VALUE}. Fields:

CHANNEL_INDEX unsigned int

Channel Index.

GPO_VALUE unsigned int

Bit-number.

HW_TRIG_IN_REQ

Interface trace for HW TRIG_IN Signal %{PORT} %{VALUE}, %{REQ_TYPE}. Fields:

PORT unsigned int

Trigger port.

REQ_TYPE enum

REQ Type.

VALUE bool

Whether or not it's a high request.

HW_TRIG_OUT_ACK

Interface trace for hardware TRIG_OUT Acknowledgement %{PORT} %{STATUS:(SET|CLEAR)}. Fields:

PORT unsigned int

Trigger port.

STATUS bool

True : Signal Set, Clear : Signal Clear.

INTERNAL_TRIG_ACK

Internal Trigger Acknowledge. %{OUT_CHANNEL} %{IN_CHANNEL} %{IS_SRC_TRIG}. Fields:

IN_CHANNEL unsigned int

Index of the channel for which the trigger is an input.

IS_SRC_TRIG bool

Whether or not it's a src trigger.

OUT_CHANNEL unsigned int

Index of the channel for which the trigger is an output.

INTERNAL_TRIG_CONFIG

Internal Trigger Configuration. %{OUT_CHANNEL} %{IN_CHANNEL} %{IS_SRC_TRIG}. Fields:

IN_CHANNEL unsigned int

Index of the channel for which the trigger is an input.

IS_SRC_TRIG bool

Whether or not it's a src trigger.

OUT_CHANNEL unsigned int

Index of the channel for which the trigger is an output.

INTERNAL_TRIG_REQ

Internal Trigger Request. %{OUT_CHANNEL} %{IN_CHANNEL} %{IS_SRC_TRIG}. Fields:

IN_CHANNEL unsigned int

Index of the channel for which the trigger is an input.

IS_SRC_TRIG bool

Whether or not it's a src trigger.

OUT_CHANNEL unsigned int

Index of the channel for which the trigger is an output.

IRQ_CHANNEL

Interface trace for Signal IRQ_CHANNEL %{INDEX} %{STATUS:(SET|CLEAR)}. Fields:

INDEX unsigned int

Bit-number.

STATUS bool

True : Signal Set, Clear : Signal Clear.

IRQ_COMB_NONSEC

Interface trace for Signal IRQ_COMB_NONSEC %{STATUS:(SET|CLEAR)}. Fields:

STATUS bool

True : Signal Set, Clear : Signal Clear.

IRQ_COMB_NONSEC_ERR

Interface trace for Signal IRQ_COMB_NONSEC_ERR %{STATUS:(SET|CLEAR)}. Fields:

STATUS bool

True : Signal Set, Clear : Signal Clear.

IRQ_COMB_SEC

Interface trace for Signal IRQ_COMB_SEC %{STATUS:(SET|CLEAR)}. Fields:

STATUS bool

True : Signal Set, Clear : Signal Clear.

IRQ_COMB_SEC_ERR

Interface trace for Signal IRQ_COMB_SEC_ERR %{STATUS:(SET|CLEAR)}. Fields:

STATUS bool

True : Signal Set, Clear : Signal Clear.

IRQ_SEC_VIOL_ERR

Interface trace for Signal IRQ_SEC_VIOL_ERR %{STATUS:(SET|CLEAR)}. Fields:

STATUS bool

True : Signal Set, Clear : Signal Clear.

MISC_CH_ENABLED

Interface trace for MISC Signal CH_ENABLED %{INDEX} %{STATUS:(SET|CLEAR)}. Fields:

INDEX unsigned int

Bit-number.

STATUS bool

True : Signal Set, Clear : Signal Clear.

MISC_CH_ERR

Interface trace for MISC Signal CH_ERR %{INDEX} %{STATUS:(SET|CLEAR)}. Fields:

INDEX unsigned int

Bit-number.

STATUS bool

True : Signal Set, Clear : Signal Clear.

MISC_CH_NONSEC

Interface trace for MISC Signal CH_NONSEC %{INDEX} %{STATUS:(SET|CLEAR)}. Fields:

INDEX unsigned int

Bit-number.

STATUS bool

True : Signal Set, Clear : Signal Clear.

MISC_CH_PAUSED

Interface trace for MISC Signal CH_PAUSED %{INDEX} %{STATUS:(SET|CLEAR)}. Fields:

INDEX unsigned int

Bit-number.

STATUS bool

True : Signal Set, Clear : Signal Clear.

MISC_CH_PRIV

Interface trace for MISC Signal CH_PRIV %{INDEX} %{STATUS:(SET|CLEAR)}. Fields:

INDEX unsigned int

Bit-number.

STATUS bool

True : Signal Set, Clear : Signal Clear.

MISC_CH_STOPPED

Interface trace for MISC Signal CH_STOPPED %{INDEX} %{STATUS:(SET|CLEAR)}. Fields:

INDEX unsigned int

Bit-number.

STATUS bool

True : Signal Set, Clear : Signal Clear.

PVBUS_M0

Interface trace for PVBUS master PVBUS_M0 %{STATUS:(OK|Fail)}. Fields:

ADDR unsigned int

Address.

IS_READ bool

Read access request.

SIZE unsigned int

SIZE.

STATUS bool

Memory access status.

PVBUS_M1

Interface trace for PVBUS master PVBUS_M1 %{STATUS:(OK|Fail)}. Fields:

ADDR unsigned int

Address.

IS_READ bool

Read access request.

SIZE unsigned int

SIZE.

STATUS bool

Memory access status.

PVBUS_S

Interface trace for PVBUS Slave(Device) PVBUS_S %{STATUS:(OK|Fail)}. Fields:

ADDR unsigned int

Address.

IS_READ bool

Read access request.

SIZE unsigned int

SIZE.

STATUS bool

Memory access status.

PVBUS_STREAM

Interface trace for Stream %{CHANNEL} %{STREAM_IN:(YES|NO)} %{SUCCESS:(TRUE|FALSE)} %{SIZE} %{TLAST} %{LOW_DATA} %{HIGH_DATA}. Fields:

CHANNEL unsigned int

Channel number.

HIGH_DATA unsigned int

The higher 64 bits of the transaction.

LOW_DATA unsigned int

The lower 64 bits of the transaction.

SIZE unsigned int

Size in bytes.

STREAM_IN bool

Whether it's a Stream in or Stream out.

SUCCESS bool

Whether the transaction was successful or not.

TLAST bool

Signal indicating this is the last transaction.

RST_N

Interface trace for Signal RST_N %{STATUS:(SET|CLEAR)}. Fields:

STATUS bool

True : Signal Set, Clear : Signal Clear.

RegisterRead

Register read. %{SECURE:(S|NS)} register read of %{NAME} at %{OFFSET}, returns %{DATA} is %{STATUS:(OK|IGNORE)}. Fields:

DATA unsigned int

The data returned by the component.

OFFSET unsigned int

Register access offset.

PRIVILEGED bool

True when the access is privileged access.

REGSTRING string

The text description of the register value.

SECURE bool

True when the access is a secure access.

STATUS bool

status of register access, true : OK, false : ignore.

RegisterWrite

Register write. %{SECURE:(S|NS)} register write of %{NAME} at %{DATA} to %{OFFSET} updated value %{UPDATED} is %{STATUS:(OK|IGNORE)}. Fields:

DATA unsigned int

The data written to the register.

OFFSET unsigned int

Register access offset.

PRIVILEGED bool

True when the access is privileged access.

REGSTRING string

The text description of the register value.

SECURE bool

True when the access is a secure access.

STATUS bool

Register of register access.

UPDATED unsigned int

Updated value of register.

RunLog

Run Log. %{TYPE:(Error|Warning)} : %{MSG}. Fields:

MSG string

Run log message.

TYPE bool

Error or Warning Message.

SW_TRIG_IN_REQ

Interface trace for HW TRIG_IN Signal %{CH_NUM} %{REQ_TYPE}. Fields:

CH_NUM unsigned int

Channel number.

REQ_TYPE enum

REQ Type.

SW_TRIG_OUT_ACK

Interface trace for software TRIG_OUT Acknowledgement %{CH_NUM}. Fields:

CH_NUM unsigned int

Channel number.

TMPLT

Tmplt feature at %{CHANNEL} %{IS_SRC:(YES|NO)} skipped %{JUMP} addresses, new bitpos %{BITPOS}. Fields:

BITPOS unsigned int

The new bit position in the template.

CHANNEL unsigned int

channel.

IS_SRC bool

Whether or not this is the src side.

JUMP unsigned int

Number of skipped addresses.

TRIG_IN

Interface trace for Signal TRIG_IN Request Type %{CH_NUM} %{REQ} %{REQ_TYPE} %{SOURCE}. Fields:

CH_NUM unsigned int

Channel number.

IS_SRC bool

Whether or not it's a source trigger.

REQ_TYPE enum

REQ Type.

TYPE enum

Trigger type.

TRIG_IN_ACK

Interface trace for Signal TRIG_IN Acknowledgement Type %{PORT} %{ACK} %{ACK_TYPE}. Fields:

ACK bool

Ack.

ACK_TYPE enum

ACK Type.

PORT unsigned int

Trigger port.

TRIG_IN_CONFIG

Trigger In Configuration. %{PORT} %{TRIG_IN_TYPE} %{TRIG_IN_MODE} %{IS_SRC_TRIG} %{BLK_SIZE} %{CONNECTED_channel}. Fields:

BLK_SIZE unsigned int

Block size in case of BLOCK requests.

CONNECTED_channel unsigned int

Connected Channel Index.

IS_SRC_TRIG bool

Whether or not it's a src trigger.

PORT unsigned int

Trigger In Port.

TRIG_IN_MODE enum

Trigger in mode.

TRIG_IN_TYPE enum

Trigger in type.

TRIG_IN_INVALID_REQ

Interface trace for when a trigin req can't be accepted %{INDEX} %{IS_SRC_TRIG:(SRC|DES)}
%{MSG}. Fields:

INDEX unsigned int

Channel number.

IS_SRC_TRIG bool

Whether or not it was a src trigger.

MSG string

The reason for not accepting the request.

TRIG_OUT

Interface trace for Signal TRIG_OUT %{PORT} %{STATUS:(SET|CLEAR)}. Fields:

PORT unsigned int

Trigger port.

STATUS bool

True : Signal Set, Clear : Signal Clear.

TRIG_OUT_CONFIG

Trigger Out Configuration. %{PORT} %{CHANNEL}. Fields:

CHANNEL unsigned int

Connected Channel Index.

PORT unsigned int

Trigger Out Port.

2.93 DMC-500

This section describes the trace sources.

DMC_AccessInResetState

Register accessed while in reset state. DISPLAY DMC Register access in reset state.

DMC_AccessToUnimplementedRegister

Accessed an unimplemented register. DISPLAY DMC Register %{READ:(WRITE|READ)} to address: %{ADDR} FAILED as the register is unimplemented. Fields:

ADDR unsigned int

Address of the accessed register.

READ bool

Transaction type is read or write.

DMC_BlockingTransactions

Blocking transaction detected. DISPLAY DMC BLOCKED TRANSACTION. TYPE: %{READ:(WRITE|READ)} ADDRESS: %{ADDR} NS: %{NS:(SECURE|NONSECURE)} MANAGERID: %{MANAGER_ID}. Fields:

ADDR unsigned int

Address for the transaction.

MANAGER_ID unsigned int

Which manager initiated the transaction.

NS bool

Whether the transaction is secure or non-secure.

READ bool

Transaction type is read or write.

DMC_CheckPermissions

Permission check performed. DISPLAY DMC %{SUCCESS:(FAILED|SUCCESS)} TYPE: %{READ:(WRITE|READ)} ADDRESS: %{ADDR} NS: %{NS:(SECURE|NONSECURE)} MANAGERID: %{MANAGER_ID} OUTSIDE_DEFAULT: %{OUTSIDE_DEFAULT:(NO|YES)} TZ_FAIL: %{TZ_FAIL:(NO|YES)}. Fields:

ADDR unsigned int

Transaction address.

MANAGER_ID unsigned int

Which manager initiated the transaction.

NS bool

Whether the transaction is secure or non-secure.

OUTSIDE_DEFAULT bool

Whether the access failed due to outside default region.

READ bool

Transaction type is read or write.

SUCCESS bool

Whether transaction succeeded or failed.

TZ_FAIL bool

Whether the access failed due to invalid permissions.

DMC_ReadToWriteOnlyRegister

Read to a write only register. DISPLAY DMC Register Read to address: %{ADDR} FAILED as the register is write-only. Fields:

ADDR unsigned int

Address of the accessed register.

DMC_RegRead

Register read. DISPLAY DMC Register Offset: %{REG_OFFSET} Value: %{VALUE} RegName: %{REG_NAME}. Fields:

REG_NAME string

The name of the register read.

REG_OFFSET unsigned int

The address of the register read.

VALUE unsigned int

Read Value from the register.

DMC_RegWrite

Register written. DISPLAY DMC Register Offset: %{REG_OFFSET} Updated: From value %{VALUE} To value %{UPDATED_VALUE} RegName: %{REG_NAME}. Fields:

REG_NAME string

The name of the register updated.

REG_OFFSET unsigned int

The address of the register updated.

UPDATED_VALUE unsigned int

New Value in the register.

VALUE unsigned int

Old Value in the register.

DMC_RestrictedAccessToRegister

Register access restricted and not allowed in current DMC state. DISPLAY DMC Register Offset: %{REG_OFFSET} Access Type: %{READ:(WRITE|READ)} RegName: %{REG_NAME} Access Restrictions %{REG_RESTRICTIONS} DMC Arch State %{DMC_STATE}. Fields:

DMC_STATE unsigned int

Architectural state of the DMC.

READ bool

Access type is read or write.

REG_NAME string

The name of the register updated.

REG_OFFSET unsigned int

The address of the register accessed.

REG_RESTRICTIONS unsigned int

Access restrictions mask of the register.

DMC_WriteToReadOnlyRegister

Write to a read only register. DISPLAY DMC Register Write to address: %{ADDR} FAILED as the register is read-only. Fields:

ADDR unsigned int

Address of the accessed register.

2.94 DMC-520

This section describes the trace sources.

DMC520_ArchStateUpdate

Attempt to change architectural state due to command. DISPLAY DMC520 %{SUCCESS: (FAILED|SUCCESS)} CMD Rcvd: %{CMD} CURRENT_STATE: %{CURRENT_STATE} NEW_STATE: %{NEW_STATE}. Fields:

CMD enum

Command received to process.

CURRENT_STATE enum

State before the command processed.

NEW_STATE enum

Final state after command processed.

SUCCESS bool

Whether succeeded or failed.

DMC_AccessInResetState

Register accessed while in reset state. DISPLAY DMC Register access in reset state.

DMC_AccessToUnimplementedRegister

Accessed an unimplemented register. DISPLAY DMC Register %{READ:(WRITE|READ)} to address: %{ADDR} FAILED as the register is unimplemented. Fields:

ADDR unsigned int

Address of the accessed register.

READ bool

Transaction type is read or write.

DMC_BlockingTransactions

Blocking transaction detected. DISPLAY DMC BLOCKED TRANSACTION. TYPE: %{READ:(WRITE|READ)} ADDRESS: %{ADDR} NS: %{NS:(SECURE|NONSECURE)} MANAGERID: %{MANAGER_ID}. Fields:

ADDR unsigned int

Address for the transaction.

MANAGER_ID unsigned int

Which manager initiated the transaction.

NS bool

Whether the transaction is secure or non-secure.

READ bool

Transaction type is read or write.

DMC_CheckPermissions

Permission check performed. DISPLAY DMC %{SUCCESS:(FAILED|SUCCESS)} TYPE: %{READ:(WRITE|READ)} ADDRESS: %{ADDR} NS: %{NS:(SECURE|NONSECURE)} MANAGERID: %{MANAGER_ID} OUTSIDE_DEFAULT: %{OUTSIDE_DEFAULT:(NO|YES)} TZ_FAIL: %{TZ_FAIL:(NO|YES)}. Fields:

ADDR unsigned int

Transaction address.

MANAGER_ID unsigned int

Which manager initiated the transaction.

NS bool

Whether the transaction is secure or non-secure.

OUTSIDE_DEFAULT bool

Whether the access failed due to outside default region.

READ bool

Transaction type is read or write.

SUCCESS bool

Whether transaction succeeded or failed.

TZ_FAIL bool

Whether the access failed due to invalid permissions.

DMC_ReadToWriteOnlyRegister

Read to a write only register. DISPLAY DMC Register Read to address: %{ADDR} FAILED as the register is write-only. Fields:

ADDR unsigned int

Address of the accessed register.

DMC_RegRead

Register read. DISPLAY DMC Register Offset: %{REG_OFFSET} Value: %{VALUE} RegName: %{REG_NAME}. Fields:

REG_NAME string

The name of the register read.

REG_OFFSET unsigned int

The address of the register read.

VALUE unsigned int

Read Value from the register.

DMC_RegWrite

Register written. DISPLAY DMC Register Offset: %{REG_OFFSET} Updated: From value %{VALUE} To value %{UPDATED_VALUE} RegName: %{REG_NAME}. Fields:

REG_NAME string

The name of the register updated.

REG_OFFSET unsigned int

The address of the register updated.

UPDATED_VALUE unsigned int

New Value in the register.

VALUE unsigned int

Old Value in the register.

DMC_RestrictedAccessToRegister

Register access restricted and not allowed in current DMC state. DISPLAY DMC Register Offset: %{REG_OFFSET} Access Type: %{READ:(WRITE|READ)} RegName: %{REG_NAME} Access Restrictions %{REG_RESTRICTIONS} DMC Arch State %{DMC_STATE}. Fields:

DMC_STATE unsigned int

Architectural state of the DMC.

READ bool

Access type is read or write.

REG_NAME string

The name of the register updated.

REG_OFFSET unsigned int

The address of the register accessed.

REG_RESTRICTIONS unsigned int

Access restrictions mask of the register.

DMC_WriteToReadOnlyRegister

Write to a read only register. DISPLAY DMC Register Write to address: %{ADDR} FAILED as the register is read-only. Fields:

ADDR unsigned int

Address of the accessed register.

2.95 DMC-620

This section describes the trace sources.

DMC620_ArchStateUpdate

Attempt to change architectural state due to command. DISPLAY DMC620 %{SUCCESS: (FAILED|SUCCESS)} CMD Rcvd: %{CMD} CURRENT_STATE: %{CURRENT_STATE} NEW_STATE: %{NEW_STATE}. Fields:

CMD enum

Command received to process.

CURRENT_STATE enum

State before the command processed.

NEW_STATE enum

Final state after command processed.

SUCCESS bool

Whether succeeded or failed.

DMC620_DetectedRASError

DMC620 RAS error detected. DISPLAY DMC620 RAS_ERROR: %{RAS_ERROR}. Fields:

RAS_ERROR enum

Type of detected RAS error.

DMC620_Interrupt

DMC620 interrupt raised. DISPLAY DMC620 INTERRUPT_RAISED: %{INTERRUPT_TYPE}. Fields:

INTERRUPT_RAISED enum

Type of the raised interrupt.

DMC_AccessInResetState

Register accessed while in reset state. DISPLAY DMC Register access in reset state.

DMC_AccessToUnimplementedRegister

Accessed an unimplemented register. DISPLAY DMC Register %{READ:(WRITE|READ)} to address: %{ADDR} FAILED as the register is unimplemented. Fields:

ADDR unsigned int

Address of the accessed register.

READ bool

Transaction type is read or write.

DMC_BlockingTransactions

Blocking transaction detected. DISPLAY DMC BLOCKED TRANSACTION. TYPE: %{READ:(WRITE|READ)} ADDRESS: %{ADDR} NS: %{NS:(SECURE|NONSECURE)} MANAGERID: %{MANAGER_ID}. Fields:

ADDR unsigned int

Address for the transaction.

MANAGER_ID unsigned int

Which manager initiated the transaction.

NS bool

Whether the transaction is secure or non-secure.

READ bool

Transaction type is read or write.

DMC_CheckPermissions

Permission check performed. DISPLAY DMC %{SUCCESS:(FAILED|SUCCESS)} TYPE: %{READ:(WRITE|READ)} ADDRESS: %{ADDR} NS: %{NS:(SECURE|NONSECURE)} MANAGERID: %{MANAGER_ID} OUTSIDE_DEFAULT: %{OUTSIDE_DEFAULT:(NO|YES)} TZ_FAIL: %{TZ_FAIL:(NO|YES)}. Fields:

ADDR unsigned int

Transaction address.

MANAGER_ID unsigned int

Which manager initiated the transaction.

NS bool

Whether the transaction is secure or non-secure.

OUTSIDE_DEFAULT bool

Whether the access failed due to outside default region.

READ bool

Transaction type is read or write.

SUCCESS bool

Whether transaction succeeded or failed.

TZ_FAIL bool

Whether the access failed due to invalid permissions.

DMC_ReadToWriteOnlyRegister

Read to a write only register. DISPLAY DMC Register Read to address: %{ADDR} FAILED as the register is write-only. Fields:

ADDR unsigned int

Address of the accessed register.

DMC_RegRead

Register read. DISPLAY DMC Register Offset: %{REG_OFFSET} Value: %{VALUE} RegName: %{REG_NAME}. Fields:

REG_NAME string

The name of the register read.

REG_OFFSET unsigned int

The address of the register read.

VALUE unsigned int

Read Value from the register.

DMC_RegWrite

Register written. DISPLAY DMC Register Offset: %{REG_OFFSET} Updated: From value %{VALUE} To value %{UPDATED_VALUE} RegName: %{REG_NAME}. Fields:

REG_NAME string

The name of the register updated.

REG_OFFSET unsigned int

The address of the register updated.

UPDATED_VALUE unsigned int

New Value in the register.

VALUE unsigned int

Old Value in the register.

DMC_RestrictedAccessToRegister

Register access restricted and not allowed in current DMC state. DISPLAY DMC Register Offset: %{REG_OFFSET} Access Type: %{READ:(WRITE|READ)} RegName: %{REG_NAME} Access Restrictions %{REG_RESTRICTIONS} DMC Arch State %{DMC_STATE}. Fields:

DMC_STATE unsigned int

Architectural state of the DMC.

READ bool

Access type is read or write.

REG_NAME string

The name of the register updated.

REG_OFFSET unsigned int

The address of the register accessed.

REG_RESTRICTIONS unsigned int

Access restrictions mask of the register.

DMC_WriteToReadOnlyRegister

Write to a read only register. DISPLAY DMC Register Write to address: %{ADDR} FAILED as the register is read-only. Fields:

ADDR unsigned int

Address of the accessed register.

2.96 DSU

This section describes the trace sources.

PSTATE_PACTIVE_OPMODE_TRANSITION

Trace pactive and Opmode mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Operating mode after the transition.

OLD_STATE enum

Operating mode before the transition.

PSTATE_PACTIVE_POWER_TRANSITION

Trace pactive and power mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Power mode after the transition.

OLD_STATE enum

Power mode before the transition.

2.97 DSU-110

This section describes the trace sources.

PSTATE_PACTIVE_OPMODE_TRANSITION

Trace pactive and Opmode mode transitions on PChannel. Fields:

DEVPACTIVE unsigned int

Integer representation of DEVPACTIVE[] signals.

NEW_STATE enum

Operating mode after the transition.

OLD_STATE enum

Operating mode before the transition.

PSTATE_PACTIVE_OPMODE_TRANSITION_DENIED

Trace the denied pactive and Opmode mode transitions on PChannel. Fields:

DEVPACTIVE **unsigned int**

Integer representation of DEVPACTIVE[] signals.

NEW_STATE **enum**

Operating mode after the transition.

OLD_STATE **enum**

Operating mode before the transition.

REASON **string**

Reason for the transition being denied.

PSTATE_PACTIVE_POWER_TRANSITION

Trace pactive and power mode transitions on PChannel. Fields:

DEVPACTIVE **unsigned int**

Integer representation of DEVPACTIVE[] signals.

NEW_STATE **enum**

Power mode after the transition.

OLD_STATE **enum**

Power mode before the transition.

PSTATE_PACTIVE_POWER_TRANSITION_DENIED

Trace the denied pactive and power mode transitions on PChannel. Fields:

DEVPACTIVE **unsigned int**

Integer representation of DEVPACTIVE[] signals.

NEW_STATE **enum**

Power mode after the transition.

OLD_STATE **enum**

Power mode before the transition.

REASON **string**

Reason for the transition being denied.

dsu.set_defaulttmp

DISPLAY DEFAULTTMP port is set to %{value}. Fields:

value **bool**

Value driven on DEFAULTTMP pin.

dsu.set_peripheral_range

DISPLAY %{isStart:AENDMPP|ASTARTMPP}%{index} is set to %{isStart:%{end_address}}%{start_address}}, so that valid peripheral address range becomes [%{start_address}, %{end_address})). Fields:

end_address unsigned int

End address of this peripheral region.

index unsigned int

Index of ASTARTMP/AENDMP peripheral port. Can be between 0 and 3.

isStart bool

If ASTARTMP is set, true, false otherwise.

start_address unsigned int

Start address of this peripheral region.

dsu.utility_bus.access

DISPLAY %{is_debug:Normal|Debug} %{is_read:write|read} access has been made to UtilityBus to access %{component_accessed} %{access_is_to_cluster_reg:to %{target_core} core}. Fields:

access_is_to_cluster_reg bool

true if this access is against cluster registers, false otherwise.

component_accessed string

component to which this transaction is made.

is_debug bool

true if the access is a debug one, false otherwise.

is_read bool

true if the access is a read one, false otherwise.

offset unsigned int

Offset to the utility bus base address used to identify which register is being accessed.

target_core unsigned int

is the target core if the transaction is to core register.

dsu.utility_bus.access_is_razwi

DISPLAY %{is_read:Write|Read} access has been made to UtilityBus with offset %{offset}, but this access results in **RAZWI** because %{msg}. Fields:

is_read bool

true if the access is a read one, false otherwise.

msg string

the reason why this access is **RAZWI**.

offset unsigned int

Offset to the base address of utility bus used to identify which register is being accessed.

2.98 DSU-120

This section describes the trace sources.

PSTATE_PACTIVE_OPMODE_TRANSITION

Trace pactive and Opmode mode transitions on PChannel. Fields:

DEVPACTIVE **unsigned int**

Integer representation of DEVPACTIVE[] signals.

NEW_STATE **enum**

Operating mode after the transition.

OLD_STATE **enum**

Operating mode before the transition.

PSTATE_PACTIVE_OPMODE_TRANSITION_DENIED

Trace the denied pactive and Opmode mode transitions on PChannel. Fields:

DEVPACTIVE **unsigned int**

Integer representation of DEVPACTIVE[] signals.

NEW_STATE **enum**

Operating mode after the transition.

OLD_STATE **enum**

Operating mode before the transition.

REASON **string**

Reason for the transition being denied.

PSTATE_PACTIVE_POWER_TRANSITION

Trace pactive and power mode transitions on PChannel. Fields:

DEVPACTIVE **unsigned int**

Integer representation of DEVPACTIVE[] signals.

NEW_STATE **enum**

Power mode after the transition.

OLD_STATE **enum**

Power mode before the transition.

PSTATE_PACTIVE_POWER_TRANSITION_DENIED

Trace the denied pactive and power mode transitions on PChannel. Fields:

DEVPACTIVE **unsigned int**

Integer representation of DEVPACTIVE[] signals.

NEW_STATE **enum**

Power mode after the transition.

OLD_STATE enum

Power mode before the transition.

REASON string

Reason for the transition being denied.

dsu.set_defaulttmp

DISPLAY DEFAULTTMP port is set to %{value}. Fields:

value bool

Value driven on DEFAULTTMP pin.

dsu.set_peripheral_range

DISPLAY %{isStart:AENDMP|ASTARTMP}[%{index}] is set to %{isStart:%{end_address}|%{start_address}}, so that valid peripheral address range becomes [%{start_address}, %{end_address}]. Fields:

end_address unsigned int

End address of this peripheral region.

index unsigned int

Index of ASTARTMP/AENDMP peripheral port. Can be between 0 and 3.

isStart bool

If ASTARTMP is set, true, false otherwise.

start_address unsigned int

Start address of this peripheral region.

dsu.utility_bus.access

DISPLAY %{is_debug:Normal|Debug} %{is_read:write|read} access has been made to UtilityBus to access %{component_accessed} %{access_is_to_cluster_reg:to %{target_core} core}. Fields:

access_is_to_cluster_reg bool

true if this access is against cluster registers, false otherwise.

component_accessed string

component to which this transaction is made.

is_debug bool

true if the access is a debug one, false otherwise.

is_read bool

true if the access is a read one, false otherwise.

offset unsigned int

Offset to the utility bus base address used to identify which register is being accessed.

target_core unsigned int

is the target core if the transaction is to core register.

dsu.utility_bus.access_is_razwi

DISPLAY %{is_read:Write|Read} access has been made to UtilityBus with offset %{offset}, but this access results in **RAZWI** because %{msg}. Fields:

is_read bool

true if the access is a read one, false otherwise.

msg string

the reason why this access is **RAZWI**.

offset unsigned int

Offset to the base address of utility bus used to identify which register is being accessed.

2.99 DualClusterSystemConfigurationBlock

This section describes the trace sources.

sw_trace_event

This is a trace event generate under SW control. Fields:

cluster_and_cpu_id unsigned int

The top byte represents the cluster id, the bottom byte the cpu id.

data0 unsigned int

SW defined data field 0.

data1 unsigned int

SW defined data field 1.

data2 unsigned int

SW defined data field 2.

message string

A message from the SW.

sequence_id unsigned int

This is a unique id per emitted trace event.

sw_trace_event2

This is a trace event generate under SW control. Fields:

cluster_and_cpu_id unsigned int

The top byte represents the cluster id, the bottom byte the cpu id.

data0 unsigned int

SW defined data field 0.

data1 unsigned int

SW defined data field 1.

data2 unsigned int

SW defined data field 2.

message string

A message from the SW.

sequence_id unsigned int

This is a unique id per emitted trace event.

sw_trace_event3

This is a trace event generate under SW control. Fields:

cluster_and_cpu_id unsigned int

The top byte represents the cluster id, the bottom byte the cpu id.

data0 unsigned int

SW defined data field 0.

data1 unsigned int

SW defined data field 1.

data2 unsigned int

SW defined data field 2.

message string

A message from the SW.

sequence_id unsigned int

This is a unique id per emitted trace event.

sw_trace_event4

This is a trace event generate under SW control. Fields:

cluster_and_cpu_id unsigned int

The top byte represents the cluster id, the bottom byte the cpu id.

data0 unsigned int

SW defined data field 0.

data1 unsigned int

SW defined data field 1.

data2 unsigned int

SW defined data field 2.

message string

A message from the SW.

sequence_id unsigned int

This is a unique id per emitted trace event.

2.100 EthosU55

This section describes the trace sources.

INFO_Irq

NPU IRQ signal %{IRQ_STATE:(SET|CLEAR)}. Fields:

IRQ_STATE bool

The IRQ state.

INFO_Read

NPU Register address: %{REG_ADDRESS} Value: %{VALUE}. Fields:

REG_ADDRESS unsigned int

The address of the register read.

VALUE unsigned int

Read Value from the register.

INFO_Reset

NPU Reset %{STATE:(START|END)}. Fields:

STATE bool

Reset state is start or end.

INFO_Write

NPU Register address: %{REG_ADDRESS} Value %{VALUE}. Fields:

REG_ADDRESS unsigned int

The address of the register write.

VALUE unsigned int

Write Value to the register.

2.101 EthosU65

This section describes the trace sources.

INFO_Irq

NPU IRQ signal %{IRQ_STATE:(SET|CLEAR)}. Fields:

IRQ_STATE bool

The IRQ state.

INFO_Read

NPU Register address: %{REG_ADDRESS} Value: %{VALUE}. Fields:

REG_ADDRESS unsigned int

The address of the register read.

VALUE unsigned int

Read Value from the register.

INFO_Reset

NPU Reset %{STATE:(START|END)}. Fields:

STATE bool

Reset state is start or end.

INFO_Write

NPU Register address: %{REG_ADDRESS} Value %{VALUE}. Fields:

REG_ADDRESS unsigned int

The address of the register write.

VALUE unsigned int

Write Value to the register.

2.102 EthosU85

This section describes the trace sources.

INFO_Irq

NPU IRQ signal %{IRQ_STATE:(SET|CLEAR)}. Fields:

IRQ_STATE bool

The IRQ state.

INFO_Read

NPU Register address: %{REG_ADDRESS} Value: %{VALUE}. Fields:

REG_ADDRESS unsigned int

The address of the register read.

VALUE unsigned int

Read Value from the register.

INFO_Reset

NPU Reset %{STATE:(START|END)}. Fields:

STATE bool

Reset state is start or end.

INFO_Write

NPU Register address: %{REG_ADDRESS} Value %{VALUE}. Fields:

REG_ADDRESS unsigned int

The address of the register write.

VALUE unsigned int

Write Value to the register.

2.103 FMU

This section describes the trace sources.

fmu_32bit_register_access

Trace fmu registers read write access. Fields:

Access string

.

Register string

.

Value unsigned int

.

fmu_64bit_register_access

Trace fmu registers read write access. Fields:

Access string

.

Register string

.

Value unsigned int

.

fmu_block_power_state_change

Trace power state of blocks %{Block}. Fields:

Block unsigned int

fmu block.

SigValue bool

power state signal value.

fmu_block_prot_mech

Trace protection enable/disable critical/non-critical. Fields:

BlkId unsigned int

.

BlkType unsigned int

.

ProtMech string

.

ProtMechId unsigned int

.

fmu_eri_cri_signal

Trace eri and cri signal. Fields:

CriSignal string

critical error send by FMU.

EriSignal string

non-critical error send by FMU.

fmu_info_log

Trace info messages. Fields:

INFO string

info messages.

fmu_inject_error

Trace fmu error injection. Fields:

BlkId unsigned int

.

BlkType unsigned int

.

Msg string

.

ProtMechId unsigned int

.

2.104 Firewall

This section describes the trace sources.

ArchMsg.Info.IgnoredRead

Register read in reserved area. DISPLAY %{NSECURE:(S|NS)} read at %{OFFSET} attempted, ignored: reserved area. Fields:

NSECURE bool

True when the access is not a secure access.

OFFSET unsigned int

Access offset.

ArchMsg.Info.IgnoredWrite

Register write in reserved area. DISPLAY %{NSECURE:(S|NS)} write of %{DATA} to %{OFFSET} attempted, ignored: reserved area. Fields:

DATA unsigned int

The data written to the address.

NSECURE bool

True when the access is not a secure access.

OFFSET unsigned int

Access offset.

ArchMsg.Info.Initialised

Initialised. DISPLAY Initialised.

ArchMsg.Info.ReadFromWriteOnlyRegister

Register read of write only register. DISPLAY %{NSECURE:(S|NS)} read at %{OFFSET} attempted, ignored: write-only register. Fields:

NSECURE bool

True when the access is not a secure access.

OFFSET unsigned int

Access offset.

ArchMsg.Info.ReadInReset

Register read during reset. DISPLAY %{NSECURE:(S|NS)} read at %{OFFSET} attempted during reset. Fields:

NSECURE bool

True when the access is not a secure access.

OFFSET unsigned int

Access offset.

ArchMsg.Info.RegisterRead

Register read. DISPLAY %{NSECURE:(S|NS)} register read at %{OFFSET}, returns %{DATA}. Fields:

DATA unsigned int

The data returned by the component.

NSECURE bool

True when the access is not a secure access.

OFFSET unsigned int

Register access offset.

ArchMsg.Info.RegisterWrite

Register write. DISPLAY %{NSECURE:(S|NS)} register write of %{DATA} to %{OFFSET}. Fields:

DATA unsigned int

The data written to the register.

NSECURE bool

True when the access is not a secure access.

OFFSET unsigned int

Register access offset.

ArchMsg.Info.Resetting

Resetting. DISPLAY Resetting.

ArchMsg.Info.UnalignedAccessRegister

Unaligned access to register. DISPLAY unaligned %{WRITE:(W|R)} access to %{NSECURE:(S|NS)} %{OFFSET} attempted. Fields:

NSECURE bool

True when the access is not a secure access.

OFFSET unsigned int

Access offset.

WRITE bool

Write access.

ArchMsg.Info.WriteInReset

Register write during reset. DISPLAY %{NSECURE:(S|NS)} write of %{DATA} to %{OFFSET} attempted during reset. Fields:

DATA unsigned int

The data written to the address.

NSECURE bool

True when the access is not a secure access.

OFFSET unsigned int

Access offset.

ArchMsg.Info.WriteToReadOnlyRegister

Register write of read only register. DISPLAY %{NSECURE:(S|NS)} write of %{DATA} to %{OFFSET} attempted, ignored: read-only register. Fields:

DATA unsigned int

The data written to the address.

NSECURE bool

True when the access is not a secure access.

OFFSET unsigned int

Access offset.

ArchMsg.Info.configurationAccessError

Configuration access error . DISPLAY Configuration Error in component with ID:%{ID}. %{MSG}. Fields:

ID unsigned int

FW Component/Controller ID.

MSG string

Debug message.

ArchMsg.Info.debugTrace

Verbose debugging messages during important code sequences. DISPLAY %{MSG}. Fields:

MSG string

Debug message.

ArchMsg.Info.programmingError

Programming Error. DISPLAY Programming Error in component with ID:%{ID}. %{MSG}. Fields:

ID unsigned int

FW Component/Controller ID.

MSG string

Debug message.

ArchMsg.Info.transactionTrace

Super Verbose traces to print updated transaction attributes DISPLAY %{MSG}. Fields:

MSG string

Debug message.

2.105 GIC_400

This section describes the trace sources.

debug_out

Debug trace from the VGIC model, this contains implementation detail that is unlikely to be generally useful. Fields:

message string

The message from the VGIC.

sequence_id unsigned int

Each successive message is labeled with an increasing number.

gic_log_errors_out

Errors from the VGIC model. Fields:

message string

The message from the VGIC.

sequence_id unsigned int

Each successive message is labeled with an increasing number.

gic_log_fatal_out

Fatal error from the VGIC model. Fields:

message string

The message from the VGIC.

gic_log_warnings_out

Warnings from the VGIC model. Fields:

message string

The message from the VGIC.

sequence_id unsigned int

Each successive message is labeled with an increasing number.

vgic_cfgsdisable

The VGIC CFGSDISABLE pin has changed its state, this (un)locks down some configuration registers. Fields:

state bool

The state of the signal.

vgic_distributor_register_access

Distributor interface accesses on the VGIC. Fields:

cpu_id unsigned int

The CPU that is making the request.

data unsigned int

The data in the access aligned to the LSB.

is_read enum

Is the access a read?.

offset enum

Register within interface, top bit indicates NS state.

offset_in_word unsigned int

The offset of the access in the word.

size_in_bytes unsigned int

The size of the access in bytes.

vgic_fiq_out

The FIQ output signal to the core. Fields:

cpu_id unsigned int

The CPU id.

state bool

The state of the signal.

vgic_hypervisor_register_access

Hypervisor interface register accesses. Fields:

cpu_id unsigned int

The CPU that is making the request.

data unsigned int

The data in the access aligned to the LSB.

is_read enum

Is the access a read?.

offset enum

Register within interface, top bit indicates NS state.

offset_in_word unsigned int

The offset of the access in the word.

size_in_bytes unsigned int

The size of the access in bytes.

vgic_irq_out

The IRQ output signal to the core. Fields:

cpu_id unsigned int

The CPU id.

state bool

The state of the signal.

vgic_legacy_fiq_in

The legacy FIQ input signal into the VGIC. Fields:

cpu_id unsigned int

The CPU id. There is one legacy IRQ/FIQ per core.

state bool

The state of the signal.

vgic_legacy_irq_in

The legacy IRQ input signal into the VGIC. Fields:

cpu_id unsigned int

The CPU id. There is one legacy IRQ/FIQ per core.

state bool

The state of the signal.

vgic_physical_register_access

Physical interface accesses on the VGIC. Fields:

cpu_id unsigned int

The CPU that is making the request.

data unsigned int

The data in the access aligned to the LSB.

is_read enum

Is the access a read?.

offset enum

Register within interface, top bit indicates NS state.

offset_in_word unsigned int

The offset of the access in the word.

size_in_bytes unsigned int

The size of the access in bytes.

vgic_reset

The reset pin on the VGIC has been changed. Fields:

state bool

The state of the SPI signal.

vgic_spi

Shared Peripheral Interrupt signal changed. Fields:

id unsigned int

The interrupt id this will generate. SPIs start at ID 32 and so you will only see ids of ≥ 32 .

state bool

The state of the SPI signal.

vgic_vfiq_out

The VFIQ output signal to the core. Fields:

cpu_id unsigned int

The CPU id.

state bool

The state of the signal.

vgic_virq_out

The VIRQ output signal to the core. Fields:

cpu_id unsigned int

The CPU id.

state bool

The state of the signal.

vgic_virtual_machine_register_access

Virtual machine interface register accesses. Fields:

cpu_id unsigned int

The CPU that is making the request.

data unsigned int

The data in the access aligned to the LSB.

is_read enum

Is the access a read?.

offset enum

Register within interface, top bit indicates NS state.

offset_in_word unsigned int

The offset of the access in the word.

size_in_bytes unsigned int

The size of the access in bytes.

vgic_virtual_maintenance_interrupt

The VGIC is signaling that it needs software help from the hypervisor to maintain the VGIC interfaces. Fields:

cpu_id unsigned int

The CPU id.

state bool

The state of the signal.

vgic_wakeup_fiq

The wakeup FIQ signal. Fields:

cpu_id unsigned int

The CPU id.

state bool

The state of the signal.

vgic_wakeup_irq

The wakeup IRQ signal. Fields:

cpu_id unsigned int

The CPU id.

state bool

The state of the signal.

2.106 GICv3CPUInterface

This section describes the trace sources.

ArchMsg.Error.GICv3_UnknownDownstreamWrite

DownstreamWrite packet with unexpected format. DISPLAY Distributor sent a DownstreamWrite packet with unsupported format. Acknowledging and ignoring. This is a PROTOCOL ERROR, the architecture allows system behaviour to become UNPREDICATBLE from this point onwards.

ArchMsg.Warning.GICv3.Invalidv4Inv3onlyModePacket

GICv4 packet received by GICv3-only interface. DISPLAY GICv4 Packet %{PACKET} received by GICv3 only interface. Ignored, further behaviour UNPRED. Fields:

PACKET string

The name of the invalid packet.

ArchMsg.Warning.GICv3_AttemptedToDeactivateLPI

Attempt to deactivate an LPI. DISPLAY Attempted to deactivate an LPI (%{INTERRUPT_ID}) through write to IC%{VIRTUAL:C|V}_DIR_EL1. Fields:

INTERRUPT_ID unsigned int

Interrupt ID that the write attempted to end.

VIRTUAL bool

Access was to virtual interface.

ArchMsg.Warning.GICv3_AttemptedToDeactivateNonExistentInterrupt

Attempt to deactivate a non-existent interrupt. DISPLAY Attempted to deactivate an interrupt (%{INTERRUPT_ID}; CPUID = %{CPUID}) that hasn't had a corresponding write to EOI register. Sending deactivate to distributor anyway in case software is manually managing active bits. Fields:

CPUID unsigned int

CPU ID specified in end request.

INTERRUPT_ID unsigned int

Interrupt ID that the write attempted to end.

NS bool

Access was non-secure.

ArchMsg.Warning.GICv3_AttemptedToDeactivateSpuriousInterrupt

Attempt to deactivate a spurious interrupt. DISPLAY Attempted to deactivate a spurious interrupt (%{SPURIOUS_INTERRUPT_ID}; CPUID = %{CPUID}) through %{NS:(secure|non-secure)} write to %{GICC_REGISTER:(|G)}IC%{VIRTUAL:C|V}_DIR%{GICC_REGISTER:(_EL1)}}. Fields:

CPUID unsigned int

CPU ID (probably 0, but included for completeness).

GICC_REGISTER bool

Access was to GICC register (rather than ICC register).

NS bool

Access was non-secure.

SPURIOUS_INTERRUPT_ID unsigned int

Interrupt ID that the write attempted to end.

VIRTUAL bool

Access was to virtual interface.

ArchMsg.Warning.GICv3_AttemptedToEndNonExistentInterrupt

A write has attempted to end a non-existent interrupt. DISPLAY Attempted to end an interrupt (%{INTERRUPT_ID}; CPUID = %{CPUID}) that isn't waiting to be ended through %{NS:(secure|non-secure)} write to GIC%{VIRTUAL:(C|V)}_EOIR or GIC%{VIRTUAL:(C|V)}_AEOIR. Fields:

CPUID unsigned int

CPU ID specified in end request.

INTERRUPT_ID unsigned int

Interrupt ID that the write attempted to end.

NS bool

Access was non-secure.

VIRTUAL bool

Access was to virtual interface.

ArchMsg.Warning.GICv3_AttemptedToEndSpuriousInterrupt

A write has attempted to end a spurious interrupt. DISPLAY Attempted to end a spurious interrupt (%{SPURIOUS_INTERRUPT_ID}; CPUID = %{CPUID}) through %{NS:(secure|non-secure)} write to GIC%{VIRTUAL:(C|V)}_EOIR or GIC%{VIRTUAL:(C|V)}_AEOIR. Fields:

CPUID unsigned int

CPU ID (probably 0, but included for completeness).

NS bool

Access was non-secure.

SPURIOUS_INTERRUPT_ID unsigned int

Interrupt ID that the write attempted to end.

VIRTUAL bool

Whether the access was virtual.

ArchMsg.Warning.GICv3_AttemptedToWriteDIRRegisterWhenEOImodeNotSet

Unexpected write to DIR register when EOImode not set. DISPLAY %{NS:(S|Non-s)}ecure write to %{GICC_REGISTER:(|G)}IC%{VIRTUAL:(C|V)}_DIR%{GICC_REGISTER:(_EL1|)} register is unexpected: attempting to deactivate interrupt ID %{INTERRUPT_ID} (CPU ID %{CPUID}) as EOImode is not set (deactivate occurs as part of write to EOI register). Fields:

CPUID unsigned int

CPU ID (probably 0, but included for completeness).

GICC_REGISTER bool

Access was to GICC register (rather than ICC register).

INTERRUPT_ID unsigned int

Interrupt ID that the write attempted to deactivate.

NS bool

Access was non-secure.

VIRTUAL bool

Access was to virtual interface.

ArchMsg.Warning.GICv3_CFGSDISABLE_unsupported

DISPLAY CFGSDISABLE signal has been set but doing so has no effect.

ArchMsg.Warning.GICv3_CPUInterface.GICC.MemoryMapped_IgnoredRegisterAccessDueToViewMismatch

DISPLAY Attempt to access register %{REGISTER_NAME} from View %{VIEW_ID} is ignored, because this register is accessible only from View %{OWNER_VIEW_ID}. Fields:

OWNER_VIEW_ID unsigned int

The view owning the register.

REGISTER_NAME string

Name of the register being accessed.

VIEW_ID unsigned int

The view trying to access the register.

ArchMsg.Warning.GICv3_CPUInterface.GICC.MemoryMapped_ReadIgnored

DISPLAY GICv3 attempting to read location %{OFFSET} but ignored and RAZ. Fields:

OFFSET unsigned int

Offset of address within the GICv3 module.

ArchMsg.Warning.GICv3_CPUInterface.GICC.MemoryMapped_ReadReserved

DISPLAY GICv3 reserved location %{OFFSET} attempting to be read. Fields:

OFFSET unsigned int

Offset of address within the GICv3 module.

ArchMsg.Warning.GICv3_CPUInterface.GICC.MemoryMapped_ReadWriteOnlyReg

DISPLAY GICv3 write-only register %{REG_NAME} attempting to be read. Fields:

OFFSET unsigned int

Offset of address within the GICv3 module.

REG_NAME string

Name of the register being read.

ArchMsg.Warning.GICv3_CPUInterface.GICC.MemoryMapped_WriteIgnored

DISPLAY GICv3 location %{OFFSET} attempting to be written with value %{VALUE} but ignored. Fields:

OFFSET unsigned int

Offset of address within the GICv3 module.

ArchMsg.Warning.GICv3_CPUInterface.GICC.MemoryMapped_WriteReadOnlyReg

DISPLAY GICv3 read-only register %{REG_NAME} attempting to be written with value %{VALUE}. Fields:

OFFSET unsigned int

Offset of address within the GICv3 module.

REG_NAME string

Name of the register being written.

VALUE unsigned int

Value written.

ArchMsg.Warning.GICv3_CPUInterface.GICC.MemoryMapped_WriteReserved

DISPLAY GICv3 reserved location %{OFFSET} attempting to be written with value %{VALUE}. Fields:

OFFSET unsigned int

Offset of address within the GICv3 module.

VALUE unsigned int

Value written.

ArchMsg.Warning.GICv3_CPUInterface.GICH.MemoryMapped_IgnoredRegisterAccessDueToViewMismatch

DISPLAY Attempt to access register %{REGISTER_NAME} from View %{VIEW_ID} is ignored, because this register is accessible only from View %{OWNER_VIEW_ID}. Fields:

OWNER_VIEW_ID unsigned int

The view owning the register.

REGISTER_NAME string

Name of the register being accessed.

VIEW_ID unsigned int

The view trying to access the register.

ArchMsg.Warning.GICv3_CPUInterface.GICH.MemoryMapped_ReadIgnored

DISPLAY GICv3 attempting to read location %{OFFSET} but ignored and RAZ. Fields:

OFFSET unsigned int

Offset of address within the GICv3 module.

ArchMsg.Warning.GICv3_CPUInterface.GICH.MemoryMapped_ReadReserved

DISPLAY GICv3 reserved location %{OFFSET} attempting to be read. Fields:

OFFSET unsigned int

Offset of address within the GICv3 module.

ArchMsg.Warning.GICv3_CPUInterface.GICH.MemoryMapped_ReadWriteOnlyReg

DISPLAY GICv3 write-only register %{REG_NAME} attempting to be read. Fields:

OFFSET unsigned int

Offset of address within the GICv3 module.

REG_NAME string

Name of the register being read.

ArchMsg.Warning.GICv3_CPUInterface.GICH.MemoryMapped_WriteIgnored

DISPLAY GICv3 location %{OFFSET} attempting to be written with value %{VALUE} but ignored. Fields:

OFFSET unsigned int

Offset of address within the GICv3 module.

ArchMsg.Warning.GICv3_CPUInterface.GICH.MemoryMapped_WriteReadOnlyReg

DISPLAY GICv3 read-only register %{REG_NAME} attempting to be written with value %{VALUE}. Fields:

OFFSET unsigned int

Offset of address within the GICv3 module.

REG_NAME string

Name of the register being written.

VALUE unsigned int

Value written.

ArchMsg.Warning.GICv3_CPUInterface.GICH.MemoryMapped_WriteReserved

DISPLAY GICv3 reserved location %{OFFSET} attempting to be written with value %{VALUE}. Fields:

OFFSET unsigned int

Offset of address within the GICv3 module.

VALUE unsigned int

Value written.

ArchMsg.Warning.GICv3_CPUInterface.GICV.MemoryMapped_IgnoredRegisterAccessDueToViewMismatch

DISPLAY Attempt to access register %{REGISTER_NAME} from View %{VIEW_ID} is ignored, because this register is accessible only from View %{OWNER_VIEW_ID}. Fields:

OWNER_VIEW_ID unsigned int

The view owning the register.

REGISTER_NAME string

Name of the register being accessed.

VIEW_ID unsigned int

The view trying to access the register.

ArchMsg.Warning.GICv3_CPUInterface.GICV.MemoryMapped_ReadIgnored

DISPLAY GICv3 attempting to read location %{OFFSET} but ignored and **RAZ**. Fields:

OFFSET unsigned int

Offset of address within the GICv3 module.

ArchMsg.Warning.GICv3_CPUInterface.GICV.MemoryMapped_ReadReserved

DISPLAY GICv3 reserved location %{OFFSET} attempting to be read. Fields:

OFFSET unsigned int

Offset of address within the GICv3 module.

ArchMsg.Warning.GICv3_CPUInterface.GICV.MemoryMapped_ReadWriteOnlyReg

DISPLAY GICv3 write-only register %{REG_NAME} attempting to be read. Fields:

OFFSET unsigned int

Offset of address within the GICv3 module.

REG_NAME string

Name of the register being read.

ArchMsg.Warning.GICv3_CPUInterface.GICV.MemoryMapped_WriteIgnored

DISPLAY GICv3 location %{OFFSET} attempting to be written with value %{VALUE} but ignored. Fields:

OFFSET unsigned int

Offset of address within the GICv3 module.

ArchMsg.Warning.GICv3_CPUInterface.GICV.MemoryMapped_WriteReadOnlyReg

DISPLAY GICv3 read-only register %{REG_NAME} attempting to be written with value %{VALUE}. Fields:

OFFSET unsigned int

Offset of address within the GICv3 module.

REG_NAME string

Name of the register being written.

VALUE unsigned int

Value written.

ArchMsg.Warning.GICv3_CPUInterface.GICV.MemoryMapped_WriteReserved

DISPLAY GICv3 reserved location %{OFFSET} attempting to be written with value %{VALUE}. Fields:

OFFSET unsigned int

Offset of address within the GICv3 module.

VALUE unsigned int

Value written.

ArchMsg.Warning.GICv3_DeactivateInterruptNotSent

Write to EOI/DIR register could not trigger a deactivate. DISPLAY The write to an EOI/DIR register did not trigger a deactivate because no group is modifiable. Fields:

ACCESS_NS bool

Access was non-secure.

CPUID unsigned int

CPU ID specified in end request.

GROUP_1 bool

Interrupt is a group-1 interrupt.

INTERRUPT_ID unsigned int

Interrupt ID that the write is ending.

INTERRUPT_NS bool

Interrupt is a non-secure interrupt.

ArchMsg.Warning.GICv3_EndInterruptIDMismatch

Interrupt ID mismatch. DISPLAY Ending an interrupt (%{END_INTERRUPT_ID}) through %{NS:(secure|non-secure)} write to EOI register that doesn't match the interrupt ID given (%{SPECIFIED_INTERRUPT_ID}). Fields:

END_INTERRUPT_ID unsigned int

Interrupt ID that the write is ending.

NS bool

Access was non-secure.

SPECIFIED_INTERRUPT_ID unsigned int

Interrupt ID that was specified by the register write.

ArchMsg.Warning.GICv3_EndInterruptNotHighestPriority

An interrupt which is not the highest-priority one has been ended. DISPLAY Ending an interrupt (%{INTERRUPT_ID}; CPUID = %{CPUID}) through %{NS:(secure|non-secure)} write to GICC_EOIR or GICC_AEOIR that isn't the highest priority interrupt waiting to be ended (%{LATEST_INTERRUPT_ID}; CPUID = %{CPUID}). Fields:

CPUID unsigned int

CPU ID specified in end request.

INTERRUPT_ID unsigned int

Interrupt ID that the write is ending.

LATEST_CPUID unsigned int

CPU ID for the interrupt that was last acknowledged.

LATEST_INTERRUPT_ID unsigned int

Interrupt ID of the interrupt that was last acknowledged.

NS bool

Access was non-secure.

ArchMsg.Warning.GICv3_EndVirtualInterruptNotAllowed

Invalid attempt to end a virtual interrupt. DISPLAY Attempted to end a virtual interrupt (%{INTERRUPT_ID}; priority = %{PRIORITY}) that isn't allowed because it is %{ACTIVE:(not active|the wrong group)}. Fields:

ACTIVE bool

Matched list register (and virtual interrupt) is active.

GROUP_1 bool

Matched list register is for a group-1 interrupt.

INTERRUPT_ID unsigned int

Interrupt ID that the write is ending.

PRIORITY unsigned int

Priority of the virtual interrupt.

ArchMsg.Warning.GICv3_EndVirtualInterruptNotHighestPriority

Attempt to end a virtual interrupt that is not the highest priority one. DISPLAY Attempted to end a virtual interrupt (%{INTERRUPT_ID}; CPUID = %{CPUID}; priority = %{INTERRUPT_PRIORITY}) that isn't the highest priority active interrupt (%{HIGHEST_PRIORITY}). Fields:

CPUID unsigned int

CPU ID specified in end request.

GROUP_1 bool

Interrupt is a group-1 interrupt.

HIGHEST_PRIORITY unsigned int

Priority of highest priority active interrupt.

INTERRUPT_ID unsigned int

Interrupt ID that the write is ending.

INTERRUPT_PRIORITY unsigned int

Priority level of this interrupt.

PRIORITY unsigned int

Priority of the virtual interrupt.

ArchMsg.Warning.GICv3_ICC_InterruptIDExceedsSupportedSize

InterruptID raw value too large. DISPLAY InterruptID raw value %{VALUE} interpreted as %{MASKEDVALUE} on write to %{REG}. Fields:

MASKEDVALUE unsigned int

The interrupt ID which was understood.

REG string

The name of the register written with the value.

VALUE unsigned int

The written value.

ArchMsg.Warning.GICv3_ICH_LR_UnpredictableHWInterruptID

ICH_LR programmed to be associated with an **UNPREDICTABLE** hw interrupt. DISPLAY %{{IS_GICH_REG:({G}})ICH_LR%{LR_NUM:d} has been programmed to be associated with a private hardware interrupt (ID %{{INTERRUPT_ID}}), which is **UNPREDICTABLE**. (Value set %{{LR_VALUE:x}}). Fields:

INTERRUPT_ID unsigned int

The hardware interrupt ID set in the LR register.

IS_GICH_REG bool

The register being programmed is GICH_LRn (rather than ICH_LRn).

LR_NUM unsigned int

The GICH_LRn register written.

LR_VALUE unsigned int

The value now written to the LR register.

ArchMsg.Warning.GICv3_InvalidRegAccessForLPI

Invalid register access for LPI. DISPLAY %{{REG}} accessed for LPI InterruptID %{{VALUE}}. Fields:

REG string

The name of the register written with the value.

VALUE unsigned int

LPI interrupt id.

ArchMsg.Warning.GICv3_ReceiveClearDoesntMatchPending

Distributor attempted to clear an interrupt which isn't the pending one. DISPLAY Distributor sent a command to the CPU interface to clear an interrupt (ID: %{{CLEAR_INTERRUPT_ID}} when the pending interrupt doesn't match (ID: %{{PENDING_INTERRUPT_ID}}). Fields:

CLEAR_INTERRUPT_ID unsigned int

Interrupt ID that the distributor is attempting to clear.

PENDING_INTERRUPT_ID unsigned int

Interrupt ID that is pending.

ArchMsg.Warning.GICv3_ReceiveClearWhenNoInterruptPending

Attempt to clear an interrupt when no interrupt is pending. DISPLAY Distributor sent a command to the CPU interface to clear an interrupt (ID: %{{INTERRUPT_ID}} when there isn't a pending interrupt. Fields:

INTERRUPT_ID unsigned int

Interrupt ID that the distributor is attempting to clear.

ArchMsg.Warning.GICv3_ReceivedInvalidCommandFromAbove

DISPLAY Received a GICv3 command type %{TYPE} of length %{ACTUAL_LENGTH} bytes from upstream port; %{EXPECTED_LENGTH} bytes was expected. Fields:

ACTUAL_LENGTH unsigned int

Actual length of the data in bytes including the header.

DATA unsigned int

The data received (including header).

EXPECTED_LENGTH unsigned int

The length in bytes including header that was expected to be received dependent on command type and potentially the data type in the case of a data write.

TYPE unsigned int

The command type.

ArchMsg.Warning.GICv3_ReceivedSetWithGroupAndGroupmod

Distributor sent a Set packet with both Group and GrpMod set. DISPLAY Distributor sent a Set packet with both Group and GrpMod set, forcing GrpMod to 0.

ArchMsg.Warning.GICv3_SGI_RSS_unsupported_unpredictable

RSS field non zero in SGI generation when the IRI does not advertise support. DISPLAY RSS field non zero in SGI generation when the IRI does not advertise support!.

ArchMsg.Warning.GICv3_UnexpectedAck

Unexpected command received from distributor. DISPLAY %{COMMAND} received from distributor when one wasn't expected. Is reset order incorrect?. Fields:

COMMAND enum

The command type received.

ArchMsg.Warning.GICv3_WriteIgnoredConfiguration

A register write has been ignored. DISPLAY %{REG_NAME} write of value %{VALUE} ignored due to GICv3 configuration. Fields:

REG_NAME string

Name of register being read or written.

VALUE unsigned int

Value attempting to be written.

GICv3_AcknowledgeInterrupt

CPU Interface is starting the process of acknowledging the interrupt %{INTERRUPT_ID}. Fields:

INTERRUPT_ID unsigned int

Interrupt ID being acknowledged.

GICv3_CFGSDISABLE

DISPLAY CFGSDISABLE signal has been %{STATE:(clear|set)}ed. Fields:

STATE bool

The new state of the signal.

GICv3_CPUIF_Updating_HPPI_Selection

DISPLAY CPUIF is updating the HPPI selection from interrupt %{OLD_CANDIDATE} to interrupt %{NEW_CANDIDATE}. Fields:

NEW_CANDIDATE unsigned int

New HPPI ID.

OLD_CANDIDATE unsigned int

New HPPI ID.

GICv3_CPUIF_clear_pending_interrupt

DISPLAY CPUIF cleared pending interrupt \${INTERRUPT_ID}. Fields:

INTERRUPT_ID unsigned int

Interrupt ID being cleared.

GICv3_CPUIF_start_releasing_interrupt

DISPLAY CPUIF start releasing interrupt \${INTERRUPT_ID}. Fields:

INTERRUPT_ID unsigned int

Interrupt ID being released.

GICv3_CPUInterface.GICC.MemoryMapped_MultipleViews_Access

DISPLAY GICv3 %{OPERATION} memory access on address %{ADDRESS} from view %{VIEW}. Fields:

ADDRESS unsigned int

The address of the register being accessed.

OPERATION string

read/write operation.

VIEW unsigned int

The view being accessed.

GICv3_CPUInterface.GICC.MemoryMapped_Read

Trace read from a GICv3 memory-mapped register. Fields:

A0 unsigned int

Affinity level 0 (for banked registers).

A1 unsigned int

Affinity level 1 (for banked registers).

A2 unsigned int

Affinity level 2 (for banked registers).

A3 unsigned int

Affinity level 3 (for banked registers).

BANKED bool

Whether the access is to a banked register.

NS bool

Access is non-secure.

OFFSET unsigned int

Offset of address within the GICv3 module.

REG_NAME string

Name of the register being read.

VALUE unsigned int

Value returned.

GICv3_CPUInterface.GICC.MemoryMapped_Read64

Trace read from a GICv3 memory-mapped register. Fields:

A0 unsigned int

Affinity level 0 (for banked registers).

A1 unsigned int

Affinity level 1 (for banked registers).

A2 unsigned int

Affinity level 2 (for banked registers).

A3 unsigned int

Affinity level 3 (for banked registers).

BANKED bool

Whether the access is to a banked register.

NS bool

Access is non-secure.

OFFSET unsigned int

Offset of address within the GICv3 module.

REG_NAME string

Name of the register being read.

VALUE unsigned int

Value returned.

GICv3_CPUInterface.GICC.MemoryMapped_Write

Trace write to a GICv3 memory-mapped register. Fields:

A0 unsigned int

Affinity level 0 (for banked registers).

A1 unsigned int

Affinity level 1 (for banked registers).

A2 unsigned int

Affinity level 2 (for banked registers).

A3 unsigned int

Affinity level 3 (for banked registers).

BANKED bool

Whether the access is to a banked register.

NS bool

Access is non-secure.

OFFSET unsigned int

Offset of address within the GICv3 module.

REG_NAME string

Name of the register being written.

UPDATED_VALUE unsigned int

New value read back from the register; for write-only registers this will be equal to VALUE.

VALUE unsigned int

Value written.

GICv3_CPUInterface.GICC.MemoryMapped_Write64

Trace write to a GICv3 memory-mapped register. Fields:

A0 unsigned int

Affinity level 0 (for banked registers).

A1 unsigned int

Affinity level 1 (for banked registers).

A2 unsigned int

Affinity level 2 (for banked registers).

A3 unsigned int

Affinity level 3 (for banked registers).

BANKED bool

Whether the access is to a banked register.

NS bool

Access is non-secure.

OFFSET unsigned int

Offset of address within the GICv3 module.

REG_NAME string

Name of the register being written.

UPDATED_VALUE unsigned int

New value read back from the register; for write-only registers this will be equal to VALUE.

VALUE unsigned int

Value written.

GICv3_CPUInterface.GICC.MemoryMapped_WritePort

Trace write to a GICv3 memory-mapped register acting as imp def port. Fields:

FIRST_BYTE_IN_BURST unsigned int

First byte written in the burst.

NS bool

Access is non-secure.

OFFSET unsigned int

Offset of address within the GICv3 module.

REG_NAME string

Name of the register being written.

GICv3_CPUInterface.GICH.MemoryMapped_MultipleViews_Access

DISPLAY GICv3 %{OPERATION} memory access on address %{ADDRESS} from view %{VIEW}. Fields:

ADDRESS unsigned int

The address of the register being accessed.

OPERATION string

read/write operation.

VIEW unsigned int

The view being accessed.

GICv3_CPUInterface.GICH.MemoryMapped_Read

Trace read from a GICv3 memory-mapped register. Fields:

A0 unsigned int

Affinity level 0 (for banked registers).

A1 unsigned int

Affinity level 1 (for banked registers).

A2 unsigned int

Affinity level 2 (for banked registers).

A3 unsigned int

Affinity level 3 (for banked registers).

BANKED bool

Whether the access is to a banked register.

NS bool

Access is non-secure.

OFFSET unsigned int

Offset of address within the GICv3 module.

REG_NAME string

Name of the register being read.

VALUE unsigned int

Value returned.

GICv3_CPUInterface.GICH.MemoryMapped_Read64

Trace read from a GICv3 memory-mapped register. Fields:

A0 unsigned int

Affinity level 0 (for banked registers).

A1 unsigned int

Affinity level 1 (for banked registers).

A2 unsigned int

Affinity level 2 (for banked registers).

A3 unsigned int

Affinity level 3 (for banked registers).

BANKED bool

Whether the access is to a banked register.

NS bool

Access is non-secure.

OFFSET unsigned int

Offset of address within the GICv3 module.

REG_NAME string

Name of the register being read.

VALUE unsigned int

Value returned.

GICv3_CPUInterface.GICH.MemoryMapped_Write

Trace write to a GICv3 memory-mapped register. Fields:

A0 unsigned int

Affinity level 0 (for banked registers).

A1 unsigned int

Affinity level 1 (for banked registers).

A2 unsigned int

Affinity level 2 (for banked registers).

A3 unsigned int

Affinity level 3 (for banked registers).

BANKED bool

Whether the access is to a banked register.

NS bool

Access is non-secure.

OFFSET unsigned int

Offset of address within the GICv3 module.

REG_NAME string

Name of the register being written.

UPDATED_VALUE unsigned int

New value read back from the register; for write-only registers this will be equal to VALUE.

VALUE unsigned int

Value written.

GICv3_CPUInterface.GICH.MemoryMapped_Write64

Trace write to a GICv3 memory-mapped register. Fields:

A0 unsigned int

Affinity level 0 (for banked registers).

A1 unsigned int

Affinity level 1 (for banked registers).

A2 unsigned int

Affinity level 2 (for banked registers).

A3 unsigned int

Affinity level 3 (for banked registers).

BANKED bool

Whether the access is to a banked register.

NS bool

Access is non-secure.

OFFSET unsigned int

Offset of address within the GICv3 module.

REG_NAME string

Name of the register being written.

UPDATED_VALUE unsigned int

New value read back from the register; for write-only registers this will be equal to VALUE.

VALUE unsigned int

Value written.

GICv3_CPUInterface.GICH.MemoryMapped_WritePort

Trace write to a GICv3 memory-mapped register acting as imp def port. Fields:

FIRST_BYTE_IN_BURST unsigned int

First byte written in the burst.

NS bool

Access is non-secure.

OFFSET unsigned int

Offset of address within the GICv3 module.

REG_NAME string

Name of the register being written.

GICv3_CPUInterface.GICV.MemoryMapped_MultipleViews_Access

DISPLAY GICv3 %{OPERATION} memory access on address %{ADDRESS} from view %{VIEW}. Fields:

ADDRESS unsigned int

The address of the register being accessed.

OPERATION string

read/write operation.

VIEW unsigned int

The view being accessed.

GICv3_CPUInterface.GICV.MemoryMapped_Read

Trace read from a GICv3 memory-mapped register. Fields:

A0 unsigned int

Affinity level 0 (for banked registers).

A1 unsigned int

Affinity level 1 (for banked registers).

A2 unsigned int

Affinity level 2 (for banked registers).

A3 unsigned int

Affinity level 3 (for banked registers).

BANKED bool

Whether the access is to a banked register.

NS bool

Access is non-secure.

OFFSET unsigned int

Offset of address within the GICv3 module.

REG_NAME string

Name of the register being read.

VALUE unsigned int

Value returned.

GICv3_CPUInterface.GICV.MemoryMapped_Read64

Trace read from a GICv3 memory-mapped register. Fields:

A0 unsigned int

Affinity level 0 (for banked registers).

A1 unsigned int

Affinity level 1 (for banked registers).

A2 unsigned int

Affinity level 2 (for banked registers).

A3 unsigned int

Affinity level 3 (for banked registers).

BANKED bool

Whether the access is to a banked register.

NS bool

Access is non-secure.

OFFSET unsigned int

Offset of address within the GICv3 module.

REG_NAME string

Name of the register being read.

VALUE unsigned int

Value returned.

GICv3_CPUInterface.GICV.MemoryMapped_Write

Trace write to a GICv3 memory-mapped register. Fields:

A0 unsigned int

Affinity level 0 (for banked registers).

A1 unsigned int

Affinity level 1 (for banked registers).

A2 unsigned int

Affinity level 2 (for banked registers).

A3 unsigned int

Affinity level 3 (for banked registers).

BANKED bool

Whether the access is to a banked register.

NS bool

Access is non-secure.

OFFSET unsigned int

Offset of address within the GICv3 module.

REG_NAME string

Name of the register being written.

UPDATED_VALUE unsigned int

New value read back from the register; for write-only registers this will be equal to VALUE.

VALUE unsigned int

Value written.

GICv3_CPUInterface.GICV.MemoryMapped_Write64

Trace write to a GICv3 memory-mapped register. Fields:

A0 unsigned int

Affinity level 0 (for banked registers).

A1 unsigned int

Affinity level 1 (for banked registers).

A2 unsigned int

Affinity level 2 (for banked registers).

A3 unsigned int

Affinity level 3 (for banked registers).

BANKED bool

Whether the access is to a banked register.

NS bool

Access is non-secure.

OFFSET unsigned int

Offset of address within the GICv3 module.

REG_NAME string

Name of the register being written.

UPDATED_VALUE unsigned int

New value read back from the register; for write-only registers this will be equal to VALUE.

VALUE unsigned int

Value written.

GICv3_CPUInterface.GICV.MemoryMapped_WritePort

Trace write to a GICv3 memory-mapped register acting as imp def port. Fields:

FIRST_BYTE_IN_BURST unsigned int

First byte written in the burst.

NS bool

Access is non-secure.

OFFSET unsigned int

Offset of address within the GICv3 module.

REG_NAME string

Name of the register being written.

GICv3_CPUInterface_Comms_Receive

GICv3 internal communications packet that has been received and is traveling towards the CPU. Fields:

COMMAND enum

The command type for this packet.

DATA unsigned int

The data (including header) contained within this packet.

DATA_LENGTH unsigned int

Number of bytes in the data (including header).

GICv3_CPUInterface_Comms_Send

GICv3 internal communications packet that is being sent towards the top-level. Fields:

COMMAND enum

The command type for this packet.

DATA unsigned int

The data (including header) contained within this packet.

DATA_LENGTH unsigned int

Number of bytes in the data (including header).

GICv3_CreateInterruptAPRWrite

An active interrupt has been created by a write to APR. **DISPLAY** An active group $\% \{ \text{GROUP_1:} (0|1) \}$ interrupt with priority $\% \{ \text{PRIORITY} \}$ has been created due to the corresponding priority bit in $\text{GICC_}\% \{ \text{NS_REG:} (| \text{NS}) \}$ APR being set. Fields:

GROUP_1 bool

Whether the interrupt is created as group 1.

NS_REG bool

Whether the access was through GICC_NSAPR.

PRIORITY unsigned int

Priority of the interrupt being created.

GICv3_DeactivateInterrupt

Trace writes to EOI/DIR registers that deactivate an interrupt. Fields:

ACCESS_NS bool

Access was non-secure.

CPUID unsigned int

CPU ID specified in end request.

GROUP_1 bool

Interrupt is a group-1 interrupt.

INTERRUPT_ID unsigned int

Interrupt ID that the write is ending.

INTERRUPT_NS bool

Interrupt is a non-secure interrupt.

GICv3_EndInterrupt

Trace writes to GICC_EOIR or GICC_AEOIR that end an interrupt. Fields:

CPUID unsigned int

CPU ID specified in end request.

DEACTIVATE bool

Interrupt is deactivated (rather than requiring a separate DIR request); will never be true for LPIs.

GROUP_1 bool

Interrupt is a group-1 interrupt.

INTERRUPT_ID unsigned int

Interrupt ID that the write is ending.

NS bool

Access was non-secure.

GICv3_EndInterruptAPRWrite

An interrupt has been deactivated because of a write to the priority register. DISPLAY Interrupt ID %{INTERRUPT_ID} with priority %{PRIORITY} has been ended due to corresponding priority bit in GICC_%{NS_REG:([NS])}APR being cleared. Fields:

INTERRUPT_ID unsigned int

Interrupt ID that the write is ending.

NS_REG bool

Whether the access was through GICC_NSAPR.

PRIORITY unsigned int

Priority of the interrupt being deactivated.

GICv3_Generating_SGI_REQUESTED

DISPLAY CPUIF request generating SGI through %{IS_DELAYED:(delayed|non-delayed)} register %{REG_NAME}. Fields:

IS_DELAYED bool

delayed or not.

REG_NAME string

Register name.

GICv3_ICH_MISR_Changed

Change of ICH_MISR register. DISPLAY ICH_MISR register value changed to %{VALUE}.
Fields:

VALUE unsigned int

The new value for ICH_MISR.

GICv3_IRQBypassSignalIn

An interrupt bypass signal has changed state. DISPLAY Input %{IS_FIQ:(IRQ|FIQ)} bypass signal has been %{VALUE:(cleared|set)}. Fields:

IS_FIQ bool

Whether the signal in question is the FIQ signal (rather than the IRQ).

VALUE bool

New value for the signal (i.e. whether it is set).

GICv3_IRQLegacySignalOut

An output legacy interrupt signal has changed state. DISPLAY Output legacy %{IS_FIQ:(IRQ|FIQ)} signal has been %{VALUE:(cleared|set)}. Fields:

IS_FIQ bool

Whether the signal in question is the FIQ signal (rather than the IRQ).

VALUE bool

New value for the signal (i.e. whether it is set).

GICv3_IRQNMIBypassSignalIn

An interrupt bypass signal has changed state. DISPLAY Input %{IS_FIQ:(IRQ|FIQ)} NMI bypass signal has been %{VALUE:(cleared|set)}. Fields:

IS_FIQ bool

Whether the signal in question is the FIQ signal (rather than the IRQ).

VALUE bool

New value for the signal (i.e. whether it is set).

GICv3_IRQNMISignalOut

An interrupt signal with NMI has changed state. DISPLAY %{VIRTUAL:(|V)}%{IS_FIQ:(IRQ|FIQ)} NMI signal has been %{VALUE:(cleared|set)}. Fields:

IS_FIQ bool

Whether the signal in question is the FIQ signal (rather than the IRQ).

VALUE bool

New value for the signal (i.e. whether it is set).

VIRTUAL bool

Whether we're changing the virtual signal.

GICv3_IRQSignalOut

An interrupt signal has changed state. DISPLAY %{VIRTUAL:([V])}%{IS_FIQ:([IRQ|FIQ])} signal has been %{VALUE:(cleared|set)}. Fields:

IS_FIQ bool

Whether the signal in question is the FIQ signal (rather than the IRQ).

VALUE bool

New value for the signal (i.e. whether it is set).

VIRTUAL bool

Whether we're changing the virtual signal.

GICv3_InterfaceQuiesced

Event fires when the CPU interface is quiesced (Quiesce Acknowledge command sent to distributor).

GICv3_MaintenanceInterruptChanged

Change of state of maintenance interrupt. DISPLAY Maintenance interrupt has been %{STATE:(cleared|set)}. Fields:

STATE bool

Whether the maintenance interrupt was set or cleared.

GICv3_PIDR

DISPLAY SW accessing %{POS} %{REG_PREFIX}_PIDR<%{INDEX}> register for view %{VIEW_ID}. Fields:

INDEX unsigned int

The index of the register.

POS string

high/low.

REG_PREFIX string

The register prefix.

VIEW_ID unsigned int

The view-id.

GICv3_PhysicalInterruptNotSignalled

DISPLAY The physical interrupt %{INTERRUPT_ID} won't be signalled because %{REASON}. Fields:

INTERRUPT_ID unsigned int

The interrupt ID.

REASON string

The reason of not signalling the interrupt.

GICv3_RegUpdated32

DISPLAY %{REG_NAME} has been updated to %{VALUE} by a system event. Fields:

REG_NAME string

The name of the register being updated.

VALUE unsigned int

The updated value.

GICv3_RegUpdated64

DISPLAY %{REG_NAME} has been updated to %{VALUE} by a system event. Fields:

REG_NAME string

The name of the register being updated.

VALUE unsigned int

The updated value.

GICv3_Updating_Interrupt_Signals

DISPLAY CPUIF updating interrupt signals as a result of pending interrupt %{INTERRUPT_ID}. Fields:

INTERRUPT_ID unsigned int

The interrupt ID.

GICv3_VirtualInterruptNotSignalled

DISPLAY The virtual interrupt %{INTERRUPT_ID} won't be signalled because %{REASON}. Fields:

INTERRUPT_ID unsigned int

The hardware interrupt ID set in the LR register.

REASON string

The reason of not signalling the interrupt.

2.107 GICv3CPUInterfaceDecoder

This section describes the trace sources.

GICv3_DECODER_DECODE

Trace accesses through the GICv3 decoder (not including pass-through). Fields:

ADDR unsigned int

Address of the access.

ADDR_OUTPUT_OFFSET unsigned int

The output offset for this transaction.

FORCE_SECURE bool

Whether this transaction is using the input port that forces transaction attributes to secure.

OUTPUT_PORT_NAME *string*

The name of the output port that we're using.

READnWRITE *bool*

Whether this transaction is a read (rather than a write).

GICV3_DECODER_PASSTHRU

Trace accesses through the GICv3 decoder that are being passed straight-through. Fields:

ADDR *unsigned int*

Address of the access.

READnWRITE *bool*

Whether this transaction is a read (rather than a write).

2.108 GICv3CommsLogger

This section describes the trace sources.

ACTIVATE

The CPU interface sends an Activate command when acknowledging an interrupt. Fields:

DIRECTION *enum*

Direction of the command.

INTID *unsigned int*

Interrupt number.

V *enum*

The Activate corresponds to a Set or VSet command.

ACTIVATE_ACKNOWLEDGE

The Redistributor sends an Activate Acknowledge response to confirm receipt of an Activate command. Fields:

DIRECTION *enum*

Direction of the command.

V *enum*

The Activate Acknowledge corresponds to a Set or VSet command.

CLEAR

The Clear command clears the specified pending interrupt. Fields:

DIRECTION *enum*

Direction of the command.

INTID *unsigned int*

Interrupt number.

CLEAR_ACKNOWLEDGE

Clear Acknowledge response to acknowledge the receipt of a Clear or VClear command.

Fields:

DIRECTION enum

Direction of the command.

V enum

The Clear Acknowledge corresponds to a Clear or VClear command.

DEACTIVATE

The Deactivate command deactivates an interrupt, provided the initiating Exception level and Security state can access the interrupt group to which the INTID belongs. Fields:

DIRECTION enum

Direction of the command.

GROUP0 bool

1=Group 0 interrupts can be modified.

GROUP1_NS bool

1=Non-secure group 1 interrupts can be modified.

GROUP1_S bool

1=Secure group 1 interrupts can be modified.

INTID unsigned int

Interrupt number.

DEACTIVATE_ACKNOWLEDGE

The Redistributor sends a Deactivate Acknowledge response to confirm receipt of a Deactivate command. Fields:

DIRECTION enum

Direction of the command.

DOWNSTREAM_CONTROL_ACKNOWLEDGE

Downstream Control Acknowledge response to confirm receipt of a Downstream Control command. Fields:

DIRECTION enum

Direction of the command.

PL enum

Size of pINTID.

VL enum

Size of vINTID.

DOWNSTREAM_CONTROL_IMP_DEF

Implementation defined command send to CPU interface. Fields:

DIRECTION enum

Direction of the command.

DOWNSTREAM_CONTROL_SETTINGS

Communicates settings to CPU interface. Fields:

DIRECTION enum

Direction of the command.

DS bool

Disable Security. Indicates the value of GICD_CTLR.DS.

PL enum

Size of pINTID.

VL enum

Size of vINTID.

GENERATE_SGI

The CPU interface sends a Generate SGI command to the Redistributor to generate an SGI.
Fields:

A1 unsigned int

Affinity level 1 values used for generating the set of target PEs.

A2 unsigned int

Affinity level 2 values used for generating the set of target PEs.

A3 unsigned int

Affinity level 3 values used for generating the set of target PEs.

A3V bool

Command includes an A3 field.

DIRECTION enum

Direction of the command.

IRM bool

Interrupt Routing Mode used.

NS bool

0=command originates from a Secure Execution state. 1=command originates from a Non-secure Execution state.

SGInum unsigned int

The INTID of the SGI to be generated.

SGT enum

Register access that caused the Generate SGI command.

Target_List unsigned int

The group of target PEs defined by the routing mode.

GENERATE_SGI_ACKNOWLEDGE

The Redistributor sends a Generate SGI Acknowledge response to confirm that it has received a Generate SGI command from the CPU interface. Fields:

DIRECTION enum

Direction of the command.

QUIESCE

The Redistributor sends a Quiesce command to request that the CPU interface enters the quiescent state. Fields:

DIRECTION enum

Direction of the command.

QUIESCE_ACKNOWLEDGE

Quiesce Acknowledge response to confirm receipt of a Quiesce command, and to confirm that it is quiescent. Fields:

DIRECTION enum

Direction of the command.

RELEASE

The CPU interface logic sends a Release response when it cannot handle a particular interrupt. Fields:

DIRECTION enum

Direction of the command.

INTID unsigned int

Interrupt number.

V enum

The Release corresponds to a Set or VSet command.

SET

The Set command sets the highest priority pending interrupt for a PE. Fields:

DIRECTION enum

Direction of the command.

GROUP bool

The interrupt group, as indicated by the corresponding GICD_IGROUPR<n>.Group status bit.

INTID unsigned int

Interrupt number.

MOD bool

The value of the GICD_IGRPMODR<n>.Group status bit for the interrupt.

PRIORITY unsigned int

The actual priority of the interrupt, that is, the Secure, unshifted view.

UPSTREAM_CONTROL_ACKNOWLEDGE

The Redistributor sends an Upstream Control Acknowledge response to confirm receipt of an Upstream Control command. Fields:

DIRECTION enum

Direction of the command.

UPSTREAM_CONTROL_PHYSICAL_ENABLE

Communicates physical CPU interface enable to the Redistributor. Fields:

DIRECTION enum

Direction of the command.

GROUP0 bool

The value of ICC_IGRPEN0_EL1.Enable.

GROUP1_NS bool

The value of the Non-secure copy of ICC_IGRPEN1_EL1.Enable.

GROUP1_S bool

The value of the Secure copy of ICC_IGRPEN1_EL1.Enable.

UPSTREAM_CONTROL_PHYSICAL_PRIORITY

Communicates Priority Mask to the Redistributor. Fields:

DIRECTION enum

Direction of the command.

PMR unsigned int

The value of PMR.

UPSTREAM_CONTROL_VIRTUAL_ENABLE

Communicates virtual CPU interface enable to the Redistributor. Fields:

DIRECTION enum

Direction of the command.

GROUP0 bool

The value of ICH_VMCR_EL2.VENG0.

GROUP1 bool

The value of ICH_VMCR_EL2.VENG1.

VCLEAR

The VClear command resets the highest priority pending virtual interrupt. Fields:

DIRECTION enum

Direction of the command.

VINTID unsigned int

Interrupt number.

VSET

The VSet command sets the highest priority pending interrupt for a PE. Fields:

DIRECTION enum

Direction of the command.

GROUP bool

The interrupt group.

PRIORITY unsigned int

The actual priority of the interrupt, that is, the Secure, unshifted view.

VINTID unsigned int

Interrupt number.

2.109 GICv3CommsPVBUS

This section describes the trace sources.

ArchMsg.Warning.GICv3CommsPVBUS_PacketTooBig

DISPLAY Received incoming PVBUS packet that is too large to be sent over GICv3Comms connection (length = %{LENGTH}, data[0] (packet type) = 0x%{DATA_0:x}, index/manager ID = %{INDEX}). Fields:

DATA_0 unsigned int

First byte of the packet (normally data type).

INDEX unsigned int

Index of GICv3Comms connection that the message should have been sent to (based on AXI manager ID).

LENGTH unsigned int

Number bytes in data packet.

ArchMsg.Warning.GICv3CommsPVBUS_UnknownDestination

DISPLAY Received incoming PVBUS packet that indicates that it should be sent to an unconnected destination (index/manager ID = %{INDEX}, length = %{LENGTH}, data[0] (packet type) = 0x%{DATA_0:x}). Fields:

DATA unsigned int

The message data.

DATA_0 unsigned int

First byte of the packet (normally data type).

INDEX unsigned int

Index of GICv3Comms connection that the message should have been sent to (based on AXI manager ID).

LENGTH unsigned int

Number bytes in data packet.

GICv3CommsPVBUS_FailedMessage

Processed message from connected GICv3Comms port and sent out, but PVBUS transaction failed. Fields:

DATA unsigned int

The message data.

INDEX unsigned int

Index of GICv3Comms connection that message came in on.

LENGTH unsigned int

Message length in bytes.

GICv3CommsPVBUS_ProcessedMessage

Processed message from connected GICv3Comms port. Fields:

DATA unsigned int

The message data.

INDEX unsigned int

Index of GICv3Comms connection that message came in on.

LENGTH unsigned int

Message length in bytes.

GICv3CommsPVBUS_ReceiveAXIManagerID

DISPLAY Received AXI manager ID %{MANAGER_ID:x} for connection %{INDEX}. Fields:

INDEX unsigned int

Index of GICv3Comms connection that message came in on.

MANAGER_ID unsigned int

Manager ID value recieved.

GICv3CommsPVBUS_SentViaGICv3Comms

Packet received via PVBUS and sent via GICv3 Comms to connected distributor/CPU. Fields:

DATA unsigned int

The message data.

INDEX unsigned int

Index of GICv3Comms connection that the message should have been sent to (based on AXI manager ID).

LENGTH unsigned int

Number bytes in data packet.

2.110 GICv3Distributor

This section describes the trace sources.

ArchMsg.Debug.GICv3_MemoryMapped_TriggerDisabledSPI

DISPLAY Triggering SPI (%{INTERRUPT_ID}) which is disabled because of (%{REASON}).

Fields:

INTERRUPT_ID unsigned int

The interrupt ID specified in the register access.

REASON string

The reason of having disabled interrupt.

ArchMsg.Info.GICv3_DroppedInternalPacket

DISPLAY Internal packet dropped as %{INTERFACE} is asleep. Fields:

INTERFACE unsigned int

Index of the interface.

ArchMsg.Info.GICv3_DroppedInternalPacketForALLCores

DISPLAY Internal packet dropped for all cores.

ArchMsg.Info.GICv3_DroppedInternalPacketSDisabled

DISPLAY Internal packet dropped as %{INTERFACE} is disabled. Fields:

INTERFACE unsigned int

Index of the interface.

ArchMsg.Info.GICv3_MemoryMapped_SkippedPossibleDestinationForSPIDueToViewMismatch

DISPLAY Skipped the redistributor with processor number %{PROCESSOR_NUMBER} for being a possible destination for SPI %{SPI_ID} because this SPI is owned by view %{SPI_OWNER_VIEW_ID}, while this Redistributor is assigned to View %{REDISTRIBUTOR_VIEW_ID}. Fields:

PROCESSOR_NUMBER unsigned int

Processor number of the skipped redistributor.

REDISTRIBUTOR_VIEW_ID unsigned int

The view owning the redistributor.

SPI_ID unsigned int

The ID of the spi.

SPI_OWNER_VIEW_ID unsigned int

The view owning the SPI.

ArchMsg.Warning.GICv3_CFGSDISABLE_unsupported

DISPLAY CFGSDISABLE signal has been set but doing so has no effect.

ArchMsg.Warning.GICv3_Distributor.MemoryMapped_IgnoredRegisterAccessDueToViewMismatch

DISPLAY Attempt to access register %{REGISTER_NAME} from View %{VIEW_ID} is ignored, because this register is accessible only from View %{OWNER_VIEW_ID}. Fields:

OWNER_VIEW_ID unsigned int

The view owning the register.

REGISTER_NAME string

Name of the register being accessed.

VIEW_ID unsigned int

The view trying to access the register.

ArchMsg.Warning.GICv3_Distributor.MemoryMapped_ReadIgnored

DISPLAY GICv3 attempting to read location %{OFFSET} but ignored and **RAZ**. Fields:

OFFSET unsigned int

Offset of address within the GICv3 module.

ArchMsg.Warning.GICv3_Distributor.MemoryMapped_ReadReserved

DISPLAY GICv3 reserved location %{OFFSET} attempting to be read. Fields:

OFFSET unsigned int

Offset of address within the GICv3 module.

ArchMsg.Warning.GICv3_Distributor.MemoryMapped_ReadWriteOnlyReg

DISPLAY GICv3 write-only register %{REG_NAME} attempting to be read. Fields:

OFFSET unsigned int

Offset of address within the GICv3 module.

REG_NAME string

Name of the register being read.

ArchMsg.Warning.GICv3_Distributor.MemoryMapped_WriteIgnored

DISPLAY GICv3 location %{OFFSET} attempting to be written with value %{VALUE} but ignored. Fields:

OFFSET unsigned int

Offset of address within the GICv3 module.

ArchMsg.Warning.GICv3_Distributor.MemoryMapped_WriteReadOnlyReg

DISPLAY GICv3 read-only register %{REG_NAME} attempting to be written with value %{VALUE}. Fields:

OFFSET unsigned int

Offset of address within the GICv3 module.

REG_NAME string

Name of the register being written.

VALUE unsigned int

Value written.

ArchMsg.Warning.GICv3_Distributor.MemoryMapped_WriteReserved

DISPLAY GICv3 reserved location %{OFFSET} attempting to be written with value %{VALUE}.

Fields:

OFFSET unsigned int

Offset of address within the GICv3 module.

VALUE unsigned int

Value written.

ArchMsg.Warning.GICv3_DroppedInternalPacket

DISPLAY Internal packet dropped as %{INTERFACE} is not connected. Fields:

INTERFACE unsigned int

Index of the interface.

ArchMsg.Warning.GICv3_MemoryMapped_AccessInIncorrectAREState

DISPLAY %{REG_NAME} accessed with %{NS:(secure|non-secure)} %{READnWRITE:(write|read)} when GICD_CTLR.ARE_NS = %{ARE_NS} and GICD_CTLR.ARE_S = %{ARE_S}. (May be due to all interrupts in register being in wrong security state for ARE settings.). Fields:

ARE_NS bool

Value of GICD_CTLR.ARE_NS.

ARE_S bool

Value of GICD_CTLR.ARE_S.

NS bool

Access is non-secure.

READnWRITE bool

Access is a read (rather than a write).

REG_NAME string

Name of the register being accessed.

ArchMsg.Warning.GICv3_MemoryMapped_AccessingSPIOutOfRange

DISPLAY Accessing %{REG_NAME} ignored as specified interrupt ID (%{INTERRUPT_ID}) is not a valid SPI. Fields:

INTERRUPT_ID unsigned int

The interrupt ID specified in the register access.

REG_NAME string

Name of the register being accessed.

ArchMsg.Warning.GICv3_MemoryMapped_ChangeUnimplementedSPI

DISPLAY Write attempted to change the configuration for an unimplemented SPI.

ArchMsg.Warning.GICv3_MemoryMapped_IRouterR_RoutingModeWrittenWhenRazWi

DISPLAY %{REG_NAME} was written with RM=1 but RM is **RAZ/WI**. Fields:

REG_NAME string

Name of the register being accessed.

ArchMsg.Warning.GICv3_MemoryMapped_IgnoredAccessToSPIDueToViewMismatch

DISPLAY Attempt to access SPI %{INTERRUPT_ID} through %{ACCESSOR_NAME} from View %{VIEW_ID} is ignored because this SPI is owned by View %{OWNER_VIEW_ID}. Fields:

ACCESSOR_NAME string

Name of the register being accessed.

INTERRUPT_ID unsigned int

The SPI attempting to be set/cleared.

OWNER_VIEW_ID unsigned int

The view owning the interrupt.

VIEW_ID unsigned int

The view trying to access the register.

ArchMsg.Warning.GICv3_MemoryMapped_IgnoredPendingChange

DISPLAY Access to %{REG_NAME} ignored as specified interrupt ID (%{INTERRUPT_ID}) is %{IS_PENDING:(not|already)} pending. Fields:

INTERRUPT_ID unsigned int

The interrupt ID specified in the register access.

IS_PENDING bool

The current pending state of the interrupt (and what the access attempted to set it to).

REG_NAME string

Name of the register being accessed.

ArchMsg.Warning.GICv3_MemoryMapped_NSAccessToSecureInterruptIgnored

DISPLAY Access to %{REG_NAME} ignored as specified interrupt ID (%{INTERRUPT_ID}) is configured as a secure interrupt. Fields:

INTERRUPT_ID unsigned int

The interrupt ID specified in the register access.

REG_NAME string

Name of the register being accessed.

ArchMsg.Warning.GICv3_MemoryMapped_NSWriteToSecureInterrupt

DISPLAY Non-secure %{READnWRITE:(write to|read from)} %{REG_NAME} ignored because it is attempting to update a secure interrupt. Fields:

INTERRUPT_ID unsigned int

The interrupt ID specified in the register access.

READnWRITE bool

Access is a read (rather than a write).

REG_NAME string

Name of the register being accessed.

ArchMsg.Warning.GICv3_MemoryMapped_SGIRWrittenUsingReservedValue

DISPLAY %{REG_NAME} write of %{VALUE} is ignored as the TargetListFilter field has the reserved value (0b11) specified. Fields:

REG_NAME string

Name of the register being written.

VALUE unsigned int

Value being written to the register.

ArchMsg.Warning.GICv3_MemoryMapped_SGISendToZeroTargets

DISPLAY %{REG_NAME} write of %{VALUE} is ignored as the CPUTargetList field specifies zero targets. Fields:

REG_NAME string

Name of the register being written.

VALUE unsigned int

Value being written to the register.

ArchMsg.Warning.GICv3_MemoryMapped_SGIWrongGroup

DISPLAY %{REG_NAME} write of %{VALUE} includes attempt to send %{CPU_NS:(non-secure|secure)} SGI ID %{SGI_ID} to target number %{TARGET_NO:d}, which is configured to receive %{CPU_NS:(secure|non-secure)} SGI interrupts. Fields:

CPU_NS bool

The destination CPU is configured to accept non-secure SGIs.

REG_NAME string

Name of the register being written.

SGI_ID unsigned int

Interrupt ID for the SGI which is attempting to be sent.

SOURCE_NO unsigned int

GICv2 processor number for the source CPU.

TARGET_NO unsigned int

GICv2 processor number for the destination CPU.

VALUE unsigned int

Value being written to the register.

ArchMsg.Warning.GICv3_MemoryMapped_SPIAssignedToView0

DISPLAY Attempt to assign SPI (%{SPI_ID}) to view0, which is an odd usecase. Fields:

SPI_ID unsigned int

The SPI ID.

ArchMsg.Warning.GICv3_MemoryMapped_SPISpecifiedOutOfRange

DISPLAY %{READnWRITE:(Write to|Read from)} %{REG_NAME} ignored as specified interrupt ID (%{INTERRUPT_ID}) is not a valid SPI. Fields:

INTERRUPT_ID unsigned int

The interrupt ID specified in the register access.

READnWRITE bool

The access in question is a read rather than a write.

REG_NAME string

Name of the register being accessed.

ArchMsg.Warning.GICv3_MemoryMapped_SecureSPIRegWrittenWhenSecurityDisabled

DISPLAY %{REG_NAME} was written when security is disabled; attempt to update SPI %{INTERRUPT_ID} is ignored. Fields:

INTERRUPT_ID unsigned int

The SPI attempting to be set/cleared.

REG_NAME string

Name of the register being accessed.

ArchMsg.Warning.GICv3_MemoryMapped_SkipTriggeringSPI

DISPLAY Skip propagating SPI %{INTERRUPT_ID} down to redistributors, as for GIC700 when there is no GCI, there are no redistributors. Fields:

INTERRUPT_ID unsigned int

The interrupt ID specified in the register access.

ArchMsg.Warning.GICv3_ReceivedInvalidCommandFromBelow

DISPLAY Received a GICv3 command type %{TYPE} of length %{ACTUAL_LENGTH} bytes from downstream port %{REDISTRIBUTOR_INDEX:d}; %{EXPECTED_LENGTH} bytes was expected. Fields:

ACTUAL_LENGTH unsigned int

Actual length of the data in bytes including the header.

DATA unsigned int

The data received (including header).

EXPECTED_LENGTH unsigned int

The length in bytes including header that was expected to be received dependent on command type and potentially the data type in the case of a data write.

REDISTRIBUTOR_INDEX unsigned int

Index for the downstream port that this command was received on.

TYPE unsigned int

The command type.

ArchMsg.Warning.GICv3_SGIDiscardedDueToInvalidTarget

DISPLAY SGI %{INTERRUPT_ID} targetted at %{A3:d}.*%{A2:d}.*%{A1:d}.* with target list %{TARGET_LIST:x} has been discarded at the top level because redistributor with affinity %{A3:d}.* is not connected. Fields:

A1 unsigned int

Affinity level 1.

A2 unsigned int

Affinity level 2.

A3 unsigned int

Affinity level 3.

INTERRUPT_ID unsigned int

The ID of the SGI.

TARGET_LIST unsigned int

Bit vector of targets.

ArchMsg.Warning.GICv3_SPI_OUT_OF_RANGE_ONLINE

DISPLAY The asserted wire %{WIRE_INDEX} corresponding to local SPI ID %{SPI_ID} cannot be mapped to a suitable multichip SPI ID when the chip's SPI range is [%{START_SPI_ID}, %{END_SPI_ID}] inclusive. Calculated SPIs are %{NUM_SPI} and extended SPIs are %{NUM_EXTENDED_SPI} for this chip. Fields:

END_SPI_ID unsigned int

End of SPI ID assigned to this distributor for multichip operation.

NUM_EXTENDED_SPI unsigned int

Number of SPIs in extended spi range for multichip operation.

NUM_SPI unsigned int

Number of SPIs in normal range for multichip operation.

SPI_ID unsigned int

The interrupt ID for the SPI from wire.

START_SPI_ID unsigned int

Beginning of SPI ID assigned to this distributor for multichip operation.

WIRE_INDEX unsigned int

SPI wire index.

ArchMsg.Warning.ReportedRevisionRegisterRead

DISPLAY Reported revision and patch do not match that of the cpu. Fields:

ORIGINAL_VALUE unsigned int

The value the register would have based on the modelled revision.

REGISTER string

The name of the register that has been read.

REPORTED_VALUE unsigned int

The value the register has been overridden to.

GICv3_AccessGICD_IVIEWR_Register

DISPLAY %{OPERATION} value '%{VALUE}' of GICD_IVIEWR from view %{VIEW}. Fields:

OPERATION string

read/write operation.

VALUE unsigned int

The value of GICD_IVIEWR.

VIEW unsigned int

The accessor view.

GICv3_CFGSDISABLE

DISPLAY CFGSDISABLE signal has been %{STATE:(clear|set)}ed. Fields:

STATE bool

The new state of the signal.

GICv3_Distributor.MemoryMapped_MultipleViews_Access

DISPLAY GICv3 %{OPERATION} memory access on address %{ADDRESS} from view %{VIEW}. Fields:

ADDRESS unsigned int

The address of the register being accessed.

OPERATION string

read/write operation.

VIEW unsigned int

The view being accessed.

GICv3_Distributor.MemoryMapped_Read

Trace read from a GICv3 memory-mapped register. Fields:

A0 unsigned int

Affinity level 0 (for banked registers).

A1 unsigned int

Affinity level 1 (for banked registers).

A2 unsigned int

Affinity level 2 (for banked registers).

A3 unsigned int

Affinity level 3 (for banked registers).

BANKED bool

Whether the access is to a banked register.

NS bool

Access is non-secure.

OFFSET unsigned int

Offset of address within the GICv3 module.

REG_NAME string

Name of the register being read.

VALUE unsigned int

Value returned.

GICv3_Distributor.MemoryMapped_Read64

Trace read from a GICv3 memory-mapped register. Fields:

A0 unsigned int

Affinity level 0 (for banked registers).

A1 unsigned int

Affinity level 1 (for banked registers).

A2 unsigned int

Affinity level 2 (for banked registers).

A3 unsigned int

Affinity level 3 (for banked registers).

BANKED bool

Whether the access is to a banked register.

NS bool

Access is non-secure.

OFFSET unsigned int

Offset of address within the GICv3 module.

REG_NAME string

Name of the register being read.

VALUE unsigned int

Value returned.

GICv3_Distributor.MemoryMapped_Write

Trace write to a GICv3 memory-mapped register. Fields:

A0 unsigned int

Affinity level 0 (for banked registers).

A1 unsigned int

Affinity level 1 (for banked registers).

A2 unsigned int

Affinity level 2 (for banked registers).

A3 unsigned int

Affinity level 3 (for banked registers).

BANKED bool

Whether the access is to a banked register.

NS bool

Access is non-secure.

OFFSET unsigned int

Offset of address within the GICv3 module.

REG_NAME string

Name of the register being written.

UPDATED_VALUE unsigned int

New value read back from the register; for write-only registers this will be equal to VALUE.

VALUE unsigned int

Value written.

GICv3_Distributor.MemoryMapped_Write64

Trace write to a GICv3 memory-mapped register. Fields:

A0 unsigned int

Affinity level 0 (for banked registers).

A1 unsigned int

Affinity level 1 (for banked registers).

A2 unsigned int

Affinity level 2 (for banked registers).

A3 unsigned int

Affinity level 3 (for banked registers).

BANKED bool

Whether the access is to a banked register.

NS bool

Access is non-secure.

OFFSET unsigned int

Offset of address within the GICv3 module.

REG_NAME string

Name of the register being written.

UPDATED_VALUE unsigned int

New value read back from the register; for write-only registers this will be equal to VALUE.

VALUE unsigned int

Value written.

GICv3_Distributor.MemoryMapped_WritePort

Trace write to a GICv3 memory-mapped register acting as imp def port. Fields:

FIRST_BYTE_IN_BURST unsigned int

First byte written in the burst.

NS bool

Access is non-secure.

OFFSET unsigned int

Offset of address within the GICv3 module.

REG_NAME string

Name of the register being written.

GICv3_Distributor_Comms_Receive

GICv3 internal communications packet that has been received and is traveling towards the top-level. Fields:

COMMAND enum

The command type for this packet.

DATA unsigned int

The data (including header) contained within this packet.

DATA_LENGTH unsigned int

Number of bytes in the data (including header).

INDEX unsigned int

Index for the redistributor at the lower level from which this packet is being sent. Not applicable for sends from CPU interface to RD0.

GICv3_Distributor_Comms_Send

GICv3 internal communications packet that is being sent towards the CPU. Fields:

COMMAND enum

The command type for this packet.

DATA unsigned int

The data (including header) contained within this packet.

DATA_LENGTH unsigned int

Number of bytes in the data (including header).

INDEX unsigned int

Index for the redistributor at the lower level from which this packet is being sent. Not applicable for sends from CPU interface to RD0.

GICv3_FoundActiveSPI

DISPLAY SPI (%{INTERRUPT_ID}) is active. Fields:

INTERRUPT_ID unsigned int

The interrupt ID.

GICv3_InterruptSetPendingButGroupNotEnabled

DISPLAY Interrupt %{INTERRUPT_ID} was set pending but not forwarded to any CPU as the corresponding interrupt group is not enabled (%{NS:(secure|non-secure)} %{GROUP1:(0|1)}). Fields:

GROUP1 bool

Whether the interrupt is group 1.

INTERRUPT_ID unsigned int

The interrupt ID that was set pending.

NS bool

Whether the interrupt is non-secure.

GICv3_InterruptSetPendingButNoDestinations

DISPLAY Interrupt %{INTERRUPT_ID} was set pending but not forwarded to any CPU as no destinations are configured for this interrupt. Fields:

INTERRUPT_ID unsigned int

The interrupt ID that was set pending.

GICv3_InterruptSetPendingButNoPoweredDestinations

DISPLAY Interrupt %{INTERRUPT_ID} was set pending but not forwarded to any CPU as no destinations have a powered redistributor. Fields:

INTERRUPT_ID unsigned int

The interrupt ID that was set pending.

GICv3_InterruptSetReservedGroup

DISPLAY Interrupt %{INTERRUPT_ID} was sent to be pended but has a reserved group (group == 1 && grpmod == 1); interrupt is treated as non-secure group 1. Fields:

INTERRUPT_ID unsigned int

The interrupt ID that was set pending.

GICv3_Multichip_SGIBroadcast_ToALLCores

DISPLAY Broadcast SGI to all cores for Interrupt ID (%{INTERRUPT_ID}). Fields:

INTERRUPT_ID unsigned int

The interrupt ID specified in the activity being traced.

GICv3_Multichip_SGIToCore

DISPLAY Broadcast SGI to core ID %{TARGET_LIST:x} for Interrupt ID (%{INTERRUPT_ID}). Fields:

INTERRUPT_ID unsigned int

The interrupt ID specified in the activity being traced.

TARGET_LIST unsigned int

Bit vector of targets.

GICv3_PIDR

DISPLAY SW accessing %{POS} %{REG_PREFIX}_PIDR<{%INDEX}> register for view %{VIEW_ID}. Fields:

INDEX unsigned int

The index of the register.

POS string

high/low.

REG_PREFIX string

The register prefix.

VIEW_ID unsigned int

The view-id.

GICv3_RegUpdated32

DISPLAY %{REG_NAME} has been updated to %{VALUE} by a system event. Fields:

REG_NAME string

The name of the register being updated.

VALUE unsigned int

The updated value.

GICv3_RegUpdated64

DISPLAY %{REG_NAME} has been updated to %{VALUE} by a system event. Fields:

REG_NAME string

The name of the register being updated.

VALUE unsigned int

The updated value.

GICv3_SPIAssignedToView

DISPLAY Assign SPI %{SPI_ID} to view %{VIEW}. Fields:

SPI_ID unsigned int

The SPI ID.

VIEW unsigned int

The view which owns the spi.

GICv3_SPISignalChanged

DISPLAY SPI signal for interrupt ID %{INTERRUPT_ID} is now %{STATE:(clear|set)}. Fields:

INTERRUPT_ID unsigned int

The interrupt ID for the SPI.

STATE bool

The new state of the signal.

GICv3_TraceWakeRequest

DISPLAY wake_request signal for interface %INTERFACE has been %{SETnCLEAR:(set|cleared)}. Fields:

INTERFACE unsigned int

Index of the interface.

SETnCLEAR bool

Whether the signal is set(true) or cleared (false).

2.111 GICv3InterruptTranslationService

This section describes the trace sources.

ArchMsg.Error.GICv3_GITS_BASER_NotProvisionedAtEnable

DISPLAY The register GITS_BASER%{BASER_INDEX:d} for type %{TYPE} should be valid at enable. The behaviour of the IRI is now unpredictable. Fields:

BASER_INDEX unsigned int

Index of the BASER register that should have been provisioned.

BASER_TYPE enum

The type of the BASER register that needs to have been provisioned.

ArchMsg.Warning.GICv3_CBASERUnpredAddress

DISPLAY GICTS_CBASER written with value %{VALUE}, UNPREDICATABLE as bit 15:12 are not zero, effective value forced with 15:12 as 0. Fields:

VALUE unsigned int

Value attempting to be written.

ArchMsg.Warning.GICv3_CFGSDISABLE_unsupported

DISPLAY CFGSDISABLE signal has been set but doing so has no effect.

ArchMsg.Warning.GICv3_CWRITER_Write

DISPLAY The offset value %{OFFSET} written to CWRITER is out of range. Fields:

OFFSET unsigned int

Offset written to CWRITE register.

ArchMsg.Warning.GICv3_CommandInvalidEncoding

DISPLAY The data returned for the ITS command from address %{ADDRESS} is not a valid ITS command and has been ignored. Fields:

ADDRESS unsigned int

Address that the command was read from.

DATA unsigned int

Data for the command.

ArchMsg.Warning.GICv3_CommandReadError

DISPLAY Got a read error attempting to read an ITS command from address %{ADDRESS}. Fields:

ADDRESS unsigned int

Address that the command was attempted to be read from.

ArchMsg.Warning.GICv3_DISCARDInvalidDevice

DISPLAY DISCARD command requests that interrupt ID %{ID} from device %{DEVICE_ID} should be silently discarded, but device is not mapped. Fields:

DEVICE_ID unsigned int

Device ID for interrupt source.

ID unsigned int

Interrupt ID that should be discarded.

ArchMsg.Warning.GICv3_DISCARDInvalidID

DISPLAY DISCARD command requests that interrupt ID %{ID} from device %{DEVICE_ID} should be silently discarded, but ID is not mapped for that device. Fields:

DEVICE_ID unsigned int

Device ID for interrupt source.

ID unsigned int

Interrupt ID that should be discarded.

ITT_ADDRESS unsigned int

The ITT address found in the device table.

ArchMsg.Warning.GICv3_GITS_BASER_AddressAligmentError

DISPLAY GITS_BASER%{BASER_INDEX:d} was configured to use %{PAGE_SIZE_KB}kB pages with base address : %{BASE_ADDRESS} which is not page size aligned. Fields:

BASER_INDEX unsigned int

Index of the BASER register that should have been provisioned.

BASE_ADDRESS unsigned int

The base address for the table.

PAGE_SIZE_KB unsigned int

The size of a page in kB.

ArchMsg.Warning.GICv3_GITS_BASER_CollectionTableLargerThanSupportedCollectionSpan

DISPLAY GITS_BASER%{BASER_INDEX:d} (%{BASER_TYPE}) provisions %(PROVISIONED_PAGE_COUNT) but this ITS can only use %{MAX_NEEDED_PAGE_COUNT} pages to hold %{COLECTION_COUNT_IN_MEM} collection entries of size %{COLLECTION_ENTRY_SIZE}. Fields:

BASER_INDEX unsigned int

Index of the BASER register that should have been provisioned.

BASER_TYPE enum

The type of the BASER register that needs to have been provisioned.

COLECTION_COUNT_IN_MEM unsigned int

The number of collections supported by this ITS.

COLLECTION_ENTRY_SIZE unsigned int

The size in bytes of a collection entry.

MAX_NEEDED_PAGE_COUNT unsigned int

The number of pages needed to hold all memory based supported collections.

PROVISIONED_PAGE_COUNT unsigned int

The number of provisioned pages.

ArchMsg.Warning.GICv3_GITS_BASER_InsufficientMemory

DISPLAY %{NAME} command was ignored because the GITS_BASER%{BASER_INDEX:d} register (%{BASER_TYPE}) has not been provisioned with sufficient memory to cope with ID %{ID}. Fields:

BASER_INDEX unsigned int

Index of the BASER register that should have been provisioned.

BASER_TYPE enum

The type of the BASER register that needs to have been provisioned.

ID unsigned int

ID being used to store information.

NAME string

Command or register name that caused device to be used.

ArchMsg.Warning.GICv3_GITS_BASER_InvalidIndirectEntry

DISPLAY %{NAME} command was ignored because the GITS_BASER%{BASER_INDEX:d} register (%{BASER_TYPE}) is indirect and the indirect entry is not valid for ID %{ID}. Fields:

BASER_INDEX unsigned int

Index of the BASER register that should have been provisioned.

BASER_TYPE enum

The type of the BASER register that needs to have been provisioned.

ID unsigned int

ID being used to store information.

NAME string

Command or register name that caused device to be used.

ArchMsg.Warning.GICv3_GITS_BASER_NotProvisioned

DISPLAY %{NAME} command/register write was ignored because the GITS_BASER %{BASER_INDEX:d} register has not been provisioned. Fields:

BASER_INDEX unsigned int

Index of the BASER register that should have been provisioned.

BASER_TYPE enum

The type of the BASER register that needs to have been provisioned.

NAME string

Command or register name that caused device to be used.

ArchMsg.Warning.GICv3_GITS_CTLR_UnpredWrite

DISPLAY GITS_CTLR write with value %{VALUE} when Enable=%{ENABLE} and Quiescent=%{QUIESCENT} changes value of ITSNumber (currently %{ITSNUM}) at a time when doing so makes the behaviour of the system **UNPREDICTABLE**. Fields:

ENABLE bool

Value of the current GITS_CTLR.Enable.

ITSNUM unsigned int

Current value of GITS_CTLR.ITSNumber.

QUIESCENT bool

Value of the current GITS_CTLR.Quiescent.

VALUE unsigned int

Value being written to the register, after masking.

ArchMsg.Warning.GICv3_INVALLInvalid

DISPLAY INVALL command requests that all configuration for all interrupts in collection %{COLLECTION_ID} should be invalidated but collection is not mapped. Fields:

COLLECTION_ID unsigned int

Collection ID to be invalidated.

ArchMsg.Warning.GICv3_INVInvalidCollection

DISPLAY INV command requests that configuration for interrupt ID %{ID} from device %{DEVICE_ID} should be invalidated, but ID is not mapped to a mapped collection. Fields:

COLLECTION_ID unsigned int

Collection ID that cannot be found.

DEVICE_ID unsigned int

Device ID for interrupt source.

ID unsigned int

Interrupt ID that should be invalidated.

ITT_ADDRESS unsigned int

The ITT address found in the device table.

ArchMsg.Warning.GICv3_INVInvalidDevice

DISPLAY INV command requests that configuration for interrupt ID %{ID} from device %{DEVICE_ID} should be invalidated, but device is not mapped. Fields:

DEVICE_ID unsigned int

Device ID for interrupt source.

ID unsigned int

Interrupt ID that should be inved.

ArchMsg.Warning.GICv3_INVInvalidID

DISPLAY INV command requests that configuration for interrupt ID %{ID} from device %{DEVICE_ID} should be invalidated, but ID is not mapped for that device. Fields:

DEVICE_ID unsigned int

Device ID for interrupt source.

ID unsigned int

Interrupt ID that should be invalidated.

ITT_ADDRESS unsigned int

The ITT address found in the device table.

ArchMsg.Warning.GICv3_ITS.MemoryMapped_IgnoredRegisterAccessDueToViewMismatch

DISPLAY Attempt to access register %{REGISTER_NAME} from View %{VIEW_ID} is ignored, because this register is accessible only from View %{OWNER_VIEW_ID}. Fields:

OWNER_VIEW_ID unsigned int

The view owning the register.

REGISTER_NAME string

Name of the register being accessed.

VIEW_ID unsigned int

The view trying to access the register.

ArchMsg.Warning.GICv3_ITS.MemoryMapped_ReadIgnored

DISPLAY GICv3 attempting to read location %{OFFSET} but ignored and RAZ. Fields:

OFFSET unsigned int

Offset of address within the GICv3 module.

ArchMsg.Warning.GICv3_ITS.MemoryMapped_ReadReserved

DISPLAY GICv3 reserved location %{OFFSET} attempting to be read. Fields:

OFFSET unsigned int

Offset of address within the GICv3 module.

ArchMsg.Warning.GICv3_ITS.MemoryMapped_ReadWriteOnlyReg

DISPLAY GICv3 write-only register %{REG_NAME} attempting to be read. Fields:

OFFSET unsigned int

Offset of address within the GICv3 module.

REG_NAME string

Name of the register being read.

ArchMsg.Warning.GICv3_ITS.MemoryMapped_WriteIgnored

DISPLAY GICv3 location %{OFFSET} attempting to be written with value %{VALUE} but ignored. Fields:

OFFSET unsigned int

Offset of address within the GICv3 module.

ArchMsg.Warning.GICv3_ITS.MemoryMapped_WriteReadOnlyReg

DISPLAY GICv3 read-only register %{REG_NAME} attempting to be written with value %{VALUE}. Fields:

OFFSET unsigned int

Offset of address within the GICv3 module.

REG_NAME string

Name of the register being written.

VALUE unsigned int

Value written.

ArchMsg.Warning.GICv3_ITS.MemoryMapped_WriteReserved

DISPLAY GICv3 reserved location %{OFFSET} attempting to be written with value %{VALUE}. Fields:

OFFSET unsigned int

Offset of address within the GICv3 module.

VALUE unsigned int

Value written.

ArchMsg.Warning.GICv3_ITS.TranslateDeviceOutOfRangeIgnored

DISPLAY GITS_TRANSLATER for device %{DEVICE_ID} ignored as only %{DEVICE_ID_BITS} bits are supported. Fields:

DEVICE_ID unsigned int

Device ID to be mapped.

DEVICE_ID_BITS unsigned int

Device ID bits supported.

ArchMsg.Warning.GICv3_ITS.TranslateEventOutOfRangeIgnored

DISPLAY GITS_TRANSLATER for event %{DEVICE_ID}, %{EVENT_ID} ignored only %{EVENT_ID_BITS} bits are supported. Fields:

DEVICE_ID unsigned int

ID of the device triggering the event.

EVENT_ID unsigned int

Event ID to be mapped.

EVENT_ID_BITS unsigned int

Event ID bits supported.

ArchMsg.Warning.GICv3_ITS_TranslateRequestIgnored

DISPLAY GITS_TRANSLATER for device %{DEVICE_ID} and event %{EVENT_ID} was not successful. Syndrome=%{SYNDROME}. Fields:

DEVICE_ID unsigned int

Device ID to be mapped.

EVENT_ID unsigned int

Event ID to be mapped.

SYNDROME enum

Error Syndrome.

ArchMsg.Warning.GICv3_InvalidDeviceMap

DISPLAY %{CMD_NAME} command requesting translation for device ID %{DEVICE_ID} with Interrupt ID %{ID} that has not been previously mapped. Fields:

CMD_NAME string

The name of ITS Command that triggered the translation.

DEVICE_ID unsigned int

Non-mapped device ID specified in the command.

ID unsigned int

Incoming interrupt ID specified in the command.

ArchMsg.Warning.GICv3_InvalidTargetAddress

DISPLAY Command %{COMMAND_NAME} was dropped because it attempted to use an invalid processor number %{PROCESSOR_NUM} in a monolithic GIC distributor configuration. Fields:

COMMAND_NAME string

Name of the command being executed.

PROCESSOR_NUM unsigned int

Processor number specified in the command.

ArchMsg.Warning.GICv3_InvalidVCPU

DISPLAY %{CMD_NAME} command requesting map info for VCPU %{VCPU} that has not been previously mapped using VMAPP. Fields:

CMD_NAME string

The name of ITS Command that requested the map.

VCPU unsigned int

Virtual CPU specified in the command.

ArchMsg.Warning.GICv3_LPIUnmappedCollection

DISPLAY Request to %{SET:(clear|set)} LPI for device %{DEVICE_ID} with ID %{ID} is ignored as the associated collection (%{COLLECTION_ID}) is not mapped. Fields:

COLLECTION_ID unsigned int

Collection ID that the interrupt maps to.

DEVICE_ID unsigned int

Device ID for the interrupt.

ID unsigned int

Incoming interrupt ID specified.

ITT_ADDRESS unsigned int

The ITT address found in the device table.

SET bool

Whether the interrupt is set (rather than cleared).

ArchMsg.Warning.GICv3_LPIUnmappedDevice

DISPLAY Request to %{SET:(clear|set)} LPI for device %{DEVICE_ID} is ignored as the device is not mapped. Fields:

DEVICE_ID unsigned int

Device ID for the interrupt.

ID unsigned int

Incoming interrupt ID specified.

SET bool

Whether the interrupt is set (rather than cleared).

ArchMsg.Warning.GICv3_LPIUnmappedID

DISPLAY Request to %{SET:(clear|set)} LPI for device %{DEVICE_ID} with ID %{ID} is ignored as the ID is not mapped. Fields:

DEVICE_ID unsigned int

Device ID for the interrupt.

ID unsigned int

Incoming interrupt ID specified.

ITT_ADDRESS unsigned int

The ITT address found in the device table.

SET bool

Whether the interrupt is set (rather than cleared).

ArchMsg.Warning.GICv3_MAPCUnmapInvalid

DISPLAY MAPC command request unmapping collection ID %{COLLECTION_ID} that was not previous mapped. Fields:

COLLECTION_ID unsigned int

Collection ID requested to be unmapped.

ArchMsg.Warning.GICv3_MAPCUnmapInvalidTarget

DISPLAY MAPC command request unmapping with non-zero target value.

ArchMsg.Warning.GICv3_MAPDUnmapInvalid

DISPLAY MAPD command request unmapping device ID %{DEVICE_ID} that was not previously mapped. Fields:

DEVICE_ID unsigned int

Device ID attempting to be unmapped.

ArchMsg.Warning.GICv3_MAPDUnmapInvalidSize

DISPLAY MAPD command request unmapping with non-zero Size value.

ArchMsg.Warning.GICv3_MAPIInvalid

DISPLAY MAPI/MAPT command requesting mapping for device ID %{DEVICE_ID} that has not been previously mapped with MAPD (ID: %{ID}, physical ID: %{PHYSICAL_ID}; COLLECTION: %{COLLECTION_ID}). Fields:

COLLECTION_ID unsigned int

Collection ID specified in the map request.

DEVICE_ID unsigned int

Non-mapped device ID specified in the map request.

ID unsigned int

Incoming interrupt ID specified in the map request.

PHYSICAL_ID unsigned int

Outgoing interrupt ID specified in the map request.

ArchMsg.Warning.GICv3_MOVIInvalidCollection

DISPLAY MOVI command requests moving interrupt ID %{ID} with device %{DEVICE_ID} from collection %{OLD_COLLECTION_ID} to %{NEW_COLLECTION_ID} but the %{NEW_INVALID:(old|new)} collection ID is not mapped. Fields:

DEVICE_ID unsigned int

Device ID for interrupt source.

ID unsigned int

Input interrupt ID that should be moved.

ITT_ADDRESS unsigned int

The ITT address found in the device table.

NEW_COLLECTION_ID unsigned int

Collection ID that was requested.

NEW_INVALID bool

It is the new collection ID that is invalid (rather than the old collection ID).

OLD_COLLECTION_ID unsigned int

Collection ID that this interrupt used to be associated with.

ArchMsg.Warning.GICv3_MOVIInvalidDevice

DISPLAY MOVI command requests moving interrupt ID %{ID} with device %{DEVICE_ID} to collection %{NEW_COLLECTION_ID} but the device is not currently mapped. Fields:

DEVICE_ID unsigned int

Device ID for interrupt source.

ID unsigned int

Input interrupt ID that should be moved.

NEW_COLLECTION_ID unsigned int

Collection ID that was requested.

ArchMsg.Warning.GICv3_MOVIInvalidID

DISPLAY MOVI command requests moving interrupt ID %{ID} with device %{DEVICE_ID} to collection %{NEW_COLLECTION_ID} but the ID given is not currently mapped. Fields:

DEVICE_ID unsigned int

Device ID for interrupt source.

ID unsigned int

Input interrupt ID that should be moved.

ITT_ADDRESS unsigned int

The ITT address found in the device table.

NEW_COLLECTION_ID unsigned int

Collection ID that was requested.

ArchMsg.Warning.GICv3_RangeErrorCollection

DISPLAY %{NAME} command was ignored because the collection ID %{COLLECTION_ID} is out of the supported range. Fields:

COLLECTION_ID unsigned int

Collection ID specified.

NAME string

Command or register name.

ArchMsg.Warning.GICv3_RangeErrorDeviceID

DISPLAY %{NAME} command was ignored because the device ID specified (%{DEVICE_ID}) is out of the supported range. Fields:

DEVICE_ID unsigned int

Device ID specified.

NAME string

Command or register name.

ArchMsg.Warning.GICv3_RangeErrorIDBits

DISPLAY %{NAME} command was ignored because the number of ID bits specified (%{ID_BITS}) is out of the supported range. Fields:

ID_BITS unsigned int

Number of ID bits specified.

NAME string

Command or register name.

ArchMsg.Warning.GICv3_RangeErrorIncomingID

DISPLAY %{NAME} command was ignored because the incoming interrupt ID %{ID} is out of the supported range (max %{ID_BITS:d} bits). Fields:

COLLECTION_ID unsigned int

Collection ID specified.

ID_BITS unsigned int

Maximum number of ID bits allowed for this access.

NAME string

Command or register name.

ArchMsg.Warning.GICv3_WriteIgnoredConfiguration

DISPLAY %{REG_NAME} write of value %{VALUE} ignored due to GICv3 configuration. Fields:

REG_NAME string

Name of register being read or written.

VALUE unsigned int

Value attempting to be written.

GICv3_CFGSDISABLE

DISPLAY CFGSDISABLE signal has been %{STATE:(clear|set)}ed. Fields:

STATE bool

The new state of the signal.

GICv3_CommandComplete

DISPLAY ITS %{COMMAND_TYPE} %{SUCCESS:(FAILED|SUCCEEDED)}. Fields:

COMMAND_TYPE enum

Type of the command being executed.

ERROR enum

The error type if the command was not successful.

SUCCESS bool

Whether the command succeeded.

GICv3_CommandDecode

DISPLAY DECODE %{OFFSET} %{DATA_0}:%{DATA_1}:%{DATA_2}:%{DATA_3}
%{COMMAND_NAME}. Fields:

COMMAND_NAME string

Name of the decoded command.

DATA_0 unsigned int

First 64-bit word from the command.

DATA_1 unsigned int

Second 64-bit word from the command.

DATA_2 unsigned int

Third 64-bit word from the command.

DATA_3 unsigned int

Forth 64-bit word from the command.

OFFSET unsigned int

Offset from GITS_CBASER that the command was read from.

VALID bool

Whether the command was valid.

GICv3_CommandStart_CLEAR

DISPLAY ITS CLEAR DEVICE:%{DEVICE} ID:%{ID}. Fields:

DEVICE unsigned int

The Device ID specified in the command.

ID unsigned int

The incoming ID of interrupts to move to a new collection.

GICv3_CommandStart_DISCARD

DISPLAY ITS DISCARD DEVICE:%{DEVICE} ID:%{ID}. Fields:

DEVICE unsigned int

The Device ID specified in the command.

ID unsigned int

The incoming ID of interrupts to discard.

GICv3_CommandStart_INT

DISPLAY ITS INT DEVICE:%{DEVICE} ID:%{ID}. Fields:

DEVICE unsigned int

The Device ID specified in the command.

ID unsigned int

The incoming ID of interrupts to move to a new collection.

GICv3_CommandStart_INV

DISPLAY ITS INV DEVICE:%{DEVICE} ID:%{ID}. Fields:

DEVICE unsigned int

The Device ID specified in the command.

ID unsigned int

The incoming ID of interrupts to invalidate caches for.

GICv3_CommandStart_INVALID

DISPLAY ITS INVALID COLLECTION:%{COLLECTION}. Fields:

COLLECTION unsigned int

The collection for which all interrupt caches should be invalidated.

GICv3_CommandStart_MAPC

DISPLAY ITS MAPC VALID:%{VALID:(0|1)} COLLECTION:%{COLLECTION} TARGET:%{TARGET}. Fields:

COLLECTION unsigned int

The ID of the collection being mapped.

TARGET unsigned int

The processor number or redistributor base address for the redistributor to synchronise.

VALID bool

Whether to create a new device mapping (true) or to discard an old one (false).

GICv3_CommandStart_MAPD

DISPLAY ITS MAPD VALID:%{VALID:(0|1)} DEVICE:%{DEVICE} ITT:%{ITT_ADDRESS} SIZE:%{SIZE}. Fields:

DEVICE unsigned int

The Device ID specified in the command.

ITT_ADDRESS unsigned int

The interrupt translation table address specified in the command.

SIZE unsigned int

Number of interrupt ID bits supported by this device, minus one.

VALID bool

Whether to create a new device mapping (true) or to discard an old one (false).

GICv3_CommandStart_MAPI

DISPLAY ITS MAPI DEVICE:%{DEVICE} ID:%{ID} COLLECTION:%{COLLECTION}. Fields:

COLLECTION unsigned int

The ID of the collection to map the interrupts to.

DEVICE unsigned int

The Device ID specified in the command.

ID unsigned int

The incoming ID of interrupts to map (and the same ID that will be passed to software).

GICv3_CommandStart_MAPTI

DISPLAY ITS MAPTI DEVICE:%{DEVICE} ID:%{ID} pID:%{PHYSICAL_ID} COLLECTION:%{COLLECTION}. Fields:

COLLECTION unsigned int

The ID of the collection to map the interrupts to.

DEVICE unsigned int

The Device ID specified in the command.

ID unsigned int

The incoming ID of interrupts to map.

PHYSICAL_ID unsigned int

The ID presented to software for these interrupts.

GICv3_CommandStart_MOVALL

DISPLAY ITS MOVALL SOURCE:%{SOURCE} TARGET:%{TARGET}. Fields:

SOURCE unsigned int

The processor number or redistributor base address for the source redistributor.

TARGET unsigned int

The processor number or redistributor base address for the target redistributor.

GICv3_CommandStart_MOVI

DISPLAY ITS MOVI DEVICE:%{DEVICE} ID:%{ID} COLLECTION:%{COLLECTION}. Fields:

COLLECTION unsigned int

The ID of the new collection to move the interrupt to.

DEVICE unsigned int

The Device ID specified in the command.

ID unsigned int

The incoming ID of interrupts to move to a new collection.

GICv3_CommandStart_SYNC

DISPLAY ITS CLEAR TARGET:%{TARGET}. Fields:

TARGET unsigned int

The processor number or redistributor base address for the redistributor to synchronise.

GICv3_CommandStart_VINVALL

DISPLAY ITS VINVALL VCPU:%{VCPU}. Fields:

VCPU unsigned int

The VCPU id for which all interrupt caches should be invalidated.

GICv3_CommandStart_VMAPI

DISPLAY ITS VMAPI DEVICE:%{DEVICE} VCPU:%{VCPU} ID:%{ID} PHYSICAL_ID:%{PHYSICAL_ID}. Fields:

DEVICE unsigned int

The Device ID specified in the command.

ID unsigned int

The incoming ID of interrupt to map.

PHYSICAL_ID unsigned int

The doorbell ID to be presented to guest.

VCPU unsigned int

The VCPU ID specified in the command.

GICv3_CommandStart_VMAPP

DISPLAY ITS VMAPP VALID:%{VALID:(0|1)} VCPU:%{VCPU} TARGET:%{TARGET_ADDRESS} VPT:%{VPT_ADDRESS} SIZE:%{SIZE}. Fields:

SIZE unsigned int

Number of interrupt ID bits supported by this device, minus one.

TARGET_ADDRESS unsigned int

The Redistributor address specified in the command.

VALID bool

Whether to create a new device mapping (true) or to discard an old one (false).

VCPU unsigned int

The VCPU ID specified in the command.

VPT_ADDRESS unsigned int

The Virtual Pending Table address specified in the command.

GICv3_CommandStart_VMAPTI

DISPLAY ITS VMAPTI DEVICE:%{DEVICE} VCPU:%{VCPU} ID:%{ID} VIRTUAL_ID:%{VIRTUAL_ID} PHYSICAL_ID:%{PHYSICAL_ID}. Fields:

DEVICE unsigned int

The Device ID specified in the command.

ID unsigned int

The incoming ID of interrupt to map.

PHYSICAL_ID unsigned int

The doorbell ID to be presented to guest.

VCPU unsigned int

The VCPU ID specified in the command.

VIRTUAL_ID unsigned int

The Virtual ID to be presented to guest.

GICv3_CommandStart_VMOVI

DISPLAY ITS VMOVI DEVICE:%{DEVICE} VCPU:%{VCPU} ID:%{ID} PHYSICAL_ID:%{PHYSICAL_ID}. Fields:

DEVICE unsigned int

The Device ID specified in the command.

ID unsigned int

The incoming ID of interrupt to map.

PHYSICAL_ID unsigned int

The doorbell ID to be presented to guest.

VCPU unsigned int

The VCPU ID specified in the command.

GICv3_CommandStart_VMOVP

DISPLAY ITS VMOVP VCPU:%{VCPU} TARGET:%{TARGET_ADDRESS} SEQUENCE_NUM:%{SEQUENCE_NUM} ITS_LIST:%{ITS_LIST}. Fields:

ITS_LIST unsigned int

The ITS numbers participating in the synchronizatio operation.

SEQUENCE_NUM unsigned int

The identifier of synchronizatio point in the command.

TARGET_ADDRESS unsigned int

The Redistributor address specified in the command.

VCPU unsigned int

The VCPU ID specified in the command.

GICv3_CommandStart_VSYNC

DISPLAY ITS VSYNC VCPU:%{VCPU}. Fields:

VCPU unsigned int

The VCPU id for which commands must be synchronized.

GICv3_DISCARD

DISPLAY DISCARD command requests that interrupt ID %{ID} from device %{DEVICE_ID} should be silently discarded. Fields:

DEVICE_ID unsigned int

Device ID for interrupt source.

ID unsigned int

Interrupt ID that should be discarded.

ITT_ADDRESS unsigned int

The ITT address found in the device table.

GICv3_INV

DISPLAY INV command requests that configuration for interrupt ID %{ID} from device %{DEVICE_ID} should be invalidated. Request sent to redistributor with base address %{REDISTRIBUTOR_BASE} with ID %{PHYSICAL_ID} for collection %{COLLECTION_ID}. Fields:

COLLECTION_ID unsigned int

Collection ID for this interrupt.

DEVICE_ID unsigned int

Device ID for interrupt source.

ID unsigned int

Input interrupt ID that should be invalidated.

ITT_ADDRESS unsigned int

The ITT address found in the device table.

PHYSICAL_ID unsigned int

Interrupt ID that will be specified to the redistributor.

REDISTRIBUTOR_BASE unsigned int

Base address for the redistributor.

GICv3_INVALL

DISPLAY INVALL command requests that all configuration for all interrupts in collection %{COLLECTION_ID} should be invalidated. Request sent to redistributor with base address %{REDISTRIBUTOR_BASE}. Fields:

COLLECTION_ID unsigned int

Collection ID to be invalidated.

REDISTRIBUTOR_BASE unsigned int

Base address for the redistributor.

GICv3_ITS.MemoryMapped_MultipleViews_Access

DISPLAY GICv3 %{OPERATION} memory access on address %{ADDRESS} from view %{VIEW}. Fields:

ADDRESS unsigned int

The address of the register being accessed.

OPERATION string

read/write operation.

VIEW unsigned int

The view being accessed.

GICv3_ITS.MemoryMapped_Read

Trace read from a GICv3 memory-mapped register. Fields:

A0 unsigned int

Affinity level 0 (for banked registers).

A1 unsigned int

Affinity level 1 (for banked registers).

A2 unsigned int

Affinity level 2 (for banked registers).

A3 unsigned int

Affinity level 3 (for banked registers).

BANKED bool

Whether the access is to a banked register.

NS bool

Access is non-secure.

OFFSET unsigned int

Offset of address within the GICv3 module.

REG_NAME string

Name of the register being read.

VALUE unsigned int

Value returned.

GICv3_ITS.MemoryMapped_Read64

Trace read from a GICv3 memory-mapped register. Fields:

A0 unsigned int

Affinity level 0 (for banked registers).

A1 unsigned int

Affinity level 1 (for banked registers).

A2 unsigned int

Affinity level 2 (for banked registers).

A3 unsigned int

Affinity level 3 (for banked registers).

BANKED bool

Whether the access is to a banked register.

NS bool

Access is non-secure.

OFFSET unsigned int

Offset of address within the GICv3 module.

REG_NAME string

Name of the register being read.

VALUE unsigned int

Value returned.

GICv3_ITS.MemoryMapped_Write

Trace write to a GICv3 memory-mapped register. Fields:

A0 unsigned int

Affinity level 0 (for banked registers).

A1 unsigned int

Affinity level 1 (for banked registers).

A2 unsigned int

Affinity level 2 (for banked registers).

A3 unsigned int

Affinity level 3 (for banked registers).

BANKED bool

Whether the access is to a banked register.

NS bool

Access is non-secure.

OFFSET unsigned int

Offset of address within the GICv3 module.

REG_NAME string

Name of the register being written.

UPDATED_VALUE unsigned int

New value read back from the register; for write-only registers this will be equal to VALUE.

VALUE unsigned int

Value written.

GICv3_ITS.MemoryMapped_Write64

Trace write to a GICv3 memory-mapped register. Fields:

A0 unsigned int

Affinity level 0 (for banked registers).

A1 unsigned int

Affinity level 1 (for banked registers).

A2 unsigned int

Affinity level 2 (for banked registers).

A3 unsigned int

Affinity level 3 (for banked registers).

BANKED bool

Whether the access is to a banked register.

NS bool

Access is non-secure.

OFFSET unsigned int

Offset of address within the GICv3 module.

REG_NAME string

Name of the register being written.

UPDATED_VALUE unsigned int

New value read back from the register; for write-only registers this will be equal to VALUE.

VALUE unsigned int

Value written.

GICv3_ITS.MemoryMapped_WritePort

Trace write to a GICv3 memory-mapped register acting as imp def port. Fields:

FIRST_BYTE_IN_BURST unsigned int

First byte written in the burst.

NS bool

Access is non-secure.

OFFSET unsigned int

Offset of address within the GICv3 module.

REG_NAME string

Name of the register being written.

GICv3_ITS_SendRequestToRedistributor

DISPLAY Send ITS request to redistributor with address %{REDISTRIUBTOR_ADDRESS}. Fields:

REDISTRIUBTOR_ADDRESS unsigned int

The address of the target redistributor including the offset.

GICv3_ITS_TranslationStart

DISPLAY Translation for deviceID:%{DEVICE_ID}, eventID:%{EVENT_ID} %{START: (IGNORED|START)}. Fields:

DEVICE_ID unsigned int

Device ID to be mapped.

EVENT_ID unsigned int

Event ID to be mapped.

START bool

Translation Started.

GICv3_ITS_TranslationStatus

DISPLAY Translation for deviceID:%{DEVICE_ID}, eventID:%{EVENT_ID} %{SUCCESS: (IGNORED|SUCCESS)}. Fields:

DEVICE_ID unsigned int

Device ID to be mapped.

EVENT_ID unsigned int

Event ID to be mapped.

SUCCESS bool

Translation Success.

GICv3_ITTAddressTransaction

DISPLAY ITS ITT TRANSACTION ID:%{DEVICE_ID} ADDRESS:%{ITT_ADDRESS}. Fields:

DEVICE_ID unsigned int

The Device ID specified in the command.

ITT_ADDRESS unsigned int

The base address specified in the MAPD command that is being used.

GICv3_ITTAddressTransactionFailed

DISPLAY ITS ITT TRANSACTION FAILED ID:%{DEVICE_ID} ADDRESS:%{ITT_ADDRESS}. Fields:

DEVICE_ID unsigned int

The Device ID specified in the command.

ITT_ADDRESS unsigned int

The base address specified in the MAPD command that is being used.

GICv3_LPISent

DISPLAY LPI %{PHYSICAL_ID} was %{SET:(cleared|set)} in redistributor with base address %{REDISTRIBUTOR_BASE} due to request for device ID %{DEVICE_ID} with ID %{ID} which mapped to collection %{COLLECTION_ID}. Fields:

COLLECTION_ID unsigned int

Collection ID that the interrupt maps to.

DEVICE_ID unsigned int

Device ID for the interrupt.

ID unsigned int

Incoming interrupt ID specified.

ITT_ADDRESS unsigned int

The ITT address found in the device table.

PHYSICAL_ID unsigned int

Outgoing interrupt ID found in the map.

REDISTRIBUTOR_BASE unsigned int

Base address for redistributor (may be an internal address for monolithic implementations).

SET bool

Whether the interrupt is set (rather than cleared).

GICv3_MAPCMap

DISPLAY MAPC command maps collection ID %{COLLECTION_ID} to redistributor at address %{ADDRESS}. Fields:

ADDRESS unsigned int

Redistributor base address.

COLLECTION_ID unsigned int

Collection ID to be mapped.

GICv3_MAPCMapMonolithic

DISPLAY MAPC command maps collection ID %{COLLECTION_ID} to redistributor for processor number %{PROC_NUM:d} (at address %{ADDRESS}). Fields:

ADDRESS unsigned int

Redistributor base address.

COLLECTION_ID unsigned int

Collection ID to be mapped.

PROC_NUM unsigned int

Processor number for redistributor.

GICv3_MAPCUnmap

DISPLAY MAPC command unmaps collection ID %{COLLECTION_ID}. Fields:

COLLECTION_ID unsigned int

Collection ID being unmapped.

GICv3_MAPDMap

DISPLAY MAPD command maps device ID %{DEVICE_ID} to table %{ITT_ADDRESS} supporting %{INTERRUPT_BITS} interrupt bits. Fields:

DEVICE_ID unsigned int

Device ID to be mapped.

INTERRUPT_BITS unsigned int

Number of interrupt ID bits supported by the device.

ITT_ADDRESS unsigned int

Base address for memory to store Interrupt Translation Table.

GICv3_MAPDUnmap

DISPLAY MAPD command request unmapped device ID %{DEVICE_ID}. Fields:

DEVICE_ID unsigned int

Device ID to be unmapped.

GICv3_MAPI

DISPLAY MAPI/MAPT command requests mapping for interrupt ID %{ID} => %{PHYSICAL_ID} for device ID %{DEVICE_ID} using collection %{COLLECTION_ID}. Fields:

COLLECTION_ID unsigned int

Collection ID specified in the map request.

DEVICE_ID unsigned int

Device ID specified in the map request.

ID unsigned int

Incoming interrupt ID specified in the map request.

ITT_ADDRESS unsigned int

The ITT address found in the device table.

PHYSICAL_ID unsigned int

Outgoing interrupt ID specified in the map request.

GICv3_MOVALL

DISPLAY MOVALL command requests moving all interrupt from redistributor at address %{OLD_REDISTRIBUTOR_BASE} to %{NEW_REDISTRIBUTOR_BASE}. Fields:

NEW_REDISTRIBUTOR_BASE unsigned int

Base address (or processor number for monolithic implementation) for destination redistributor.

OLD_REDISTRIBUTOR_BASE unsigned int

Base address (or processor number for monolithic implementation) for source redistributor.

GICv3_MOVI

DISPLAY MOVI command requests moving interrupt ID %{ID} with device %{DEVICE_ID} (physical ID %{PHYSICAL_ID} should be moved from collection %{OLD_COLLECTION_ID} to %{NEW_COLLECTION_ID}. This %{MOVE_REQUIRED:(does not require|requires)} a move between redistributors. Fields:

DEVICE_ID unsigned int

Device ID for interrupt source.

ID unsigned int

Input interrupt ID that was moved.

ITT_ADDRESS unsigned int

The ITT address found in the device table.

MOVE_REQUIRED bool

Whether a move between redistributors was required.

NEW_COLLECTION_ID unsigned int

Collection ID that this interrupt is now associated with.

OLD_COLLECTION_ID unsigned int

Collection ID that this interrupt used to be associated with.

PHYSICAL_ID unsigned int

Interrupt ID that will be specified to the redistributor.

GICv3_MultiChip_Forwarding_LPI_CMD_ACK_to_Dist

DISPLAY ITS on Chip (ID:%{THIS_CHIP_ID}) is forwarding LPI_CMD_ACK to the local Distributor. Fields:

THIS_CHIP_ID unsigned int

Chip .

GICv3_MultiChip_Forwarding_LPI_CMD_REQ_to_Dist

DISPLAY ITS on Chip (ID:%{THIS_CHIP_ID}) is forwarding LPI_CMD_REQ to the local Distributor, along with the following info: CMD_TYPE:%{CMD_TYPE} INTID:%{LPI_ID} SRC_CHIP:%{SRC_CHIP} SRC_CORE:%{SRC_CORE} DST_CHIP:%{DST_CHIP} DST_CORE:%{DST_CORE}. Fields:

CMD_TYPE unsigned int

The command type.

DST_CHIP unsigned int

Destination chip for the command.

DST_CORE unsigned int

Destination core for the command.

LPI_ID unsigned int

LPI interrupt ID being operated on by the command.

SRC_CHIP unsigned int

Source chip for the command.

SRC_CORE unsigned int

Source core for the command.

THIS_CHIP_ID unsigned int

Chip ID.

GICv3_MultiChip_Forwarding_VPE_CTLR_ACK_to_Dist

DISPLAY ITS on Chip (ID:%{THIS_CHIP_ID}) is forwarding VPE_CTLR_ACK to the local Distributor for sending cross-chip. Fields:

THIS_CHIP_ID unsigned int

Chip ID.

GICv3_MultiChip_Forwarding_VPE_CTLR_REQ_to_Dist

DISPLAY ITS on Chip (ID:%{THIS_CHIP_ID}) is forwarding VPE_CTLR_REQ to the local Distributor for sending cross-chip. The information passed is: SRC_CHIP_ID:%{SRC_CHIP_ID} VPEID:%{VPEID} TARGET_CHIP_ID:%{TARGET_CHIP_ID} TARGET_PE:%{TARGET_PE}. Fields:

SRC_CHIP_ID unsigned int

Source Chip ID.

TARGET_CHIP_ID unsigned int

Target Chip ID.

TARGET_PE unsigned int

Target PE on dest chip.

VPEID unsigned int

Identifier of the virtual PE being moved.

GICv3_MultiChip_SETLPI_BY_MOVI

DISPLAY Chip (ID:%{THIS_CHIP_ID}) received MOVI command to move LPI (ID:%{LPI}) from chip (ID:%{SRC_TGT}) to chip (ID:%{DST_TGT}). Fields:

DST_TGT unsigned int

Destination chip and core for MOVI.

LPI unsigned int

LPI to be moved.

SRC_TGT unsigned int

Source chip and core for MOVI.

THIS_CHIP_ID unsigned int

Chip .

GICv3_MultiChip_VMOVP_Received

DISPLAY Chip (ID:%{THIS_CHIP_ID}) received VMOVP command to move vPE (ID:%{vPEID}) from chip (ID:%{SRC_TGT}) to chip (ID:%{DST_TGT}). Fields:

DST_TGT unsigned int

Destination chip and core for VMOVP.

SRC_TGT unsigned int

Source chip and core for VMOVP.

THIS_CHIP_ID unsigned int

Chip .

vPEID unsigned int

ID of the vPE to be moved.

GICv3_PIDR

DISPLAY SW accessing %{POS} %{REG_PREFIX}_PIDR<{%INDEX}> register for view %{VIEW_ID}. Fields:

INDEX unsigned int

The index of the register.

POS string

high/low.

REG_PREFIX string

The register prefix.

VIEW_ID unsigned int

The view-id.

GICv3_RegUpdated32

DISPLAY %{REG_NAME} has been updated to %{VALUE} by a system event. Fields:

REG_NAME string

The name of the register being updated.

VALUE unsigned int

The updated value.

GICv3_RegUpdated64

DISPLAY %{REG_NAME} has been updated to %{VALUE} by a system event. Fields:

REG_NAME string

The name of the register being updated.

VALUE unsigned int

The updated value.

GICv3_TableWalk_Collection

DISPLAY ITS TABLE COL ID:%{ID} TARGET:%{TARGET}. Fields:

ID unsigned int

The ID of the collection being looked up.

TARGET unsigned int

The target address (or processor number) specified for this collection.

GICv3_TableWalk_CollectionInvalid

DISPLAY ITS TABLE COL ID:%{ID}. Fields:

ID unsigned int

The ID of the collection being looked up.

GICv3_TableWalk_Device

DISPLAY ITS TABLE DEV ID:%{DEVICE_ID} ITT:%{ITT_ADDRESS} BITS:
%{INTERRUPT_BITS:d}. Fields:

DEVICE_ID unsigned int

The device ID being looked up.

INTERRUPT_BITS unsigned int

The number of interrupt bits specified in the device table.

ITT_ADDRESS unsigned int

The ITT address found in the device table.

GICv3_TableWalk_DeviceInvalid

DISPLAY ITS TABLE DEV FAULT ID:%{DEVICE_ID}. Fields:

DEVICE_ID unsigned int

The device ID being looked up.

GICv3_TableWalk_ITT

DISPLAY ITS TABLE ITT ID:%{ID} pID:%{PHYSICAL_ID} COLLECTION:%{COLLECTION_ID}. Fields:

COLLECTION_ID unsigned int

The collection ID used for routing this interrupt to the appropriate CPU.

ID unsigned int

The input identifier used for the translation process.

ITT_ADDRESS unsigned int

The base address specified in the previous MAPD command for this table.

PHYSICAL_ID unsigned int

The physical interrupt ID presented to software.

GICv3_TableWalk_ITTInvalid

DISPLAY ITS TABLE ITT FAULT ID:%{ID}. Fields:

ID unsigned int

The input identifier used for the translation process.

ITT_ADDRESS unsigned int

The base address specified in the MAPD command that is being used.

GICv3_TableWalk_ITT_VLPI

DISPLAY ITS TABLE ITT ID:%{ID} pID:%{PHYSICAL_ID} VCPU:%{VCPU}. Fields:

ID unsigned int

The input identifier used for the translation process.

ITT_ADDRESS unsigned int

The base address specified in the previous MAPD command for this table.

PHYSICAL_ID unsigned int

The physical interrupt ID presented to software.

VCPU unsigned int

The VCPU ID used for routing this interrupt to the appropriate CPU.

GICv3_TableWalk_VCPU

DISPLAY ITS VCPU TABLE :%{VCPU} VPT:%{VPT_ADDRESS} BITS:%{VINTERRUPT_BITS:d} TARGET_ADDRESS:%{TARGET}. Fields:

TARGET unsigned int

The Redistributor address specified in VCPU table.

VCPU unsigned int

The VCPU being looked up.

VINTERRUPT_BITS unsigned int

The number of virtual interrupt bits specified in the VCPU table.

VPT_ADDRESS unsigned int

The Virtual Pending Table address assigned to VCPU.

GICv3_TableWalk_VCPUInvalid

DISPLAY ITS TABLE VCPU FAULT ID:%{VCPU}. Fields:

VCPU unsigned int

The Virtual CPU ID being looked up.

GICv3_TableWalk_VPTAddrInvalid

DISPLAY ITS TABLE VPTAddr FAULT ID:%{VPTAddr}. Fields:

VPTAddr unsigned int

The Virtual Pending Table address supposedly corresponding to a VCPU, being looked up.

GICv3_VMAPI

DISPLAY VMAPI/VMAPT command requests mapping for interrupt ID %{ID} => PhysicalID %{PHYSICAL_ID} and VirtualID %{VIRTUAL_ID} for device ID %{DEVICE_ID} using vcpu %{VCPU}. Fields:

DEVICE_ID unsigned int

Device ID specified in the map request.

ID unsigned int

Incoming interrupt ID specified in the map request.

ITT_ADDRESS unsigned int

The ITT address found in the device table.

PHYSICAL_ID unsigned int

Outgoing Physical interrupt ID specified in the map request.

VCPU unsigned int

Virtual CPU specified in the map request.

VIRTUAL_ID unsigned int

Outgoing Virtual interrupt ID specified in the map request.

GICv3_VMAPP_Map

DISPLAY VMAPP command maps VCPU ID %{VCPU} to Redistributor target %{TARGET_ADDRESS} and VPT Address %{VPT_ADDRESS}. Fields:

TARGET_ADDRESS unsigned int

The Redistributor address assigned to VCPU.

VCPU unsigned int

VCPU ID to be mapped.

VPT_ADDRESS unsigned int

The Virtual Pending Table address assigned to VCPU.

GICv3_VMAPP_Unmap

DISPLAY VMAPP command request unmapped VCPU %{VCPU}. Fields:

VCPU unsigned int

Virtual CPU being unmapped.

GICv3_VMOVI

DISPLAY VMOVI command requests moving virtual interrupt ID %{ID} with device %{DEVICE_ID} from VCPU %{OLD_VCPU} to VCPU %{NEW_VCPU} Virtual ID %{VIRTUAL_ID} and PHYSICAL ID %{PHYSICAL_ID} This %{MOVE_REQUIRED:(does not require|requires)} a move between redistributors. Fields:

DEVICE_ID unsigned int

Device ID for interrupt source.

ID unsigned int

Input interrupt ID that was moved.

MOVE_REQUIRED bool

Whether a move between redistributors was required.

NEW_VCPU unsigned int

Collection ID that this interrupt is now associated with.

OLD_VCPU unsigned int

Collection ID that this interrupt used to be associated with.

PHYSICAL_ID unsigned int

DoorBell Interrupt ID that will be specified to the redistributor.

VIRTUAL_ID unsigned int

Interrupt ID that will be specified to the redistributor.

GICv4_ITS_VMOVPSynchronization

DISPLAY ITS with ITSNumber %{ITSNUM} %{SENDINGnRECEIVED:(received|is submitting)} VMOVP(VCPU:%{VCPU}, RD_BASE:%{REDISTRIBUTOR_BASE}) %{SENDINGnRECEIVED:(from|to)} the Router unit for synchronizing it. Fields:

ITSNUM unsigned int

Current value of GITS_CTLR.ITSNumber.

REDISTRIBUTOR_BASE unsigned int

The base address of the new redistributor that the VCPU should be mapped to.

SENDINGnRECEIVED bool

Whether the ITS is sending or has received a VMOVP to/from the router unit.

VCPU unsigned int

The VCPU to be remapped.

GICv4_ITS_VMOVPSynchronizationAck

DISPLAY ITS with ITSNumber %{ITSNUM} is sending a VMOVPSync Ack to the Router unit.
Fields:

ITSNUM unsigned int

Current value of GITS_CTLR.ITSNumber.

GICv4_ITS_VMOVPSynchronizationDone

DISPLAY ITS with ITSNumber %{ITSNUM} completed VMOVPSynchronization. Fields:

ITSNUM unsigned int

Current value of GITS_CTLR.ITSNumber.

GICv4_ITS_VMOVPSynchronizationStarted

DISPLAY ITS with ITSNumber %{ITSNUM} is starting VMOVPSync(VCPU:%{VCPU}, RDBASE:%{RDBase}) synchronization to other interested ITSs in the IRI. Fields:

ITSNUM unsigned int

Current value of GITS_CTLR.ITSNumber.

RDBase unsigned int

The base address of the new redistributor that the VCPU should be mapped to.

VCPU unsigned int

The VCPU to be remapped.

2.112 GICv3ProtocolChecker

This section describes the trace sources.

GICv3ProtocolChecker.protocol_violation

DISPLAY %{MESSAGE}. Fields:

MESSAGE string

Protocol Violation Message.

2.113 GICv3Redistributor

This section describes the trace sources.

ArchMsg.Error.GICv3IgnorePPILegacyPendingNoSource

DISPLAY PPI %{INTERRUPT_ID} is a GICv2 style SGI, marked pending no sources are pending. Fields:

INTERRUPT_ID unsigned int

The interrupt ID of the PPI.

ArchMsg.Error.GICv3_UnexpectedClearAcknowledge

DISPLAY Received a ClearAcknowledge command but not expecting one.

ArchMsg.Error.GICv3_UnknownTypeUpstreamWrite

DISPLAY Upstream write of unknown type received. Acknowledging and ignoring. This is a PROTOCOL ERROR, the architecture allows system behaviour to become UNPREDICATBLE from this point onwards.

ArchMsg.Error.LPIConfigReadError

DISPLAY LPI ID:%{INTERRUPT_ID} config read at %{ADDRESS} did not succeed. Treated as zero. Fields:

ADDRESS unsigned int

Address used for the access.

INTERRUPT_ID unsigned int

The interrupt ID for the LPI.

ArchMsg.Error.LPIDeviceWriteError

DISPLAY LPI related device write for %{ADDRESS} did not succeed. Further behaviour is UNPREDICATBLE. Fields:

ADDRESS unsigned int

Address used for the access.

ArchMsg.Error.LPIPendingAccessError

DISPLAY LPI pending table %{READnWRITE}:{write|read} access at %{ADDRESS} did not succeed. RAZ/WI, further behaviour is UNPREDICATBLE. Fields:

ADDRESS unsigned int

Address used for the access.

READnWRITE bool

Whether this is a read or a write.

ArchMsg.Error.Using_uninitialized_PROPBASE_Address

DISPLAY Use of uninitialized PROPBASE address %{ADDRESS}, which will be used in config read/write. Fields:

ADDRESS unsigned int

Address used for the access.

ArchMsg.Info.GICv3_DroppedInternalPacket

DISPLAY Internal packet dropped as %{INTERFACE} is asleep. Fields:

INTERFACE unsigned int

Index of the interface.

ArchMsg.Info.GICv3_DroppedInternalPacketForALLCores

DISPLAY Internal packet dropped for all cores.

ArchMsg.Info.GICv3_DroppedInternalPacketSDisabled

DISPLAY Internal packet dropped as %{INTERFACE} is disabled. Fields:

INTERFACE unsigned int

Index of the interface.

ArchMsg.Info.GICv3_LPI_CachedProperty_Discard

DISPLAY LPICache Discard ID:%{INTERRUPT_ID} as it is disabled. Fields:

INTERRUPT_ID unsigned int

The interrupt ID for the LPI.

ArchMsg.Info.GICv3_LPI_CachedProperty_Update

DISPLAY LPICache Update ID:%{INTERRUPT_ID} Priority:%{PRIORITY} Enabled. Fields:

INTERRUPT_ID unsigned int

The interrupt ID for the LPI.

PRIORITY unsigned int

The priority for the Cached interrupt (unshifted value).

ArchMsg.Warning.GICv3-PITS.DeactivateIgnored

DISPLAY Interrupt %{INTERRUPT_ID} is being deactivated but the group modifiable in the received Deactivate command do not allow this. Fields:

INTERRUPT_ID unsigned int

The interrupt ID in the EOI message.

ArchMsg.Warning.GICv3-PITS.TrackActive-NotFound

DISPLAY Received an activate message but interrupt ID (%{INTERRUPT_ID}) is not believed to be pending. Fields:

INTERRUPT_ID unsigned int

The interrupt ID in the Activate message.

ArchMsg.Warning.GICv3.GICR_CTLR_EnableLPI_negedge_ignored

DISPLAY Update to GICR_CTLR has EnableLPI as 0, but EnableLPI has been set to 1 in the past, and this bit only goes one way.

ArchMsg.Warning.GICv3.GenerateSGI_A3VnotSupported

DISPLAY GenerateSGI packet received from CPU with A3V set, the Redistributor does not support A3, it will be acknowledged but is ignored.

ArchMsg.Warning.GICv3.GenerateSGI_IRMwithA3VorAff

DISPLAY GenerateSGI packet received from CPU with IRM=1 but A3V or one of the Ax fields is not zero, this packet should not exist but will be handled as if A0=A1=A2=A3=0.

ArchMsg.Warning.GICv3.GenerateSGI_Ignored

DISPLAY GenerateSGI packet received from CPU but ARE_S and ARE_NS are 0 and distributor is configured to ignore such packets.

ArchMsg.Warning.GICv3.PENDBASERUpdateIgnored

DISPLAY Update to PENDBASER when GICR_CTLR.EnableLPI=1 and GICR_WAKER.Quiescent=0 is ignored.

ArchMsg.Warning.GICv3.PROPBASERUpdateIgnored

DISPLAY Update to PROPBASER when GICR_CTLR.EnableLPI=1 and GICR_WAKER.Quiescent=0 is ignored.

ArchMsg.Warning.GICv3.PROPBASERUpdateReadOnly

DISPLAY Update to PROPBASER when register is read only – Has been ignored and probably won't perform as you would expect.

ArchMsg.Warning.GICv3IgnorePPINotImplemented

DISPLAY Change to pending state for PPI %{INTERRUPT_ID} is ignored because the platform indicates that this interrupt isn't implemented. Fields:

INTERRUPT_ID unsigned int

The interrupt ID of the PPI.

ArchMsg.Warning.GICv3_A3NonZeroNotSupported

DISPLAY Distributor received packet from CPU with non-zero A3 field (%{A3:d}) and the distributor does not support this feature. Fields:

A3 unsigned int

Affinity level 3 value.

ArchMsg.Warning.GICv3_CFGSDISABLE_unsupported

DISPLAY CFGSDISABLE signal has been set but doing so has no effect.

ArchMsg.Warning.GICv3_DroppedInternalPacket

DISPLAY Internal packet dropped as %{INTERFACE} is not connected. Fields:

INTERFACE unsigned int

Index of the interface.

ArchMsg.Warning.GICv3_GICR_WAKER.PROCESSOR_SLEEP clear ignored

DISPLAY GICR_WAKER.PROCESSOR_SLEEP clear ignored as SLEEP is true.

ArchMsg.Warning.GICv3_GICR_WAKER.SLEEP set ignored

DISPLAY GICR_WAKER.SLEEP set ignored as PROCESSOR_SLEEP is not true.

ArchMsg.Warning.GICv3_IgnoreAssigningPEToView0

DISPLAY Ignore assigning the PE which is connected to the redistributor with affinity %{AFFINITY} to view '0' from accessor view %{ACCESSOR_VIEW}. Fields:

ACCESSOR_VIEW unsigned int

The view where this operation is running.

AFFINITY string

Affinity of the redistributor.

ArchMsg.Warning.GICv3_IgnoredActivate

DISPLAY Received Activate packet with unusable payload (8196>ID>1023 and 9:4 != 0).

ArchMsg.Warning.GICv3_IgnoredActivateARE

DISPLAY Received Activate packet with unusual payload (8196>ID>1023) and 9:4 == 0 but ARE forbids the packet to be considered.

ArchMsg.Warning.GICv3_IgnoredDeactivate

DISPLAY Received Deactivate packet with unusable payload (8196>ID>1023 and 9:4 != 0).

ArchMsg.Warning.GICv3_IgnoredDeactivateARE

DISPLAY Received Deactivate packet with unusual payload (8196>ID>1023) and 9:4 == 0 but ARE forbids the packet to be considered.

ArchMsg.Warning.GICv3_IgnoredDeactivateSpurious

DISPLAY Received Deactivate packet with a special ID (1020-1023) or an out of range ID.

ArchMsg.Warning.GICv3_LPIDroppedOutOfRange

DISPLAY LPI %{INTERRUPT_ID:d} has been dropped because that ID supplied in %{REG_NAME} was not a valid LPI. Fields:

INTERRUPT_ID unsigned int

The interrupt ID for the LPI.

REG_NAME string

Name of the register being written.

ArchMsg.Warning.GICv3_LPI_CachedProperty_CacheMismatch

DISPLAY Config for LPI ID:%{INTERRUPT_ID} cached as %{CACHED_CFG} but is %{MEMORY_CFG} in memory. Fields:

CACHED_CFG unsigned int

Configuration the cache returned.

INTERRUPT_ID unsigned int

The interrupt ID for the LPI.

MEMORY_CFG unsigned int

Configuration present in memory for the LPI.

ArchMsg.Warning.GICv3_LPISDisabled

DISPLAY %{READnWRITE:(Write to|Read from)} %{REG_NAME} was ignored because GICR_CTLR.EnableLPIs == 0. Fields:

READnWRITE bool

Whether the access was a read.

REG_NAME string

Name of the register being accessed.

ArchMsg.Warning.GICv3_MemoryMapped_AccessInIncorrectAREState

DISPLAY %{REG_NAME} accessed with %{NS:(secure|non-secure)} %{READnWRITE:(write|read)} when GICD_CTLR.ARE_NS = %{ARE_NS} and GICD_CTLR.ARE_S = %{ARE_S}. (May be due to all interrupts in register being in wrong security state for ARE settings.). Fields:

ARE_NS bool

Value of GICD_CTLR.ARE_NS.

ARE_S bool

Value of GICD_CTLR.ARE_S.

NS bool

Access is non-secure.

READnWRITE bool

Access is a read (rather than a write).

REG_NAME string

Name of the register being accessed.

ArchMsg.Warning.GICv3_MemoryMapped_IgnoredAccessDueToViewMismatch

DISPLAY Attempt to access Redistributor with affinity %{AFFINITY} through %{ACCESSOR_NAME} from View %{VIEW_ID} is ignored because this Redistributor is assigned to View %{OWNER_VIEW_ID}. Fields:

ACCESSOR_NAME string

Name of the register being accessed.

AFFINITY string

Affinity of the redistributor.

OWNER_VIEW_ID unsigned int

The view owning the interrupt.

VIEW_ID unsigned int

The view trying to access the register.

ArchMsg.Warning.GICv3_MemoryMapped_IgnoredPendingChange

DISPLAY Access to %{REG_NAME} ignored as specified interrupt ID (%{INTERRUPT_ID}) is %{IS_PENDING:(not|already)} pending. Fields:

INTERRUPT_ID unsigned int

The interrupt ID specified in the register access.

IS_PENDING bool

The current pending state of the interrupt (and what the access attempted to set it to).

REG_NAME string

Name of the register being accessed.

ArchMsg.Warning.GICv3_MemoryMapped_NSAccessToSecureInterruptIgnored

DISPLAY Access to %{REG_NAME} ignored as specified interrupt ID (%{INTERRUPT_ID}) is configured as a secure interrupt. Fields:

INTERRUPT_ID unsigned int

The interrupt ID specified in the register access.

REG_NAME string

Name of the register being accessed.

ArchMsg.Warning.GICv3_MemoryMapped_WriteIgnored

DISPLAY %{REG_FIELD_NAME} write of value %{VALUE} ignored. Fields:

REG_FIELD_NAME string

Name of register and optionally field being written.

VALUE unsigned int

Value attempting to be written.

ArchMsg.Warning.GICv3_ReceivedInvalidCommandFromAbove

DISPLAY Received a GICv3 command type %{TYPE} of length %{ACTUAL_LENGTH} bytes from upstream port; %{EXPECTED_LENGTH} bytes was expected. Fields:

ACTUAL_LENGTH unsigned int

Actual length of the data in bytes including the header.

DATA unsigned int

The data received (including header).

EXPECTED_LENGTH unsigned int

The length in bytes including header that was expected to be received dependent on command type and potentially the data type in the case of a data write.

TYPE unsigned int

The command type.

ArchMsg.Warning.GICv3_ReceivedInvalidCommandFromBelow

DISPLAY Received a GICv3 command type %{TYPE} of length %{ACTUAL_LENGTH} bytes from downstream port %{REDISTRIBUTOR_INDEX:d}; %{EXPECTED_LENGTH} bytes was expected. Fields:

ACTUAL_LENGTH unsigned int

Actual length of the data in bytes including the header.

DATA unsigned int

The data received (including header).

EXPECTED_LENGTH unsigned int

The length in bytes including header that was expected to be received dependent on command type and potentially the data type in the case of a data write.

REDISTRIBUTOR_INDEX unsigned int

Index for the downstream port that this command was received on.

TYPE unsigned int

The command type.

ArchMsg.Warning.GICv3_Redistributor.MemoryMapped_IgnoredRegisterAccessDueToViewMismatch

DISPLAY Attempt to access register %{REGISTER_NAME} from View %{VIEW_ID} is ignored, because this register is accessible only from View %{OWNER_VIEW_ID}. Fields:

OWNER_VIEW_ID unsigned int

The view owning the register.

REGISTER_NAME string

Name of the register being accessed.

VIEW_ID unsigned int

The view trying to access the register.

ArchMsg.Warning.GICv3_Redistributor.MemoryMapped_ReadIgnored

DISPLAY GICv3 attempting to read location %{OFFSET} but ignored and **RAZ**. Fields:

OFFSET unsigned int

Offset of address within the GICv3 module.

ArchMsg.Warning.GICv3_Redistributor.MemoryMapped_ReadReserved

DISPLAY GICv3 reserved location %{OFFSET} attempting to be read. Fields:

OFFSET unsigned int

Offset of address within the GICv3 module.

ArchMsg.Warning.GICv3_Redistributor.MemoryMapped_ReadWriteOnlyReg

DISPLAY GICv3 write-only register %{REG_NAME} attempting to be read. Fields:

OFFSET unsigned int

Offset of address within the GICv3 module.

REG_NAME string

Name of the register being read.

ArchMsg.Warning.GICv3_Redistributor.MemoryMapped_WriteIgnored

DISPLAY GICv3 location %{OFFSET} attempting to be written with value %{VALUE} but ignored. Fields:

OFFSET unsigned int

Offset of address within the GICv3 module.

ArchMsg.Warning.GICv3_Redistributor.MemoryMapped_WriteReadOnlyReg

DISPLAY GICv3 read-only register %{REG_NAME} attempting to be written with value %{VALUE}. Fields:

OFFSET unsigned int

Offset of address within the GICv3 module.

REG_NAME string

Name of the register being written.

VALUE unsigned int

Value written.

ArchMsg.Warning.GICv3_Redistributor.MemoryMapped_WriteReserved

DISPLAY GICv3 reserved location %{OFFSET} attempting to be written with value %{VALUE}.

Fields:

OFFSET unsigned int

Offset of address within the GICv3 module.

VALUE unsigned int

Value written.

ArchMsg.Warning.GICv3_Redistributor_Internal.MemoryMapped_IgnoredRegisterAccessDueToViewMismatch

DISPLAY Attempt to access register %{REGISTER_NAME} from View %{VIEW_ID} is ignored, because this register is accessible only from View %{OWNER_VIEW_ID}. Fields:

OWNER_VIEW_ID unsigned int

The view owning the register.

REGISTER_NAME string

Name of the register being accessed.

VIEW_ID unsigned int

The view trying to access the register.

ArchMsg.Warning.GICv3_Redistributor_Internal.MemoryMapped_ReadIgnored

DISPLAY GICv3 attempting to read location %{OFFSET} but ignored and RAZ. Fields:

OFFSET unsigned int

Offset of address within the GICv3 module.

ArchMsg.Warning.GICv3_Redistributor_Internal.MemoryMapped_ReadReserved

DISPLAY GICv3 reserved location %{OFFSET} attempting to be read. Fields:

OFFSET unsigned int

Offset of address within the GICv3 module.

ArchMsg.Warning.GICv3_Redistributor_Internal.MemoryMapped_ReadWriteOnlyReg

DISPLAY GICv3 write-only register %{REG_NAME} attempting to be read. Fields:

OFFSET unsigned int

Offset of address within the GICv3 module.

REG_NAME string

Name of the register being read.

ArchMsg.Warning.GICv3_Redistributor_Internal.MemoryMapped_WriteIgnored

DISPLAY GICv3 location %{OFFSET} attempting to be written with value %{VALUE} but ignored. Fields:

OFFSET unsigned int

Offset of address within the GICv3 module.

ArchMsg.Warning.GICv3_Redistributor_Internal.MemoryMapped_WriteReadOnlyReg

DISPLAY GICv3 read-only register %{REG_NAME} attempting to be written with value %{VALUE}. Fields:

OFFSET unsigned int

Offset of address within the GICv3 module.

REG_NAME string

Name of the register being written.

VALUE unsigned int

Value written.

ArchMsg.Warning.GICv3_Redistributor_Internal.MemoryMapped_WriteReserved

DISPLAY GICv3 reserved location %{OFFSET} attempting to be written with value %{VALUE}. Fields:

OFFSET unsigned int

Offset of address within the GICv3 module.

VALUE unsigned int

Value written.

ArchMsg.Warning.GICv3_ReleasedNotSent

DISPLAY The redistributor ignored highest pending Interrupt ID %{INTERRUPT_ID} as it is just released by the CPU Interface. This is done to prevent an infinite loop which might be caused due to a mismatch in interrupt group enable settings at CPU Interface and Redistributor. Fields:

INTERRUPT_ID unsigned int

The interrupt ID of the interrupt considered.

ArchMsg.Warning.GICv3_SGIDiscardedDueToInvalidTarget

DISPLAY SGI %{INTERRUPT_ID} targetted at %{A3:d}:%{A2:d}:%{A1:d}.* with target list %{TARGET_LIST:x}, range selector %{RANGE_SELECTOR} has been discarded because the redistributor cannot be reached. Fields:

A1 unsigned int

Affinity level 1.

A2 unsigned int

Affinity level 2.

A3 unsigned int

Affinity level 3.

INTERRUPT_ID unsigned int

The ID of the SGI.

RANGE_SELECTOR unsigned int

Value of the RS field used for generating the SGI if applicable, SBZ otherwise.

TARGET_LIST unsigned int

Bit vector of targets.

ArchMsg.Warning.GICv3_SGIRNotSupported

DISPLAY IRI received Generate SGI packet from CPU with RSV=1 and RS non-zero and the IRI does not support this feature. RS treated as 0.

ArchMsg.Warning.GICv3_SGI_Dropped

DISPLAY SGI %{INTERRUPT_ID:d} (%{INTERRUPT_NS:(|non-)}secure, group %{INTERRUPT_G1:(0|1)}) activated through %{NS:(|non-)}secure write to %{SGT} register is ignored as it is disabled through configuration (enable = %{ENABLED}) or permission is denied (maybe because of NSACR field = %{NSACR}) or it is a GICv2 style SGI (ARE== %{LEGACY:(1|0)}). Fields:

ENABLED bool

Whether the SGI is enable in the destination CPU.

INTERRUPT_G1 bool

Whether the SGI is configured as group 1 in this redistributor.

INTERRUPT_ID unsigned int

The interrupt ID of the SPI.

INTERRUPT_NS bool

Whether the SGI is configured as non-secure in this redistributor.

LEGACY bool

IS the SGI a legacy type interrupt.

NS bool

Original register access was non-secure.

NSACR unsigned int

The value of the NSACR field for the SGI.

SGT enum

The SGT field in the incoming GenerateSGI packet (corresponds to the ICC register that was written).

ArchMsg.Warning.GICv3_SGI_GICD_SGIR_Broadcast_Dropped

DISPLAY SGI %{INTERRUPT_ID:d} (%{INTERRUPT_NS:(|non-)}secure, group %{INTERRUPT_G1:(0|1)}) activated through %{NS:(|non-)}secure write to GICD_SGIR register is ignored as it is disabled through configuration (enable = %{ENABLED}) or permission is denied (maybe because of NSACR field = %{NSACR}) . Fields:

ENABLED bool

Whether the SGI is enable in the destination CPU.

INTERRUPT_G1 bool

Whether the SGI is configured as group 1 in this redistributor.

INTERRUPT_ID unsigned int

The interrupt ID of the SPI.

INTERRUPT_NS bool

Whether the SGI is configured as non-secure in this redistributor.

NS bool

Original register access was non-secure.

NSACR unsigned int

The value of the NSACR field for the SGI.

ArchMsg.Warning.GICv3_SPIDropped

DISPLAY SPI %{INTERRUPT_ID:d} has been dropped because the destination (%{A3:d}. %{A2:d}. %{A1:d}. %{A0:d}) does not exist; The SPI remains pending. Fields:

A0 unsigned int

Affinity level 0 address.

A1 unsigned int

Affinity level 1 address.

A2 unsigned int

Affinity level 2 address.

A3 unsigned int

Affinity level 3 address.

INTERRUPT_ID unsigned int

The interrupt ID for the LPI.

ArchMsg.Warning.GICv3_VLPIIDroppedOutOfRange

DISPLAY VLPI %{INTERRUPT_ID:d} for %{VPT} has been dropped because that ID supplied in %{REG_NAME} is not valid or out of the supported range. Fields:

INTERRUPT_ID unsigned int

The interrupt ID for the LPI.

REG_NAME string

Name of the register being written.

VPT unsigned int

The address of the virtual pending table.

ArchMsg.Warning.GICv3_VLPIsDisabled

DISPLAY %{READnWRITE:(Write to|Read from)} %{REG_NAME} was ignored because GICR_CTLR.EnableLPIs == 0. Fields:

READnWRITE bool

Whether the access was a read.

REG_NAME string

Name of the register being accessed.

ArchMsg.Warning.GICv3_WakerequestIgnored

DISPLAY Wakerequest was asserted but the component is not asleep.

ArchMsg.Warning.GICv3_access_asleep_rd

DISPLAY %{REG_NAME} is treated as **RAZ/WI** and has no effect as RD is asleep because GICR_CTLR.EnableLPIs == 0. Fields:

REG_NAME string

Name of the register being accessed.

ArchMsg.Warning.GICv3_command_received_processorsleep

DISPLAY Command received from CPU when ProcessorSleep and ChildrenAsleep.

ArchMsg.Warning.ReportedRevisionRegisterRead

DISPLAY Reported revision and patch do not match that of the cpu. Fields:

ORIGINAL_VALUE unsigned int

The value the register would have based on the modelled revision.

REGISTER string

The name of the register that has been read.

REPORTED_VALUE unsigned int

The value the register has been overridden to.

GICv3-PITS.NoHPPIsFound

DISPLAY IRI did not find any HPPI to send.

GICv3-PITS.SkipSendingTheInterrupt

DISPLAY Skip sending the interrupt, and the reason is '%{REASON}'. Fields:

REASON string

The reason of skip sending the interrupt.

GICv3-PITS.TrackActive

DISPLAY Interrupt ID %{INTERRUPT_ID} : Priority %{PRIORITY} -> Active. Fields:

INTERRUPT_ID unsigned int

The interrupt ID of the interrupt considered.

PRIORITY unsigned int

The interrupt Priority of the interrupt considered.

GICv3-PITS.TrackPending

DISPLAY Interrupt ID %{INTERRUPT_ID} : Priority %{PRIORITY} -> Pending. Fields:

INTERRUPT_ID unsigned int

The interrupt ID of the interrupt considered.

PRIORITY unsigned int

The interrupt Priority of the interrupt considered.

GICv3-PITS.TrackSent

DISPLAY Interrupt ID %{INTERRUPT_ID} : Priority %{PRIORITY} -> Sent. Fields:

INTERRUPT_ID unsigned int

The interrupt ID of the interrupt considered.

PRIORITY unsigned int

The interrupt Priority of the interrupt considered.

GICv3_CFGSDISABLE

DISPLAY CFGSDISABLE signal has been %{STATE:(clear|set)}ed. Fields:

STATE bool

The new state of the signal.

GICv3_ClearDropped

DISPLAY Clear for SPI %{INTERRUPT_ID:d} has been dropped because the destination (%{A3:d}.%{A2:d}.%{A1:d}.%{A0:d}) does not exist. Fields:

A0 unsigned int

Affinity level 0 address.

A1 unsigned int

Affinity level 1 address.

A2 unsigned int

Affinity level 2 address.

A3 unsigned int

Affinity level 3 address.

INTERRUPT_ID unsigned int

The interrupt ID for the LPI.

GICv3_Handle_Clear_Interrupt

DISPLAY IRI start handling clear %{INTERRUPT_TYPE} %{INTERRUPT_ID}. Fields:

INTERRUPT_ID unsigned int

The interrupt ID of the PPI.

INTERRUPT_TYPE string

interrupt type.

GICv3_InterruptSetPendingButGroupNotEnabled

DISPLAY Interrupt %{INTERRUPT_ID} was set pending but not forwarded to any CPU as the corresponding interrupt group is not enabled (%{NS:(secure|non-secure)} %{GROUP1:(0|1)}). Fields:

GROUP1 bool

Whether the interrupt is group 1.

INTERRUPT_ID unsigned int

The interrupt ID that was set pending.

NS bool

Whether the interrupt is non-secure.

GICv3_InterruptSetReservedGroup

DISPLAY Interrupt %{INTERRUPT_ID} was sent to be pended but has a reserved group (group == 1 && grpmid == 1); interrupt is treated as non-secure group 1. Fields:

INTERRUPT_ID unsigned int

The interrupt ID that was set pending.

GICv3_NoChangeInPendingStatus

DISPLAY Access to %{REG_NAME} ignored as specified interrupt ID (%{INTERRUPT_ID}) is %{IS_PENDING:(not|already)} pending. Fields:

INTERRUPT_ID unsigned int

The interrupt ID specified in the register access.

IS_PENDING bool

The current pending state of the interrupt (and what the access attempted to set it to).

REG_NAME string

Name of the register being accessed.

GICv3_PE_AssignedToView

DISPLAY assign the PE which is connected to the redistributor with affinity %{AFFINITY} to view %{ASSIGNED_VIEW} from view %{ACCESSOR_VIEW}. Fields:

ACCESSOR_VIEW unsigned int

The view where this operation is running.

AFFINITY string

Affinity of the redistributor.

ASSIGNED_VIEW unsigned int

The PE will be assigned to that view.

GICv3_PIDR

DISPLAY SW accessing %{POS} %{REG_PREFIX}_PIDR<%{INDEX}> register for view %{VIEW_ID}. Fields:

INDEX unsigned int

The index of the register.

POS string

high/low.

REG_PREFIX string

The register prefix.

VIEW_ID unsigned int

The view-id.

GICv3_PPISignalChanged

DISPLAY PPI signal for interrupt ID %{INTERRUPT_ID} is now %{STATE:(clear|set)}. Fields:

INTERRUPT_ID unsigned int

The interrupt ID for the PPI.

STATE bool

The new state of the signal.

GICv3_RaiseDoorbellToNonResidentVCPU

DISPLAY Raising a doorbell %{DOORBELL} for a non-resident VCPU while serving virtual interrupt %{VIRTUAL_INTERRUPT_ID}. Fields:

DOORBELL unsigned int

The Doorbell ID.

VIRTUAL_INTERRUPT_ID unsigned int

The virtual interrupt ID.

GICv3_Read_ViewId_Which_Owns_PE

DISPLAY The PE which is connected to the redistributor with affinity %{AFFINITY} is assigned to view %{ASSIGNED_VIEW} from view %{ACCESSOR_VIEW}. Fields:

ACCESSOR_VIEW unsigned int

The view where this operation is running.

AFFINITY string

Affinity of the redistributor.

ASSIGNED_VIEW unsigned int

The PE will be assigned to that view.

GICv3_Redistributor.MemoryMapped_MultipleViews_Access

DISPLAY GICv3 %{OPERATION} memory access on address %{ADDRESS} from view %{VIEW}. Fields:

ADDRESS unsigned int

The address of the register being accessed.

OPERATION string

read/write operation.

VIEW unsigned int

The view being accessed.

GICv3_Redistributor.MemoryMapped_Read

Trace read from a GICv3 memory-mapped register. Fields:

A0 unsigned int

Affinity level 0 (for banked registers).

A1 unsigned int

Affinity level 1 (for banked registers).

A2 unsigned int

Affinity level 2 (for banked registers).

A3 unsigned int

Affinity level 3 (for banked registers).

BANKED bool

Whether the access is to a banked register.

NS bool

Access is non-secure.

OFFSET unsigned int

Offset of address within the GICv3 module.

REG_NAME string

Name of the register being read.

VALUE unsigned int

Value returned.

GICv3_Redistributor.MemoryMapped_Read64

Trace read from a GICv3 memory-mapped register. Fields:

A0 unsigned int

Affinity level 0 (for banked registers).

A1 unsigned int

Affinity level 1 (for banked registers).

A2 unsigned int

Affinity level 2 (for banked registers).

A3 unsigned int

Affinity level 3 (for banked registers).

BANKED bool

Whether the access is to a banked register.

NS bool

Access is non-secure.

OFFSET unsigned int

Offset of address within the GICv3 module.

REG_NAME string

Name of the register being read.

VALUE unsigned int

Value returned.

GICv3_Redistributor.MemoryMapped_Write

Trace write to a GICv3 memory-mapped register. Fields:

A0 unsigned int

Affinity level 0 (for banked registers).

A1 unsigned int

Affinity level 1 (for banked registers).

A2 unsigned int

Affinity level 2 (for banked registers).

A3 unsigned int

Affinity level 3 (for banked registers).

BANKED bool

Whether the access is to a banked register.

NS bool

Access is non-secure.

OFFSET unsigned int

Offset of address within the GICv3 module.

REG_NAME string

Name of the register being written.

UPDATED_VALUE unsigned int

New value read back from the register; for write-only registers this will be equal to VALUE.

VALUE unsigned int

Value written.

GICv3_Redistributor.MemoryMapped_Write64

Trace write to a GICv3 memory-mapped register. Fields:

A0 unsigned int

Affinity level 0 (for banked registers).

A1 unsigned int

Affinity level 1 (for banked registers).

A2 unsigned int

Affinity level 2 (for banked registers).

A3 unsigned int

Affinity level 3 (for banked registers).

BANKED bool

Whether the access is to a banked register.

NS bool

Access is non-secure.

OFFSET unsigned int

Offset of address within the GICv3 module.

REG_NAME string

Name of the register being written.

UPDATED_VALUE unsigned int

New value read back from the register; for write-only registers this will be equal to VALUE.

VALUE unsigned int

Value written.

GICv3_Redistributor.MemoryMapped_WritePort

Trace write to a GICv3 memory-mapped register acting as imp def port. Fields:

FIRST_BYTE_IN_BURST unsigned int

First byte written in the burst.

NS bool

Access is non-secure.

OFFSET unsigned int

Offset of address within the GICv3 module.

REG_NAME string

Name of the register being written.

GICv3_RedistributorPowerManagementByPWRR

DISPLAY Redistributor power management by PWRR is %{PWRR_PM:(disabled|enabled)}. Fields:

PWRR_PM bool

Whether redistributor power is managed by GICR_PWRR (1 enabled, 0 disabled).

GICv3_RedistributorSettingNewPowerState

DISPLAY Redistributor moving power state from %{OLD_POWER_STATE} to %{NEW_POWER_STATE} due to the following reason: %{REASON}. Fields:

NEW_POWER_STATE string

New power state.

OLD_POWER_STATE string

Old power state.

REASON string

Reason for moving to the new power state.

GICv3_RedistributorSettingNewProcessorPowerState

DISPLAY Redistributor moving processor power state from %{OLD_PROCESSOR_POWER_STATE} to %{NEW_PROCESSOR_POWER_STATE} due to the following reason: %{REASON}. Fields:

NEW_PROCESSOR_POWER_STATE string

New processor power state.

OLD_PROCESSOR_POWER_STATE string

Old processor power state.

REASON string

Reason for moving to the new processor power state.

GICv3_Redistributor_Comms_ReceiveAbove

GICv3 internal communications packet that has been received and is traveling towards the CPU. Fields:

COMMAND enum

The command type for this packet.

DATA unsigned int

The data (including header) contained within this packet.

DATA_LENGTH unsigned int

Number of bytes in the data (including header).

GICv3_Redistributor_Comms_ReceiveBelow

GICv3 internal communications packet that has been received and is traveling towards the top-level. Fields:

COMMAND enum

The command type for this packet.

DATA unsigned int

The data (including header) contained within this packet.

DATA_LENGTH unsigned int

Number of bytes in the data (including header).

INDEX unsigned int

Index for the redistributor at the lower level from which this packet is being sent. Not applicable for sends from CPU interface to RDO.

GICv3_Redistributor_Comms_SendAbove

GICv3 internal communications packet that is being sent towards the top-level. Fields:

COMMAND enum

The command type for this packet.

DATA unsigned int

The data (including header) contained within this packet.

DATA_LENGTH unsigned int

Number of bytes in the data (including header).

GICv3_Redistributor_Comms_SendBelow

GICv3 internal communications packet that is being sent towards the CPU. Fields:

COMMAND enum

The command type for this packet.

DATA unsigned int

The data (including header) contained within this packet.

DATA_LENGTH unsigned int

Number of bytes in the data (including header).

INDEX unsigned int

Index for the redistributor at the lower level from which this packet is being sent. Not applicable for sends from CPU interface to RD0.

GICv3_Redistributor_Internal.MemoryMapped_MultipleViews_Access

DISPLAY GICv3 %{OPERATION} memory access on address %{ADDRESS} from view %{VIEW}. Fields:

ADDRESS unsigned int

The address of the register being accessed.

OPERATION string

read/write operation.

VIEW unsigned int

The view being accessed.

GICv3_Redistributor_Internal.MemoryMapped_Read

Trace read from a GICv3 memory-mapped register. Fields:

A0 unsigned int

Affinity level 0 (for banked registers).

A1 unsigned int

Affinity level 1 (for banked registers).

A2 unsigned int

Affinity level 2 (for banked registers).

A3 unsigned int

Affinity level 3 (for banked registers).

BANKED bool

Whether the access is to a banked register.

NS bool

Access is non-secure.

OFFSET unsigned int

Offset of address within the GICv3 module.

REG_NAME string

Name of the register being read.

VALUE unsigned int

Value returned.

GICv3_Redistributor_Internal.MemoryMapped_Read64

Trace read from a GICv3 memory-mapped register. Fields:

A0 unsigned int

Affinity level 0 (for banked registers).

A1 unsigned int

Affinity level 1 (for banked registers).

A2 unsigned int

Affinity level 2 (for banked registers).

A3 unsigned int

Affinity level 3 (for banked registers).

BANKED bool

Whether the access is to a banked register.

NS bool

Access is non-secure.

OFFSET unsigned int

Offset of address within the GICv3 module.

REG_NAME string

Name of the register being read.

VALUE unsigned int

Value returned.

GICv3_Redistributor_Internal.MemoryMapped_Write

Trace write to a GICv3 memory-mapped register. Fields:

A0 unsigned int

Affinity level 0 (for banked registers).

A1 unsigned int

Affinity level 1 (for banked registers).

A2 unsigned int

Affinity level 2 (for banked registers).

A3 unsigned int

Affinity level 3 (for banked registers).

BANKED bool

Whether the access is to a banked register.

NS bool

Access is non-secure.

OFFSET unsigned int

Offset of address within the GICv3 module.

REG_NAME string

Name of the register being written.

UPDATED_VALUE unsigned int

New value read back from the register; for write-only registers this will be equal to VALUE.

VALUE unsigned int

Value written.

GICv3_Redistributor_Internal.MemoryMapped_Write64

Trace write to a GICv3 memory-mapped register. Fields:

A0 unsigned int

Affinity level 0 (for banked registers).

A1 unsigned int

Affinity level 1 (for banked registers).

A2 unsigned int

Affinity level 2 (for banked registers).

A3 unsigned int

Affinity level 3 (for banked registers).

BANKED bool

Whether the access is to a banked register.

NS bool

Access is non-secure.

OFFSET unsigned int

Offset of address within the GICv3 module.

REG_NAME string

Name of the register being written.

UPDATED_VALUE unsigned int

New value read back from the register; for write-only registers this will be equal to VALUE.

VALUE unsigned int

Value written.

GICv3_Redistributor_Internal.MemoryMapped_WritePort

Trace write to a GICv3 memory-mapped register acting as imp def port. Fields:

FIRST_BYTE_IN_BURST unsigned int

First byte written in the burst.

NS bool

Access is non-secure.

OFFSET unsigned int

Offset of address within the GICv3 module.

REG_NAME string

Name of the register being written.

GICv3_RegUpdated32

DISPLAY %{REG_NAME} has been updated to %{VALUE} by a system event. Fields:

REG_NAME string

The name of the register being updated.

VALUE unsigned int

The updated value.

GICv3_RegUpdated64

DISPLAY %{REG_NAME} has been updated to %{VALUE} by a system event. Fields:

REG_NAME string

The name of the register being updated.

VALUE unsigned int

The updated value.

GICv3_Start_Changing_SGI_PendingStatus

DISPLAY IRI start changing SGI %{INTERRUPT_ID} pending status. Fields:

INTERRUPT_ID unsigned int

The interrupt ID of the SGI.

GICv3_TraceWakeRequest

DISPLAY wake_request signal for interface %INTERFACE has been %{SETnCLEAR:(set|cleared)}. Fields:

INTERFACE unsigned int

Index of the interface.

SETnCLEAR bool

Whether the signal is set(true) or cleared (false).

GICv3_VLPICollectedDoorbell

DISPLAY VLPI %{DOORBELL} which will be used. Fields:

DOORBELL unsigned int

Either doorbell or default doorbell.

2.114 GICv3RedistributorInternal

This section describes the trace sources.

ArchMsg.Info.GICv3_DroppedInternalPacket

DISPLAY Internal packet dropped as %{INTERFACE} is asleep. Fields:

INTERFACE unsigned int

Index of the interface.

ArchMsg.Info.GICv3_DroppedInternalPacketForALLCores

DISPLAY Internal packet dropped for all cores.

ArchMsg.Info.GICv3_DroppedInternalPacketSDisabled

DISPLAY Internal packet dropped as %{INTERFACE} is disabled. Fields:

INTERFACE unsigned int

Index of the interface.

ArchMsg.Warning.GICv3_CFGSDISABLE_unsupported

DISPLAY CFGSDISABLE signal has been set but doing so has no effect.

ArchMsg.Warning.GICv3_DroppedInternalPacket

DISPLAY Internal packet dropped as %{INTERFACE} is not connected. Fields:

INTERFACE unsigned int

Index of the interface.

ArchMsg.Warning.GICv3_ReceivedInvalidCommandFromAbove

DISPLAY Received a GICv3 command type %{TYPE} of length %{ACTUAL_LENGTH} bytes from upstream port; %{EXPECTED_LENGTH} bytes was expected. Fields:

ACTUAL_LENGTH unsigned int

Actual length of the data in bytes including the header.

DATA unsigned int

The data received (including header).

EXPECTED_LENGTH unsigned int

The length in bytes including header that was expected to be received dependent on command type and potentially the data type in the case of a data write.

TYPE unsigned int

The command type.

ArchMsg.Warning.GICv3_ReceivedInvalidCommandFromBelow

DISPLAY Received a GICv3 command type %{TYPE} of length %{ACTUAL_LENGTH} bytes from downstream port %{REDISTRIBUTOR_INDEX:d}; %{EXPECTED_LENGTH} bytes was expected. Fields:

ACTUAL_LENGTH unsigned int

Actual length of the data in bytes including the header.

DATA unsigned int

The data received (including header).

EXPECTED_LENGTH unsigned int

The length in bytes including header that was expected to be received dependent on command type and potentially the data type in the case of a data write.

REDISTRIBUTOR_INDEX unsigned int

Index for the downstream port that this command was received on.

TYPE unsigned int

The command type.

ArchMsg.Warning.GICv3_Redistributor.MemoryMapped_IgnoredRegisterAccessDueToViewMismatch

DISPLAY Attempt to access register %{REGISTER_NAME} from View %{VIEW_ID} is ignored, because this register is accessible only from View %{OWNER_VIEW_ID}. Fields:

OWNER_VIEW_ID unsigned int

The view owning the register.

REGISTER_NAME string

Name of the register being accessed.

VIEW_ID unsigned int

The view trying to access the register.

ArchMsg.Warning.GICv3_Redistributor.MemoryMapped_ReadIgnored

DISPLAY GICv3 attempting to read location %{OFFSET} but ignored and **RAZ**. Fields:

OFFSET unsigned int

Offset of address within the GICv3 module.

ArchMsg.Warning.GICv3_Redistributor.MemoryMapped_ReadReserved

DISPLAY GICv3 reserved location %{OFFSET} attempting to be read. Fields:

OFFSET unsigned int

Offset of address within the GICv3 module.

ArchMsg.Warning.GICv3_Redistributor.MemoryMapped_ReadWriteOnlyReg

DISPLAY GICv3 write-only register %{REG_NAME} attempting to be read. Fields:

OFFSET unsigned int

Offset of address within the GICv3 module.

REG_NAME string

Name of the register being read.

ArchMsg.Warning.GICv3_Redistributor.MemoryMapped_WriteIgnored

DISPLAY GICv3 location %{OFFSET} attempting to be written with value %{VALUE} but ignored. Fields:

OFFSET unsigned int

Offset of address within the GICv3 module.

ArchMsg.Warning.GICv3_Redistributor.MemoryMapped_WriteReadOnlyReg

DISPLAY GICv3 read-only register %{REG_NAME} attempting to be written with value %{VALUE}. Fields:

OFFSET unsigned int

Offset of address within the GICv3 module.

REG_NAME string

Name of the register being written.

VALUE unsigned int

Value written.

ArchMsg.Warning.GICv3_Redistributor.MemoryMapped_WriteReserved

DISPLAY GICv3 reserved location %{OFFSET} attempting to be written with value %{VALUE}. Fields:

OFFSET unsigned int

Offset of address within the GICv3 module.

VALUE unsigned int

Value written.

ArchMsg.Warning.GICv3_SPIDropped

DISPLAY SPI %{INTERRUPT_ID:d} has been dropped because the destination (%{A3:d}. %{A2:d}. %{A1:d}. %{A0:d}) does not exist; The SPI remains pending. Fields:

A0 unsigned int

Affinity level 0 address.

A1 unsigned int

Affinity level 1 address.

A2 unsigned int

Affinity level 2 address.

A3 unsigned int

Affinity level 3 address.

INTERRUPT_ID unsigned int

The interrupt ID for the LPI.

ArchMsg.Warning.GICv3_WakerequestIgnored

DISPLAY Wakerequest was asserted but the component is not asleep.

GICv3_CFGSDISABLE

DISPLAY CFGSDISABLE signal has been %{STATE:(clear|set)}ed. Fields:

STATE bool

The new state of the signal.

GICv3_ClearDropped

DISPLAY Clear for SPI %{INTERRUPT_ID:d} has been dropped because the destination (%{A3:d}.%{A2:d}.%{A1:d}.%{A0:d}) does not exist. Fields:

A0 unsigned int

Affinity level 0 address.

A1 unsigned int

Affinity level 1 address.

A2 unsigned int

Affinity level 2 address.

A3 unsigned int

Affinity level 3 address.

INTERRUPT_ID unsigned int

The interrupt ID for the LPI.

GICv3_PIDR

DISPLAY SW accessing %{POS} %{REG_PREFIX}_PIDR<%{INDEX}> register for view %{VIEW_ID}. Fields:

INDEX unsigned int

The index of the register.

POS string

high/low.

REG_PREFIX string

The register prefix.

VIEW_ID unsigned int

The view-id.

GICv3_Redistributor.MemoryMapped_MultipleViews_Access

DISPLAY GICv3 %{OPERATION} memory access on address %{ADDRESS} from view %{VIEW}. Fields:

ADDRESS unsigned int

The address of the register being accessed.

OPERATION string

read/write operation.

VIEW unsigned int

The view being accessed.

GICv3_Redistributor.MemoryMapped_Read

Trace read from a GICv3 memory-mapped register. Fields:

A0 unsigned int

Affinity level 0 (for banked registers).

A1 unsigned int

Affinity level 1 (for banked registers).

A2 unsigned int

Affinity level 2 (for banked registers).

A3 unsigned int

Affinity level 3 (for banked registers).

BANKED bool

Whether the access is to a banked register.

NS bool

Access is non-secure.

OFFSET unsigned int

Offset of address within the GICv3 module.

REG_NAME string

Name of the register being read.

VALUE unsigned int

Value returned.

GICv3_Redistributor.MemoryMapped_Read64

Trace read from a GICv3 memory-mapped register. Fields:

A0 unsigned int

Affinity level 0 (for banked registers).

A1 unsigned int

Affinity level 1 (for banked registers).

A2 unsigned int

Affinity level 2 (for banked registers).

A3 unsigned int

Affinity level 3 (for banked registers).

BANKED bool

Whether the access is to a banked register.

NS bool

Access is non-secure.

OFFSET unsigned int

Offset of address within the GICv3 module.

REG_NAME string

Name of the register being read.

VALUE unsigned int

Value returned.

GICv3_Redistributor.MemoryMapped_Write

Trace write to a GICv3 memory-mapped register. Fields:

A0 unsigned int

Affinity level 0 (for banked registers).

A1 unsigned int

Affinity level 1 (for banked registers).

A2 unsigned int

Affinity level 2 (for banked registers).

A3 unsigned int

Affinity level 3 (for banked registers).

BANKED bool

Whether the access is to a banked register.

NS bool

Access is non-secure.

OFFSET unsigned int

Offset of address within the GICv3 module.

REG_NAME string

Name of the register being written.

UPDATED_VALUE unsigned int

New value read back from the register; for write-only registers this will be equal to VALUE.

VALUE unsigned int

Value written.

GICv3_Redistributor.MemoryMapped_Write64

Trace write to a GICv3 memory-mapped register. Fields:

A0 unsigned int

Affinity level 0 (for banked registers).

A1 unsigned int

Affinity level 1 (for banked registers).

A2 unsigned int

Affinity level 2 (for banked registers).

A3 unsigned int

Affinity level 3 (for banked registers).

BANKED bool

Whether the access is to a banked register.

NS bool

Access is non-secure.

OFFSET unsigned int

Offset of address within the GICv3 module.

REG_NAME string

Name of the register being written.

UPDATED_VALUE unsigned int

New value read back from the register; for write-only registers this will be equal to VALUE.

VALUE unsigned int

Value written.

GICv3_Redistributor.MemoryMapped_WritePort

Trace write to a GICv3 memory-mapped register acting as imp def port. Fields:

FIRST_BYTE_IN_BURST unsigned int

First byte written in the burst.

NS bool

Access is non-secure.

OFFSET unsigned int

Offset of address within the GICv3 module.

REG_NAME string

Name of the register being written.

GICv3_RedistributorPowerManagementByPWRR

DISPLAY Redistributor power management by PWRR is %{PWRR_PM:(disabled|enabled)}. Fields:

PWRR_PM bool

Whether redistributor power is managed by GICR_PWRR (1 enabled, 0 disabled).

GICv3_RedistributorSettingNewPowerState

DISPLAY Redistributor moving power state from %{OLD_POWER_STATE} to %{NEW_POWER_STATE} due to the following reason: %{REASON}. Fields:

NEW_POWER_STATE string

New power state.

OLD_POWER_STATE string

Old power state.

REASON string

Reason for moving to the new power state.

GICv3_RedistributorSettingNewProcessorPowerState

DISPLAY Redistributor moving processor power state from %{OLD_PROCESSOR_POWER_STATE} to %{NEW_PROCESSOR_POWER_STATE} due to the following reason: %{REASON}. Fields:

NEW_PROCESSOR_POWER_STATE string

New processor power state.

OLD_PROCESSOR_POWER_STATE string

Old processor power state.

REASON string

Reason for moving to the new processor power state.

GICv3_Redistributor_Comms_ReceiveAbove

GICv3 internal communications packet that has been received and is traveling towards the CPU. Fields:

COMMAND enum

The command type for this packet.

DATA unsigned int

The data (including header) contained within this packet.

DATA_LENGTH unsigned int

Number of bytes in the data (including header).

GICv3_Redistributor_Comms_ReceiveBelow

GICv3 internal communications packet that has been received and is traveling towards the top-level. Fields:

COMMAND enum

The command type for this packet.

DATA unsigned int

The data (including header) contained within this packet.

DATA_LENGTH unsigned int

Number of bytes in the data (including header).

INDEX unsigned int

Index for the redistributor at the lower level from which this packet is being sent. Not applicable for sends from CPU interface to RD0.

GICv3_Redistributor_Comms_SendAbove

GICv3 internal communications packet that is being sent towards the top-level. Fields:

COMMAND enum

The command type for this packet.

DATA unsigned int

The data (including header) contained within this packet.

DATA_LENGTH unsigned int

Number of bytes in the data (including header).

GICv3_Redistributor_Comms_SendBelow

GICv3 internal communications packet that is being sent towards the CPU. Fields:

COMMAND enum

The command type for this packet.

DATA unsigned int

The data (including header) contained within this packet.

DATA_LENGTH unsigned int

Number of bytes in the data (including header).

INDEX unsigned int

Index for the redistributor at the lower level from which this packet is being sent. Not applicable for sends from CPU interface to RD0.

GICv3_RegUpdated32

DISPLAY %{REG_NAME} has been updated to %{VALUE} by a system event. Fields:

REG_NAME string

The name of the register being updated.

VALUE unsigned int

The updated value.

GICv3_RegUpdated64

DISPLAY %{REG_NAME} has been updated to %{VALUE} by a system event. Fields:

REG_NAME string

The name of the register being updated.

VALUE unsigned int

The updated value.

GICv3_TraceWakeRequest

DISPLAY wake_request signal for interface %INTERFACE has been %{SETnCLEAR:(set|cleared)}. Fields:

INTERFACE unsigned int

Index of the interface.

SETnCLEAR bool

Whether the signal is set(true) or cleared (false).

2.115 GICv4InterruptTranslationService

This section describes the trace sources.

ArchMsg.Error.GICv3_GITS_BASER_NotProvisionedAtEnable

DISPLAY The register GITS_BASER%{BASER_INDEX:d} for type %TYPE should be valid at enable. The behaviour of the IRI is now unpredictable. Fields:

BASER_INDEX unsigned int

Index of the BASER register that should have been provisioned.

BASER_TYPE enum

The type of the BASER register that needs to have been provisioned.

ArchMsg.Warning.GICv3_CBASERUnpredAddress

DISPLAY GICTS_CBASER written with value %VALUE, UNPREDICATABLE as bit 15:12 are not zero, effective value forced with 15:12 as 0. Fields:

VALUE unsigned int

Value attempting to be written.

ArchMsg.Warning.GICv3_CFGSDISABLE_unsupported

DISPLAY CFGSDISABLE signal has been set but doing so has no effect.

ArchMsg.Warning.GICv3_CWRITER_Write

DISPLAY The offset value %OFFSET written to CWRITER is out of range. Fields:

OFFSET unsigned int

Offset written to CWRITE register.

ArchMsg.Warning.GICv3_CommandInvalidEncoding

DISPLAY The data returned for the ITS command from address %ADDRESS is not a valid ITS command and has been ignored. Fields:

ADDRESS unsigned int

Address that the command was read from.

DATA unsigned int

Data for the command.

ArchMsg.Warning.GICv3_CommandReadError

DISPLAY Got a read error attempting to read an ITS command from address %ADDRESS. Fields:

ADDRESS unsigned int

Address that the command was attempted to be read from.

ArchMsg.Warning.GICv3_DISCARDInvalidDevice

DISPLAY DISCARD command requests that interrupt ID %ID from device %DEVICE_ID should be silently discarded, but device is not mapped. Fields:

DEVICE_ID unsigned int

Device ID for interrupt source.

ID unsigned int

Interrupt ID that should be discarded.

ArchMsg.Warning.GICv3_DISCARDInvalidID

DISPLAY DISCARD command requests that interrupt ID %{ID} from device %{DEVICE_ID} should be silently discarded, but ID is not mapped for that device. Fields:

DEVICE_ID unsigned int

Device ID for interrupt source.

ID unsigned int

Interrupt ID that should be discarded.

ITT_ADDRESS unsigned int

The ITT address found in the device table.

ArchMsg.Warning.GICv3_GITS_BASER_AddressAligmentError

DISPLAY GITS_BASER%{BASER_INDEX:d} was configured to use %{PAGE_SIZE_KB}kB pages with base address : %{BASE_ADDRESS} which is not page size aligned. Fields:

BASER_INDEX unsigned int

Index of the BASER register that should have been provisioned.

BASE_ADDRESS unsigned int

The base address for the table.

PAGE_SIZE_KB unsigned int

The size of a page in kB.

ArchMsg.Warning.GICv3_GITS_BASER_CollectionTableLargerThanSupportedCollectionSpan

DISPLAY GITS_BASER%{BASER_INDEX:d} (%{BASER_TYPE}) provisions %(PROVISIONED_PAGE_COUNT) but this ITS can only use %{MAX_NEEDED_PAGE_COUNT} pages to hold %{COLECTION_COUNT_IN_MEM} collection entries of size %{COLLECTION_ENTRY_SIZE}. Fields:

BASER_INDEX unsigned int

Index of the BASER register that should have been provisioned.

BASER_TYPE enum

The type of the BASER register that needs to have been provisioned.

COLECTION_COUNT_IN_MEM unsigned int

The number of collections suportred by this ITS.

COLLECTION_ENTRY_SIZE unsigned int

The size in bytes of a collection entry.

MAX_NEEDED_PAGE_COUNT unsigned int

The number of pages needed to hold all memory based supported collections.

PROVISIONED_PAGE_COUNT unsigned int

The number of provisioned pages.

ArchMsg.Warning.GICv3_GITS_BASER_InsufficientMemory

DISPLAY %{NAME} command was ignored because the GITS_BASER%{BASER_INDEX:d} register (%{BASER_TYPE}) has not been provisioned with sufficient memory to cope with ID %{ID}. Fields:

BASER_INDEX unsigned int

Index of the BASER register that should have been provisioned.

BASER_TYPE enum

The type of the BASER register that needs to have been provisioned.

ID unsigned int

ID being used to store information.

NAME string

Command or register name that caused device to be used.

ArchMsg.Warning.GICv3_GITS_BASER_InvalidIndirectEntry

DISPLAY %{NAME} command was ignored because the GITS_BASER%{BASER_INDEX:d} register (%{BASER_TYPE}) is indirect and the indirect entry is not valid for ID %{ID}. Fields:

BASER_INDEX unsigned int

Index of the BASER register that should have been provisioned.

BASER_TYPE enum

The type of the BASER register that needs to have been provisioned.

ID unsigned int

ID being used to store information.

NAME string

Command or register name that caused device to be used.

ArchMsg.Warning.GICv3_GITS_BASER_NotProvisioned

DISPLAY %{NAME} command/register write was ignored because the GITS_BASER %{BASER_INDEX:d} register has not been provisioned. Fields:

BASER_INDEX unsigned int

Index of the BASER register that should have been provisioned.

BASER_TYPE enum

The type of the BASER register that needs to have been provisioned.

NAME string

Command or register name that caused device to be used.

ArchMsg.Warning.GICv3_GITS_CTLR_UnpredWrite

DISPLAY GITS_CTLR write with value %{VALUE} when Enable=%{ENABLE} and Quiescent=%{QUIESCENT} changes value of ITSNumber (currently %{ITSNUM}) at a time when doing so makes the behaviour of the system **UNPREDICTABLE**. Fields:

ENABLE bool

Value of the current GITS_CTLR.Enable.

ITSNUM unsigned int

Current value of GITS_CTLR.ITSNumber.

QUIESCENT bool

Value of the current GITS_CTLR.Quiescent.

VALUE unsigned int

Value being written to the register, after masking.

ArchMsg.Warning.GICv3_INVALInvalid

DISPLAY INVAL command requests that all configuration for all interrupts in collection %{COLLECTION_ID} should be invalidated but collection is not mapped. Fields:

COLLECTION_ID unsigned int

Collection ID to be invalidated.

ArchMsg.Warning.GICv3_INVInvalidCollection

DISPLAY INV command requests that configuration for interrupt ID %{ID} from device %{DEVICE_ID} should be invalidated, but ID is not mapped to a mapped collection. Fields:

COLLECTION_ID unsigned int

Collection ID that cannot be found.

DEVICE_ID unsigned int

Device ID for interrupt source.

ID unsigned int

Interrupt ID that should be invalidated.

ITT_ADDRESS unsigned int

The ITT address found in the device table.

ArchMsg.Warning.GICv3_INVInvalidDevice

DISPLAY INV command requests that configuration for interrupt ID %{ID} from device %{DEVICE_ID} should be invalidated, but device is not mapped. Fields:

DEVICE_ID unsigned int

Device ID for interrupt source.

ID unsigned int

Interrupt ID that should be inv.

ArchMsg.Warning.GICv3_INVInvalidID

DISPLAY INV command requests that configuration for interrupt ID %{ID} from device %{DEVICE_ID} should be invalidated, but ID is not mapped for that device. Fields:

DEVICE_ID unsigned int

Device ID for interrupt source.

ID unsigned int

Interrupt ID that should be invalidated.

ITT_ADDRESS unsigned int

The ITT address found in the device table.

ArchMsg.Warning.GICv3_ITS.MemoryMapped_IgnoredRegisterAccessDueToViewMismatch

DISPLAY Attempt to access register %{REGISTER_NAME} from View %{VIEW_ID} is ignored, because this register is accessible only from View %{OWNER_VIEW_ID}. Fields:

OWNER_VIEW_ID unsigned int

The view owning the register.

REGISTER_NAME string

Name of the register being accessed.

VIEW_ID unsigned int

The view trying to access the register.

ArchMsg.Warning.GICv3_ITS.MemoryMapped_ReadIgnored

DISPLAY GICv3 attempting to read location %{OFFSET} but ignored and **RAZ**. Fields:

OFFSET unsigned int

Offset of address within the GICv3 module.

ArchMsg.Warning.GICv3_ITS.MemoryMapped_ReadReserved

DISPLAY GICv3 reserved location %{OFFSET} attempting to be read. Fields:

OFFSET unsigned int

Offset of address within the GICv3 module.

ArchMsg.Warning.GICv3_ITS.MemoryMapped_ReadWriteOnlyReg

DISPLAY GICv3 write-only register %{REG_NAME} attempting to be read. Fields:

OFFSET unsigned int

Offset of address within the GICv3 module.

REG_NAME string

Name of the register being read.

ArchMsg.Warning.GICv3_ITS.MemoryMapped_WriteIgnored

DISPLAY GICv3 location %{OFFSET} attempting to be written with value %{VALUE} but ignored. Fields:

OFFSET unsigned int

Offset of address within the GICv3 module.

ArchMsg.Warning.GICv3_ITS.MemoryMapped_WriteReadOnlyReg

DISPLAY GICv3 read-only register %{REG_NAME} attempting to be written with value %{VALUE}. Fields:

OFFSET unsigned int

Offset of address within the GICv3 module.

REG_NAME string

Name of the register being written.

VALUE unsigned int

Value written.

ArchMsg.Warning.GICv3_ITS.MemoryMapped_WriteReserved

DISPLAY GICv3 reserved location %{OFFSET} attempting to be written with value %{VALUE}. Fields:

OFFSET unsigned int

Offset of address within the GICv3 module.

VALUE unsigned int

Value written.

ArchMsg.Warning.GICv3_ITS.TranslateDeviceOutOfRangeIgnored

DISPLAY GITS_TRANSLATER for device %{DEVICE_ID} ignored as only %{DEVICE_ID_BITS} bits are supported. Fields:

DEVICE_ID unsigned int

Device ID to be mapped.

DEVICE_ID_BITS unsigned int

Device ID bits supported.

ArchMsg.Warning.GICv3_ITS.TranslateEventOutOfRangeIgnored

DISPLAY GITS_TRANSLATER for event %{DEVICE_ID}, %{EVENT_ID} ignored only %{EVENT_ID_BITS} bits are supported. Fields:

DEVICE_ID unsigned int

ID of the device triggering the event.

EVENT_ID unsigned int

Event ID to be mapped.

EVENT_ID_BITS unsigned int

Event ID bits supported.

ArchMsg.Warning.GICv3_ITS.TranslateRequestIgnored

DISPLAY GITS_TRANSLATER for device %{DEVICE_ID} and event %{EVENT_ID} was not successful. Syndrome=%{SYNDROME}. Fields:

DEVICE_ID unsigned int

Device ID to be mapped.

EVENT_ID unsigned int

Event ID to be mapped.

SYNDROME enum

Error Syndrome.

ArchMsg.Warning.GICv3_InvalidDeviceMap

DISPLAY %{CMD_NAME} command requesting translation for device ID %{DEVICE_ID} with Interrupt ID %{ID} that has not been previously mapped. Fields:

CMD_NAME string

The name of ITS Command that triggered the translation.

DEVICE_ID unsigned int

Non-mapped device ID specified in the command.

ID unsigned int

Incoming interrupt ID specified in the command.

ArchMsg.Warning.GICv3_InvalidTargetAddress

DISPLAY Command %{COMMAND_NAME} was dropped because it attempted to use an invalid processor number %{PROCESSOR_NUM} in a monolithic GIC distributor configuration. Fields:

COMMAND_NAME string

Name of the command being executed.

PROCESSOR_NUM unsigned int

Processor number specified in the command.

ArchMsg.Warning.GICv3_InvalidVCPU

DISPLAY %{CMD_NAME} command requesting map info for VCPU %{VCPU} that has not been previously mapped using VMAPP. Fields:

CMD_NAME string

The name of ITS Command that requested the map.

VCPU unsigned int

Virtual CPU specified in the command.

ArchMsg.Warning.GICv3_LPIUnmappedCollection

DISPLAY Request to %{SET:(clear|set)} LPI for device %{DEVICE_ID} with ID %{ID} is ignored as the associated collection (%{COLLECTION_ID}) is not mapped. Fields:

COLLECTION_ID unsigned int

Collection ID that the interrupt maps to.

DEVICE_ID unsigned int

Device ID for the interrupt.

ID unsigned int

Incoming interrupt ID specified.

ITT_ADDRESS unsigned int

The ITT address found in the device table.

SET bool

Whether the interrupt is set (rather than cleared).

ArchMsg.Warning.GICv3_LPIUnmappedDevice

DISPLAY Request to %{SET:(clear|set)} LPI for device %{DEVICE_ID} is ignored as the device is not mapped. Fields:

DEVICE_ID unsigned int

Device ID for the interrupt.

ID unsigned int

Incoming interrupt ID specified.

SET bool

Whether the interrupt is set (rather than cleared).

ArchMsg.Warning.GICv3_LPIUnmappedID

DISPLAY Request to %{SET:(clear|set)} LPI for device %{DEVICE_ID} with ID %{ID} is ignored as the ID is not mapped. Fields:

DEVICE_ID unsigned int

Device ID for the interrupt.

ID unsigned int

Incoming interrupt ID specified.

ITT_ADDRESS unsigned int

The ITT address found in the device table.

SET bool

Whether the interrupt is set (rather than cleared).

ArchMsg.Warning.GICv3_MAPCUnmapInvalid

DISPLAY MAPC command request unmapping collection ID %{COLLECTION_ID} that was not previous mapped. Fields:

COLLECTION_ID unsigned int

Collection ID requested to be unmapped.

ArchMsg.Warning.GICv3_MAPCUnmapInvalidTarget

DISPLAY MAPC command request unmapping with non-zero target value.

ArchMsg.Warning.GICv3_MAPDUnmapInvalid

DISPLAY MAPD command request unmapping device ID %{DEVICE_ID} that was not previous mapped. Fields:

DEVICE_ID unsigned int

Device ID attempting to be unmapped.

ArchMsg.Warning.GICv3_MAPDUnmapInvalidSize

DISPLAY MAPD command request unmapping with non-zero Size value.

ArchMsg.Warning.GICv3_MAPIInvalid

DISPLAY MAPI/MAPT command requesting mapping for device ID %{DEVICE_ID} that has not been previously mapped with MAPD (ID: %{ID}, physical ID: %{PHYSICAL_ID}; COLLECTION: %{COLLECTION_ID}). Fields:

COLLECTION_ID unsigned int

Collection ID specified in the map request.

DEVICE_ID unsigned int

Non-mapped device ID specified in the map request.

ID unsigned int

Incoming interrupt ID specified in the map request.

PHYSICAL_ID unsigned int

Outgoing interrupt ID specified in the map request.

ArchMsg.Warning.GICv3_MOVIInvalidCollection

DISPLAY MOVI command requests moving interrupt ID %{ID} with device %{DEVICE_ID} from collection %{OLD_COLLECTION_ID} to %{NEW_COLLECTION_ID} but the %{NEW_INVALID:(old|new)} collection ID is not mapped. Fields:

DEVICE_ID unsigned int

Device ID for interrupt source.

ID unsigned int

Input interrupt ID that should be moved.

ITT_ADDRESS unsigned int

The ITT address found in the device table.

NEW_COLLECTION_ID unsigned int

Collection ID that was requested.

NEW_INVALID bool

It is the new collection ID that is invalid (rather than the old collection ID).

OLD_COLLECTION_ID unsigned int

Collection ID that this interrupt used to be associated with.

ArchMsg.Warning.GICv3_MOVIInvalidDevice

DISPLAY MOVI command requests moving interrupt ID %{ID} with device %{DEVICE_ID} to collection %{NEW_COLLECTION_ID} but the device is not currently mapped. Fields:

DEVICE_ID unsigned int

Device ID for interrupt source.

ID unsigned int

Input interrupt ID that should be moved.

NEW_COLLECTION_ID unsigned int

Collection ID that was requested.

ArchMsg.Warning.GICv3_MOVIInvalidID

DISPLAY MOVI command requests moving interrupt ID %{ID} with device %{DEVICE_ID} to collection %{NEW_COLLECTION_ID} but the ID given is not currently mapped. Fields:

DEVICE_ID unsigned int

Device ID for interrupt source.

ID unsigned int

Input interrupt ID that should be moved.

ITT_ADDRESS unsigned int

The ITT address found in the device table.

NEW_COLLECTION_ID unsigned int

Collection ID that was requested.

ArchMsg.Warning.GICv3_RangeErrorCollection

DISPLAY %{NAME} command was ignored because the collection ID %{COLLECTION_ID} is out of the supported range. Fields:

COLLECTION_ID unsigned int

Collection ID specified.

NAME string

Command or register name.

ArchMsg.Warning.GICv3_RangeErrorDeviceID

DISPLAY %{NAME} command was ignored because the device ID specified (%{DEVICE_ID}) is out of the supported range. Fields:

DEVICE_ID unsigned int

Device ID specified.

NAME string

Command or register name.

ArchMsg.Warning.GICv3_RangeErrorIDBits

DISPLAY %{NAME} command was ignored because the number of ID bits specified (%{ID_BITS}) is out of the supported range. Fields:

ID_BITS unsigned int

Number of ID bits specified.

NAME string

Command or register name.

ArchMsg.Warning.GICv3_RangeErrorIncomingID

DISPLAY %{NAME} command was ignored because the incoming interrupt ID %{ID} is out of the supported range (max %{ID_BITS:d} bits). Fields:

COLLECTION_ID unsigned int

Collection ID specified.

ID_BITS unsigned int

Maximum number of ID bits allowed for this access.

NAME string

Command or register name.

ArchMsg.Warning.GICv3_WriteIgnoredConfiguration

DISPLAY %{REG_NAME} write of value %{VALUE} ignored due to GICv3 configuration. Fields:

REG_NAME string

Name of register being read or written.

VALUE unsigned int

Value attempting to be written.

ArchMsg.Warning.GICv4_InvalidVPTAddress

DISPLAY %{CMD_NAME} command requesting map info for VPTAddress %{VPTAddr} that has not been previously mapped using VMAPP. Fields:

CMD_NAME string

The name of ITS Command that requested the map.

VPTAddr unsigned int

VPTAddress specified in the command.

ArchMsg.Warning.GICv4_VSGIUnmappedVCPU

DISPLAY Request to %{SET:(clear|set)} vSGI with ID %{ID} for Virtual CPU (%{VCPU}) is ignored as the associated Virtual CPU is not mapped in the ITS. Fields:

ID unsigned int

Incoming interrupt ID specified.

SET bool

Whether the interrupt is set (rather than cleared).

VCPU unsigned int

Virtual CPU specified.

GICv3_CFGSDISABLE

DISPLAY CFGSDISABLE signal has been %{STATE:(clear|set)}ed. Fields:

STATE bool

The new state of the signal.

GICv3_CommandComplete

DISPLAY ITS %{COMMAND_TYPE} %{SUCCESS:(FAILED|SUCCEEDED)}. Fields:

COMMAND_TYPE *enum*

Type of the command being executed.

ERROR *enum*

The error type if the command was not successful.

SUCCESS *bool*

Whether the command succeeded.

GICv3_CommandDecode

DISPLAY DECODE %{OFFSET} %{DATA_0}:%{DATA_1}:%{DATA_2}:%{DATA_3}
%{COMMAND_NAME}. Fields:

COMMAND_NAME *string*

Name of the decoded command.

DATA_0 *unsigned int*

First 64-bit word from the command.

DATA_1 *unsigned int*

Second 64-bit word from the command.

DATA_2 *unsigned int*

Third 64-bit word from the command.

DATA_3 *unsigned int*

Forth 64-bit word from the command.

OFFSET *unsigned int*

Offset from GITS_CBASER that the command was read from.

VALID *bool*

Whether the command was valid.

GICv3_CommandStart_CLEAR

DISPLAY ITS CLEAR DEVICE:%{DEVICE} ID:%{ID}. Fields:

DEVICE *unsigned int*

The Device ID specified in the command.

ID *unsigned int*

The incoming ID of interrupts to move to a new collection.

GICv3_CommandStart_DISCARD

DISPLAY ITS DISCARD DEVICE:%{DEVICE} ID:%{ID}. Fields:

DEVICE *unsigned int*

The Device ID specified in the command.

ID *unsigned int*

The incoming ID of interrupts to discard.

GICv3_CommandStart_INT

DISPLAY ITS INT DEVICE:%{DEVICE} ID:%{ID}. Fields:

DEVICE unsigned int

The Device ID specified in the command.

ID unsigned int

The incoming ID of interrupts to move to a new collection.

GICv3_CommandStart_INV

DISPLAY ITS INV DEVICE:%{DEVICE} ID:%{ID}. Fields:

DEVICE unsigned int

The Device ID specified in the command.

ID unsigned int

The incoming ID of interrupts to invalidate caches for.

GICv3_CommandStart_INVALID

DISPLAY ITS INVALID COLLECTION:%{COLLECTION}. Fields:

COLLECTION unsigned int

The collection for which all interrupt caches should be invalidated.

GICv3_CommandStart_MAPC

DISPLAY ITS MAPC VALID:%{VALID:(0|1)} COLLECTION:%{COLLECTION} TARGET:%{TARGET}. Fields:

COLLECTION unsigned int

The ID of the collection being mapped.

TARGET unsigned int

The processor number or redistributor base address for the redistributor to synchronise.

VALID bool

Whether to create a new device mapping (true) or to discard an old one (false).

GICv3_CommandStart_MAPD

DISPLAY ITS MAPD VALID:%{VALID:(0|1)} DEVICE:%{DEVICE} ITT:%{ITT_ADDRESS} SIZE:%{SIZE}. Fields:

DEVICE unsigned int

The Device ID specified in the command.

ITT_ADDRESS unsigned int

The interrupt translation table address specified in the command.

SIZE unsigned int

Number of interrupt ID bits supported by this device, minus one.

VALID bool

Whether to create a new device mapping (true) or to discard an old one (false).

GICv3_CommandStart_MAPI

DISPLAY ITS MAPI DEVICE:%{DEVICE} ID:%{ID} COLLECTION:%{COLLECTION}. Fields:

COLLECTION unsigned int

The ID of the collection to map the interrupts to.

DEVICE unsigned int

The Device ID specified in the command.

ID unsigned int

The incoming ID of interrupts to map (and the same ID that will be passed to software).

GICv3_CommandStart_MAPTI

DISPLAY ITS MAPTI DEVICE:%{DEVICE} ID:%{ID} pID:%{PHYSICAL_ID} COLLECTION:%{COLLECTION}. Fields:

COLLECTION unsigned int

The ID of the collection to map the interrupts to.

DEVICE unsigned int

The Device ID specified in the command.

ID unsigned int

The incoming ID of interrupts to map.

PHYSICAL_ID unsigned int

The ID presented to software for these interrupts.

GICv3_CommandStart_MOVALL

DISPLAY ITS MOVALL SOURCE:%{SOURCE} TARGET:%{TARGET}. Fields:

SOURCE unsigned int

The processor number or redistributor base address for the source redistributor.

TARGET unsigned int

The processor number or redistributor base address for the target redistributor.

GICv3_CommandStart_MOVI

DISPLAY ITS MOVI DEVICE:%{DEVICE} ID:%{ID} COLLECTION:%{COLLECTION}. Fields:

COLLECTION unsigned int

The ID of the new collection to move the interrupt to.

DEVICE unsigned int

The Device ID specified in the command.

ID unsigned int

The incoming ID of interrupts to move to a new collection.

GICv3_CommandStart_SYNC

DISPLAY ITS CLEAR TARGET:%{TARGET}. Fields:

TARGET unsigned int

The processor number or redistributor base address for the redistributor to synchronise.

GICv3_CommandStart_VINVALL

DISPLAY ITS VINALL VCPU:%{VCPU}. Fields:

VCPU unsigned int

The VCPU id for which all interrupt caches should be invalidated.

GICv3_CommandStart_VMAPI

DISPLAY ITS VMAPI DEVICE:%{DEVICE} VCPU:%{VCPU} ID:%{ID} PHYSICAL_ID:%{PHYSICAL_ID}. Fields:

DEVICE unsigned int

The Device ID specified in the command.

ID unsigned int

The incoming ID of interrupt to map.

PHYSICAL_ID unsigned int

The doorbell ID to be presented to guest.

VCPU unsigned int

The VCPU ID specified in the command.

GICv3_CommandStart_VMAPP

DISPLAY ITS VMAPP VALID:%{VALID:(0|1)} VCPU:%{VCPU} TARGET:%{TARGET_ADDRESS} VPT:%{VPT_ADDRESS} SIZE:%{SIZE}. Fields:

SIZE unsigned int

Number of interrupt ID bits supported by this device, minus one.

TARGET_ADDRESS unsigned int

The Redistributor address specified in the command.

VALID bool

Whether to create a new device mapping (true) or to discard an old one (false).

VCPU unsigned int

The VCPU ID specified in the command.

VPT_ADDRESS unsigned int

The Virtual Pending Table address specified in the command.

GICv3_CommandStart_VMAPTI

DISPLAY ITS VMAPTI DEVICE:%{DEVICE} VCPU:%{VCPU} ID:%{ID} VIRTUAL_ID:%{VIRTUAL_ID} PHYSICAL_ID:%{PHYSICAL_ID}. Fields:

DEVICE unsigned int

The Device ID specified in the command.

ID unsigned int

The incoming ID of interrupt to map.

PHYSICAL_ID unsigned int

The doorbell ID to be presented to guest.

VCPU unsigned int

The VCPU ID specified in the command.

VIRTUAL_ID unsigned int

The Virtual ID to be presented to guest.

GICv3_CommandStart_VMOVI

DISPLAY ITS VMOVI DEVICE:%{DEVICE} VCPU:%{VCPU} ID:%{ID} PHYSICAL_ID:%{PHYSICAL_ID}. Fields:

DEVICE unsigned int

The Device ID specified in the command.

ID unsigned int

The incoming ID of interrupt to map.

PHYSICAL_ID unsigned int

The doorbell ID to be presented to guest.

VCPU unsigned int

The VCPU ID specified in the command.

GICv3_CommandStart_VMOVP

DISPLAY ITS VMOVP VCPU:%{VCPU} TARGET:%{TARGET_ADDRESS} SEQUENCE_NUM:%{SEQUENCE_NUM} ITS_LIST:%{ITS_LIST}. Fields:

ITS_LIST unsigned int

The ITS numbers participating in the synchronizatio operation.

SEQUENCE_NUM unsigned int

The identifier of synchronizatio point in the command.

TARGET_ADDRESS unsigned int

The Redistributor address specified in the command.

VCPU unsigned int

The VCPU ID specified in the command.

GICv3_CommandStart_VSYNC

DISPLAY ITS VSYNC VCPU:%{VCPU}. Fields:

VCPU unsigned int

The VCPU id for which commands must be synchronized.

GICv3_DISCARD

DISPLAY DISCARD command requests that interrupt ID %{ID} from device %{DEVICE_ID} should be silently discarded. Fields:

DEVICE_ID unsigned int

Device ID for interrupt source.

ID unsigned int

Interrupt ID that should be discarded.

ITT_ADDRESS unsigned int

The ITT address found in the device table.

GICv3_INV

DISPLAY INV command requests that configuration for interrupt ID %{ID} from device %{DEVICE_ID} should be invalidated. Request sent to redistributor with base address %{REDISTRIBUTOR_BASE} with ID %{PHYSICAL_ID} for collection %{COLLECTION_ID}. Fields:

COLLECTION_ID unsigned int

Collection ID for this interrupt.

DEVICE_ID unsigned int

Device ID for interrupt source.

ID unsigned int

Input interrupt ID that should be invalidated.

ITT_ADDRESS unsigned int

The ITT address found in the device table.

PHYSICAL_ID unsigned int

Interrupt ID that will be specified to the redistributor.

REDISTRIBUTOR_BASE unsigned int

Base address for the redistributor.

GICv3_INVALL

DISPLAY INVALL command requests that all configuration for all interrupts in collection %{COLLECTION_ID} should be invalidated. Request sent to redistributor with base address %{REDISTRIBUTOR_BASE}. Fields:

COLLECTION_ID unsigned int

Collection ID to be invalidated.

REDISTRIBUTOR_BASE unsigned int

Base address for the redistributor.

GICv3_ITS.MemoryMapped_MultipleViews_Access

DISPLAY GICv3 %{OPERATION} memory access on address %{ADDRESS} from view %{VIEW}. Fields:

ADDRESS unsigned int

The address of the register being accessed.

OPERATION string

read/write operation.

VIEW unsigned int

The view being accessed.

GICv3_ITS.MemoryMapped_Read

Trace read from a GICv3 memory-mapped register. Fields:

A0 unsigned int

Affinity level 0 (for banked registers).

A1 unsigned int

Affinity level 1 (for banked registers).

A2 unsigned int

Affinity level 2 (for banked registers).

A3 unsigned int

Affinity level 3 (for banked registers).

BANKED bool

Whether the access is to a banked register.

NS bool

Access is non-secure.

OFFSET unsigned int

Offset of address within the GICv3 module.

REG_NAME string

Name of the register being read.

VALUE unsigned int

Value returned.

GICv3_ITS.MemoryMapped_Read64

Trace read from a GICv3 memory-mapped register. Fields:

A0 unsigned int

Affinity level 0 (for banked registers).

A1 unsigned int

Affinity level 1 (for banked registers).

A2 unsigned int

Affinity level 2 (for banked registers).

A3 unsigned int

Affinity level 3 (for banked registers).

BANKED bool

Whether the access is to a banked register.

NS bool

Access is non-secure.

OFFSET unsigned int

Offset of address within the GICv3 module.

REG_NAME string

Name of the register being read.

VALUE unsigned int

Value returned.

GICv3_ITS.MemoryMapped_Write

Trace write to a GICv3 memory-mapped register. Fields:

A0 unsigned int

Affinity level 0 (for banked registers).

A1 unsigned int

Affinity level 1 (for banked registers).

A2 unsigned int

Affinity level 2 (for banked registers).

A3 unsigned int

Affinity level 3 (for banked registers).

BANKED bool

Whether the access is to a banked register.

NS bool

Access is non-secure.

OFFSET unsigned int

Offset of address within the GICv3 module.

REG_NAME string

Name of the register being written.

UPDATED_VALUE unsigned int

New value read back from the register; for write-only registers this will be equal to VALUE.

VALUE unsigned int

Value written.

GICv3_ITS.MemoryMapped_Write64

Trace write to a GICv3 memory-mapped register. Fields:

A0 unsigned int

Affinity level 0 (for banked registers).

A1 unsigned int

Affinity level 1 (for banked registers).

A2 unsigned int

Affinity level 2 (for banked registers).

A3 unsigned int

Affinity level 3 (for banked registers).

BANKED bool

Whether the access is to a banked register.

NS bool

Access is non-secure.

OFFSET unsigned int

Offset of address within the GICv3 module.

REG_NAME string

Name of the register being written.

UPDATED_VALUE unsigned int

New value read back from the register; for write-only registers this will be equal to VALUE.

VALUE unsigned int

Value written.

GICv3_ITS.MemoryMapped_WritePort

Trace write to a GICv3 memory-mapped register acting as imp def port. Fields:

FIRST_BYTE_IN_BURST unsigned int

First byte written in the burst.

NS bool

Access is non-secure.

OFFSET unsigned int

Offset of address within the GICv3 module.

REG_NAME string

Name of the register being written.

GICv3_ITS_SendRequestToRedistributor

DISPLAY Send ITS request to redistributor with address %{REDISTRIUBTOR_ADDRESS}. Fields:

REDISTRIUBTOR_ADDRESS unsigned int

The address of the target redistributor including the offset.

GICv3_ITS_TranslationStart

DISPLAY Translation for deviceID:%{DEVICE_ID}, eventID:%{EVENT_ID} %{START: (IGNORED|START)}. Fields:

DEVICE_ID unsigned int

Device ID to be mapped.

EVENT_ID unsigned int

Event ID to be mapped.

START bool

Translation Started.

GICv3_ITS_TranslationStatus

DISPLAY Translation for deviceID:%{DEVICE_ID}, eventID:%{EVENT_ID} %{SUCCESS: (IGNORED|SUCCESS)}. Fields:

DEVICE_ID unsigned int

Device ID to be mapped.

EVENT_ID unsigned int

Event ID to be mapped.

SUCCESS bool

Translation Success.

GICv3_ITTAddressTransaction

DISPLAY ITS ITT TRANSACTION ID:%{DEVICE_ID} ADDRESS:%{ITT_ADDRESS}. Fields:

DEVICE_ID unsigned int

The Device ID specified in the command.

ITT_ADDRESS unsigned int

The base address specified in the MAPD command that is being used.

GICv3_ITTAddressTransactionFailed

DISPLAY ITS ITT TRANSACTION FAILED ID:%{DEVICE_ID} ADDRESS:%{ITT_ADDRESS}. Fields:

DEVICE_ID unsigned int

The Device ID specified in the command.

ITT_ADDRESS unsigned int

The base address specified in the MAPD command that is being used.

GICv3_LPISent

DISPLAY LPI %{PHYSICAL_ID} was %{SET:(cleared|set)} in redistributor with base address %{REDISTRIBUTOR_BASE} due to request for device ID %{DEVICE_ID} with ID %{ID} which mapped to collection %{COLLECTION_ID}. Fields:

COLLECTION_ID unsigned int

Collection ID that the interrupt maps to.

DEVICE_ID unsigned int

Device ID for the interrupt.

ID unsigned int

Incoming interrupt ID specified.

ITT_ADDRESS unsigned int

The ITT address found in the device table.

PHYSICAL_ID unsigned int

Outgoing interrupt ID found in the map.

REDISTRIBUTOR_BASE unsigned int

Base address for redistributor (may be an internal address for monolithic implementations).

SET bool

Whether the interrupt is set (rather than cleared).

GICv3_MAPCMap

DISPLAY MAPC command maps collection ID %{COLLECTION_ID} to redistributor at address %{ADDRESS}. Fields:

ADDRESS unsigned int

Redistributor base address.

COLLECTION_ID unsigned int

Collection ID to be mapped.

GICv3_MAPCMapMonolithic

DISPLAY MAPC command maps collection ID %{COLLECTION_ID} to redistributor for processor number %{PROC_NUM:d} (at address %{ADDRESS}). Fields:

ADDRESS unsigned int

Redistributor base address.

COLLECTION_ID unsigned int

Collection ID to be mapped.

PROC_NUM unsigned int

Processor number for redistributor.

GICv3_MAPCUnmap

DISPLAY MAPC command unmaps collection ID %{COLLECTION_ID}. Fields:

COLLECTION_ID unsigned int

Collection ID being unmapped.

GICv3_MAPDMap

DISPLAY MAPD command maps device ID %{DEVICE_ID} to table %{ITT_ADDRESS} supporting %{INTERRUPT_BITS} interrupt bits. Fields:

DEVICE_ID unsigned int

Device ID to be mapped.

INTERRUPT_BITS unsigned int

Number of interrupt ID bits supported by the device.

ITT_ADDRESS unsigned int

Base address for memory to store Interrupt Translation Table.

GICv3_MAPDUnmap

DISPLAY MAPD command request unmapped device ID %{DEVICE_ID}. Fields:

DEVICE_ID unsigned int

Device ID to be unmapped.

GICv3_MAPI

DISPLAY MAPI/MAPTl command requests mapping for interrupt ID %{ID} => %{PHYSICAL_ID} for device ID %{DEVICE_ID} using collection %{COLLECTION_ID}. Fields:

COLLECTION_ID unsigned int

Collection ID specified in the map request.

DEVICE_ID unsigned int

Device ID specified in the map request.

ID unsigned int

Incoming interrupt ID specified in the map request.

ITT_ADDRESS unsigned int

The ITT address found in the device table.

PHYSICAL_ID unsigned int

Outgoing interrupt ID specified in the map request.

GICv3_MOVALL

DISPLAY MOVALL command requests moving all interrupt from redistributor at address %{OLD_REDISTRIBUTOR_BASE} to %{NEW_REDISTRIBUTOR_BASE}. Fields:

NEW_REDISTRIBUTOR_BASE unsigned int

Base address (or processor number for monolithic implementation) for destination redistributor.

OLD_REDISTRIBUTOR_BASE unsigned int

Base address (or processor number for monolithic implementation) for source redistributor.

GICv3_MOVI

DISPLAY MOVI command requests moving interrupt ID %{ID} with device %{DEVICE_ID} (physical ID %{PHYSICAL_ID} should be moved from collection %{OLD_COLLECTION_ID} to %{NEW_COLLECTION_ID}. This %{MOVE_REQUIRED:(does not require|requires)} a move between redistributors. Fields:

DEVICE_ID unsigned int

Device ID for interrupt source.

ID unsigned int

Input interrupt ID that was moved.

ITT_ADDRESS unsigned int

The ITT address found in the device table.

MOVE_REQUIRED bool

Whether a move between redistributors was required.

NEW_COLLECTION_ID unsigned int

Collection ID that this interrupt is now associated with.

OLD_COLLECTION_ID unsigned int

Collection ID that this interrupt used to be associated with.

PHYSICAL_ID unsigned int

Interrupt ID that will be specified to the redistributor.

GICv3_MultiChip_Forwarding_LPI_CMD_ACK_to_Dist

DISPLAY ITS on Chip (ID:%{THIS_CHIP_ID}) is forwarding LPI_CMD_ACK to the local Distributor. Fields:

THIS_CHIP_ID unsigned int

Chip .

GICv3_MultiChip_Forwarding_LPI_CMD_REQ_to_Dist

DISPLAY ITS on Chip (ID:%{THIS_CHIP_ID}) is forwarding LPI_CMD_REQ to the local Distributor, along with the following info: CMD_TYPE:%{CMD_TYPE} INTID:%{LPI_ID} SRC_CHIP:%{SRC_CHIP} SRC_CORE:%{SRC_CORE} DST_CHIP:%{DST_CHIP} DST_CORE:%{DST_CORE}. Fields:

CMD_TYPE unsigned int

The command type.

DST_CHIP unsigned int

Destination chip for the command.

DST_CORE unsigned int

Destination core for the command.

LPI_ID unsigned int

LPI interrupt ID being operated on by the command.

SRC_CHIP unsigned int

Source chip for the command.

SRC_CORE unsigned int

Source core for the command.

THIS_CHIP_ID unsigned int

Chip ID.

GICv3_MultiChip_Forwarding_VPE_CTLR_ACK_to_Dist

DISPLAY ITS on Chip (ID:%{THIS_CHIP_ID}) is forwarding VPE_CTLR_ACK to the local Distributor for sending cross-chip. Fields:

THIS_CHIP_ID unsigned int

Chip ID.

GICv3_MultiChip_Forwarding_VPE_CTLR_REQ_to_Dist

DISPLAY ITS on Chip (ID:%{THIS_CHIP_ID}) is forwarding VPE_CTLR_REQ to the local Distributor for sending cross-chip. The information passed is: SRC_CHIP_ID:%{SRC_CHIP_ID} VPEID:%{VPEID} TARGET_CHIP_ID:%{TARGET_CHIP_ID} TARGET_PE:%{TARGET_PE}. Fields:

SRC_CHIP_ID unsigned int

Source Chip ID.

TARGET_CHIP_ID unsigned int

Target Chip ID.

TARGET_PE unsigned int

Target PE on dest chip.

VPEID unsigned int

Identifier of the virtual PE being moved.

GICv3_MultiChip_SETLPI_BY_MOVI

DISPLAY Chip (ID:%{THIS_CHIP_ID}) received MOVI command to move LPI (ID:%{LPI}) from chip (ID:%{SRC_TGT}) to chip (ID:%{DST_TGT}). Fields:

DST_TGT unsigned int

Destination chip and core for MOVI.

LPI unsigned int

LPI to be moved.

SRC_TGT unsigned int

Source chip and core for MOVI.

THIS_CHIP_ID unsigned int

Chip .

GICv3_MultiChip_VMOVP_Received

DISPLAY Chip (ID:%{THIS_CHIP_ID}) received VMOVP command to move vPE (ID:%{vPEID}) from chip (ID:%{SRC_TGT}) to chip (ID:%{DST_TGT}). Fields:

DST_TGT unsigned int

Destination chip and core for VMOVP.

SRC_TGT unsigned int

Source chip and core for VMOVP.

THIS_CHIP_ID unsigned int

Chip .

vPEID unsigned int

ID of the vPE to be moved.

GICv3_PIDR

DISPLAY SW accessing %{POS} %{REG_PREFIX}_PIDR< %{INDEX}> register for view %{VIEW_ID}. Fields:

INDEX unsigned int

The index of the register.

POS string

high/low.

REG_PREFIX string

The register prefix.

VIEW_ID unsigned int

The view-id.

GICv3_RegUpdated32

DISPLAY %{REG_NAME} has been updated to %{VALUE} by a system event. Fields:

REG_NAME string

The name of the register being updated.

VALUE unsigned int

The updated value.

GICv3_RegUpdated64

DISPLAY %{REG_NAME} has been updated to %{VALUE} by a system event. Fields:

REG_NAME string

The name of the register being updated.

VALUE unsigned int

The updated value.

GICv3_TableWalk_Collection

DISPLAY ITS TABLE COL ID:%{ID} TARGET:%{TARGET}. Fields:

ID unsigned int

The ID of the collection being looked up.

TARGET unsigned int

The target address (or processor number) specified for this collection.

GICv3_TableWalk_CollectionInvalid

DISPLAY ITS TABLE COL ID:%{ID}. Fields:

ID unsigned int

The ID of the collection being looked up.

GICv3_TableWalk_Device

DISPLAY ITS TABLE DEV ID:%{DEVICE_ID} ITT:%{ITT_ADDRESS} BITS:
%{INTERRUPT_BITS:d}. Fields:

DEVICE_ID unsigned int

The device ID being looked up.

INTERRUPT_BITS unsigned int

The number of interrupt bits specified in the device table.

ITT_ADDRESS unsigned int

The ITT address found in the device table.

GICv3_TableWalk_DeviceInvalid

DISPLAY ITS TABLE DEV FAULT ID:%{DEVICE_ID}. Fields:

DEVICE_ID unsigned int

The device ID being looked up.

GICv3_TableWalk_ITT

DISPLAY ITS TABLE ITT ID:%{ID} pID:%{PHYSICAL_ID} COLLECTION:%{COLLECTION_ID}. Fields:

COLLECTION_ID unsigned int

The collection ID used for routing this interrupt to the appropriate CPU.

ID unsigned int

The input identifier used for the translation process.

ITT_ADDRESS unsigned int

The base address specified in the previous MAPD command for this table.

PHYSICAL_ID unsigned int

The physical interrupt ID presented to software.

GICv3_TableWalk_ITTInvalid

DISPLAY ITS TABLE ITT FAULT ID:%{ID}. Fields:

ID unsigned int

The input identifier used for the translation process.

ITT_ADDRESS unsigned int

The base address specified in the MAPD command that is being used.

GICv3_TableWalk_ITT_VLPI

DISPLAY ITS TABLE ITT ID:%{ID} pID:%{PHYSICAL_ID} VCPU:%{VCPU}. Fields:

ID unsigned int

The input identifier used for the translation process.

ITT_ADDRESS unsigned int

The base address specified in the previous MAPD command for this table.

PHYSICAL_ID unsigned int

The physical interrupt ID presented to software.

VCPU unsigned int

The VCPU ID used for routing this interrupt to the appropriate CPU.

GICv3_TableWalk_VCPU

DISPLAY ITS VCPU TABLE :%{VCPU} VPT:%{VPT_ADDRESS} BITS:%{VINTERRUPT_BITS:d}
TARGET_ADDRESS:%{TARGET}. Fields:

TARGET unsigned int

The Redistributor address specified in VCPU table.

VCPU unsigned int

The VCPU being looked up.

VINTERRUPT_BITS unsigned int

The number of virtual interrupt bits specified in the VCPU table.

VPT_ADDRESS unsigned int

The Virtual Pending Table address assigned to VCPU.

GICv3_TableWalk_VCPUInvalid

DISPLAY ITS TABLE VCPU FAULT ID:%{VCPU}. Fields:

VCPU unsigned int

The Virtual CPU ID being looked up.

GICv3_TableWalk_VPTAddrInvalid

DISPLAY ITS TABLE VPTAddr FAULT ID:%{VPTAddr}. Fields:

VPTAddr unsigned int

The Virtual Pending Table address supposedly corresponding to a VCPU, being looked up.

GICv3_VMAPI

DISPLAY VMAPI/VMAPT command requests mapping for interrupt ID %{ID} => PhysicalID
%{PHYSICAL_ID} and VirtualID %{VIRTUAL_ID} for device ID %{DEVICE_ID} using vcpu
%{VCPU}. Fields:

DEVICE_ID unsigned int

Device ID specified in the map request.

ID unsigned int

Incoming interrupt ID specified in the map request.

ITT_ADDRESS unsigned int

The ITT address found in the device table.

PHYSICAL_ID unsigned int

Outgoing Physical interrupt ID specified in the map request.

VCPU unsigned int

Virtual CPU specified in the map request.

VIRTUAL_ID unsigned int

Outgoing Virtual interrupt ID specified in the map request.

GICv3_VMAPP_Map

DISPLAY VMAPP command maps VCPU ID %{VCPU} to Redistributor target %{TARGET_ADDRESS} and VPT Address %{VPT_ADDRESS}. Fields:

TARGET_ADDRESS unsigned int

The Redistributor address assigned to VCPU.

VCPU unsigned int

VCPU ID to be mapped.

VPT_ADDRESS unsigned int

The Virtual Pending Table address assigned to VCPU.

GICv3_VMAPP_Unmap

DISPLAY VMAPP command request unmapped VCPU %{VCPU}. Fields:

VCPU unsigned int

Virtual CPU being unmapped.

GICv3_VMOVI

DISPLAY VMOVI command requests moving virtual interrupt ID %{ID} with device %{DEVICE_ID} from VCPU %{OLD_VCPU} to VCPU %{NEW_VCPU} Virtual ID %{VIRTUAL_ID} and PHYSICAL ID %{PHYSICAL_ID} This %{MOVE_REQUIRED:(does not require|requires)} a move between redistributors. Fields:

DEVICE_ID unsigned int

Device ID for interrupt source.

ID unsigned int

Input interrupt ID that was moved.

MOVE_REQUIRED bool

Whether a move between redistributors was required.

NEW_VCPU unsigned int

Collection ID that this interrupt is now associated with.

OLD_VCPU unsigned int

Collection ID that this interrupt used to be associated with.

PHYSICAL_ID unsigned int

DoorBell Interrupt ID that will be specified to the redistributor.

VIRTUAL_ID unsigned int

Interrupt ID that will be specified to the redistributor.

GICv4_CommandStart_INVDB

DISPLAY ITS INVDB vCPU:%{VCPU}). Fields:

VCPU unsigned int

Virtual CPU specified.

GICv4_CommandStart_VMAPP

DISPLAY ITS VMAPP valid:%{VALID}} VCPU:%{VCPU} Redistributor:
 %{REDISTRIBUTOR_BASE} Virtual Config Table:%{VIRTUAL_CONFIG_ADDRESS}
 Virtual Pending Table:%{VIRTUAL_PENDING_ADDRESS} PTZ:%{PTZ} Default Doorbell:
 %{DEFAULT_DOORBELL} Alloc:%{ALLOC} VPT_SIZE:%{VPT_SIZE}. Fields:

ALLOC bool

Whether VMAPP is for first mapping, or last unmapping among multiple ITSs.

DEFAULT_DOORBELL unsigned int

Default doorbell interrupt ID.

PTZ bool

Whether virtual pending table is zero-initialized.

REDISTRIBUTOR_BASE unsigned int

Base address for redistributor (may be an internal address for monolithic implementations).

VALID bool

Whether to create a new device mapping (true) or to discard an old one (false).

VCPU unsigned int

Virtual CPU specified.

VIRTUAL_CONFIG_ADDRESS unsigned int

Base address of vPE virtual configuration table.

VIRTUAL_PENDING_ADDRESS unsigned int

Base address of vPE virtual pending table.

VPT_SIZE unsigned int

The number of bits of vINTID for the VCPU.

GICv4_ITS_OperationQueued

DISPLAY Operation of type:%{TYPE} directed towards Redistributor:
 %{REDISTRIBUTOR_BASE} is queued at ITS, waiting for a SYNC command before it can be
 released. Fields:

REDISTRIBUTOR_BASE unsigned int

Base address for redistributor (may be an internal address for monolithic implementations).

TYPE enum

Type of operation expected to be routed to Redistributor.

GICv4_ITS_OperationReleased

DISPLAY qQueued operation of type:%{TYPE} directed towards Redistributor:
 %{REDISTRIBUTOR_BASE} is now released at ITS. Fields:

REDISTRIBUTOR_BASE unsigned int

Base address for redistributor (may be an internal address for monolithic implementations).

TYPE enum

Type of operation expected to be routed to Redistributor.

GICv4_ITS_VMOVPSynchronization

DISPLAY ITS with ITSNumber %{ITSNUM} %{SENDINGnRECEIVED:(received|is submitting)} VMOVPP(VCPU:%{VCPU}, RD_BASE:%{REDISTRIBUTOR_BASE}) %{SENDINGnRECEIVED:(from|to)} the Router unit for synchronizing it. Fields:

ITSNUM unsigned int

Current value of GITS_CTLR.ITSNumber.

REDISTRIBUTOR_BASE unsigned int

The base address of the new redistributor that the VCPU should be mapped to.

SENDINGnRECEIVED bool

Whether the ITS is sending or has received a VMOVPP to/from the router unit.

VCPU unsigned int

The VCPU to be remapped.

GICv4_ITS_VMOVPSynchronizationAck

DISPLAY ITS with ITSNumber %{ITSNUM} is sending a VMOVPP sync Ack to the Router unit. Fields:

ITSNUM unsigned int

Current value of GITS_CTLR.ITSNumber.

GICv4_ITS_VMOVPSynchronizationDone

DISPLAY ITS with ITSNumber %{ITSNUM} completed VMOVPP synchronization. Fields:

ITSNUM unsigned int

Current value of GITS_CTLR.ITSNumber.

GICv4_ITS_VMOVPSynchronizationStarted

DISPLAY ITS with ITSNumber %{ITSNUM} is starting VMOVPP(VCPU:%{VCPU}, RDBASE:%{RDBase}) synchronization to other interested ITSs in the IRI. Fields:

ITSNUM unsigned int

Current value of GITS_CTLR.ITSNumber.

RDBase unsigned int

The base address of the new redistributor that the VCPU should be mapped to.

VCPU unsigned int

The VCPU to be remapped.

GICv4_ITS_VSGIConfigSending

DISPLAY ITS with ITSNumber %{ITSNUM} is sending VSGI configuration (VCPU:%{VCPU}, ID:%{INTERRUPT_ID}, ENABLE:%{ENABLE}, GROUP:%{GROUP}, PRIORITY:%{PRIORITY}). Forwarding it to redistributor RD_BASE:%{REDISTRIBUTOR_BASE} through internal router. Fields:

ENABLE bool

Whether to enable or disable this VSGI.

GROUP bool

The group that this VSGI belongs to.

INTERRUPT_ID unsigned int

The interrupt ID for the VSGI.

ITSNUM unsigned int

Value of GITS_CTLR.ITSNumber.

PRIORITY unsigned int

The priority of this VSGI.

REDISTRIBUTOR_BASE unsigned int

The base address of the destination redistributor for the VSGI.

VCPU unsigned int

The VCPU destination for the VSGI.

GICv4 ITS_VSGIConfigSending_NMI

DISPLAY ITS with ITSNumber %{ITSNUM} is sending VSGI configuration (VCPU: %{VCPU}, ID:%{INTERRUPT_ID}, ENABLE:%{ENABLE}, GROUP:%{GROUP}, PRIORITY: %{PRIORITY}, NMI_PROPERTY:%{NMI}). Forwarding it to redistributor RD_BASE: %{REDISTRIBUTOR_BASE} through internal router. Fields:

ENABLE bool

Whether to enable or disable this VSGI.

GROUP bool

The group that this VSGI belongs to.

INTERRUPT_ID unsigned int

The interrupt ID for the VSGI.

ITSNUM unsigned int

Value of GITS_CTLR.ITSNumber.

NMI bool

Whether the VSGI has the Non-maskable Interrupt (NMI) property.

PRIORITY unsigned int

The priority of this VSGI.

REDISTRIBUTOR_BASE unsigned int

The base address of the destination redistributor for the VSGI.

VCPU unsigned int

The VCPU destination for the VSGI.

GICv4_ITS_VSGIInterruptSending

DISPLAY ITS with ITSNumber %{ITSNUM} is sending VSGI %{SET:(CLEAR|SET)} (VCPU: %{VCPU}, ID:%{INTERRUPT_ID}) to redistributor RD_BASE:%{REDISTRIBUTOR_BASE} through internal router. Fields:

INTERRUPT_ID unsigned int

The interrupt ID for the VSGI.

ITSNUM unsigned int

Value of GITS_CTLR.ITSNumber.

REDISTRIBUTOR_BASE unsigned int

The base address of the destination redistributor for the VSGI.

SET bool

Whether VSGI is to be set or cleared.

VCPU unsigned int

The VCPU destination for the VSGI.

GICv4_ITS_VSGIRSettingStatus

DISPLAY Request to %{SET:(clear|set)} vSGI with ID %{ID} for Virtual CPU (%{VCPU}) %{SUCCESS:(IGNORED|ACKNOWLEDGED)}. %{MESSAGE}. Fields:

ID unsigned int

Incoming interrupt ID specified.

MESSAGE string

message.

SUCCESS bool

Setting request success.

VCPU unsigned int

Virtual CPU specified.

GICv4_VMAPI_IndividualDoorbellDisabled

DISPLAY VMAPI specified individual doorbell:%{INDIVIDUAL_DOORBELL} for interrupt:%{ID} targeting VCPU:%{VCPU} when individual doorbell is not permitted by GITS_TYPER.nID.GITS_TYPER.nID is honored, so replaced the individual doorbell with 1023. Fields:

ID unsigned int

Incoming interrupt ID specified.

INDIVIDUAL_DOORBELL unsigned int

Default doorbell interrupt ID.

VCPU unsigned int

Virtual CPU specified.

GICv4_VSGISent

DISPLAY VSGI with ID: %{ID} directed to Virtual CPU (%{VCPU}) was %{SET:(cleared|set)} in redistributor with base address %{REDISTRIBUTOR_BASE} due to %{SET:(ITS command VSGI|write of register GITS_SGIR)}. Fields:

ID unsigned int

Incoming interrupt ID specified.

REDISTRIBUTOR_BASE unsigned int

Base address for redistributor (may be an internal address for monolithic implementations).

SET bool

Whether the interrupt is set (rather than cleared).

VCPU unsigned int

Virtual CPU specified.

2.116 GICv4RedistributorInternal

This section describes the trace sources.

ArchMsg.Info.GICv3_DroppedInternalPacket

DISPLAY Internal packet dropped as %{INTERFACE} is asleep. Fields:

INTERFACE unsigned int

Index of the interface.

ArchMsg.Info.GICv3_DroppedInternalPacketForALLCores

DISPLAY Internal packet dropped for all cores.

ArchMsg.Info.GICv3_DroppedInternalPacketSDisabled

DISPLAY Internal packet dropped as %{INTERFACE} is disabled. Fields:

INTERFACE unsigned int

Index of the interface.

ArchMsg.Warning.GICv3_CFGSDISABLE_unsupported

DISPLAY CFGSDISABLE signal has been set but doing so has no effect.

ArchMsg.Warning.GICv3_DroppedInternalPacket

DISPLAY Internal packet dropped as %{INTERFACE} is not connected. Fields:

INTERFACE unsigned int

Index of the interface.

ArchMsg.Warning.GICv3_ReceivedInvalidCommandFromAbove

DISPLAY Received a GICv3 command type %{TYPE} of length %{ACTUAL_LENGTH} bytes from upstream port; %{EXPECTED_LENGTH} bytes was expected. Fields:

ACTUAL_LENGTH unsigned int

Actual length of the data in bytes including the header.

DATA unsigned int

The data received (including header).

EXPECTED_LENGTH unsigned int

The length in bytes including header that was expected to be received dependent on command type and potentially the data type in the case of a data write.

TYPE unsigned int

The command type.

ArchMsg.Warning.GICv3_ReceivedInvalidCommandFromBelow

DISPLAY Received a GICv3 command type %{TYPE} of length %{ACTUAL_LENGTH} bytes from downstream port %{REDISTRIBUTOR_INDEX:d}; %{EXPECTED_LENGTH} bytes was expected. Fields:

ACTUAL_LENGTH unsigned int

Actual length of the data in bytes including the header.

DATA unsigned int

The data received (including header).

EXPECTED_LENGTH unsigned int

The length in bytes including header that was expected to be received dependent on command type and potentially the data type in the case of a data write.

REDISTRIBUTOR_INDEX unsigned int

Index for the downstream port that this command was received on.

TYPE unsigned int

The command type.

ArchMsg.Warning.GICv3_Redistributor.MemoryMapped_IgnoredRegisterAccessDueToViewMismatch

DISPLAY Attempt to access register %{REGISTER_NAME} from View %{VIEW_ID} is ignored, because this register is accessible only from View %{OWNER_VIEW_ID}. Fields:

OWNER_VIEW_ID unsigned int

The view owning the register.

REGISTER_NAME string

Name of the register being accessed.

VIEW_ID unsigned int

The view trying to access the register.

ArchMsg.Warning.GICv3_Redistributor.MemoryMapped_ReadIgnored

DISPLAY GICv3 attempting to read location %{OFFSET} but ignored and **RAZ**. Fields:

OFFSET unsigned int

Offset of address within the GICv3 module.

ArchMsg.Warning.GICv3_Redistributor.MemoryMapped_ReadReserved

DISPLAY GICv3 reserved location %{OFFSET} attempting to be read. Fields:

OFFSET unsigned int

Offset of address within the GICv3 module.

ArchMsg.Warning.GICv3_Redistributor.MemoryMapped_ReadWriteOnlyReg

DISPLAY GICv3 write-only register %{REG_NAME} attempting to be read. Fields:

OFFSET unsigned int

Offset of address within the GICv3 module.

REG_NAME string

Name of the register being read.

ArchMsg.Warning.GICv3_Redistributor.MemoryMapped_WriteIgnored

DISPLAY GICv3 location %{OFFSET} attempting to be written with value %{VALUE} but ignored. Fields:

OFFSET unsigned int

Offset of address within the GICv3 module.

ArchMsg.Warning.GICv3_Redistributor.MemoryMapped_WriteReadOnlyReg

DISPLAY GICv3 read-only register %{REG_NAME} attempting to be written with value %{VALUE}. Fields:

OFFSET unsigned int

Offset of address within the GICv3 module.

REG_NAME string

Name of the register being written.

VALUE unsigned int

Value written.

ArchMsg.Warning.GICv3_Redistributor.MemoryMapped_WriteReserved

DISPLAY GICv3 reserved location %{OFFSET} attempting to be written with value %{VALUE}. Fields:

OFFSET unsigned int

Offset of address within the GICv3 module.

VALUE unsigned int

Value written.

ArchMsg.Warning.GICv3_SPIDropped

DISPLAY SPI %{INTERRUPT_ID:d} has been dropped because the destination (%{A3:d}. %{A2:d}. %{A1:d}. %{A0:d}) does not exist; The SPI remains pending. Fields:

A0 unsigned int

Affinity level 0 address.

A1 unsigned int

Affinity level 1 address.

A2 unsigned int

Affinity level 2 address.

A3 unsigned int

Affinity level 3 address.

INTERRUPT_ID unsigned int

The interrupt ID for the LPI.

ArchMsg.Warning.GICv3_WakerequestIgnored

DISPLAY Wakerequest was asserted but the component is not asleep.

GICv3_CFGSDISABLE

DISPLAY CFGSDISABLE signal has been %{STATE:(clear|set)}ed. Fields:

STATE bool

The new state of the signal.

GICv3_ClearDropped

DISPLAY Clear for SPI %{INTERRUPT_ID:d} has been dropped because the destination (%{A3:d}.%{A2:d}.%{A1:d}.%{A0:d}) does not exist. Fields:

A0 unsigned int

Affinity level 0 address.

A1 unsigned int

Affinity level 1 address.

A2 unsigned int

Affinity level 2 address.

A3 unsigned int

Affinity level 3 address.

INTERRUPT_ID unsigned int

The interrupt ID for the LPI.

GICv3_PIDR

DISPLAY SW accessing %{POS} %{REG_PREFIX}_PIDR<%{INDEX}> register for view %{VIEW_ID}. Fields:

INDEX unsigned int

The index of the register.

POS string

high/low.

REG_PREFIX string

The register prefix.

VIEW_ID unsigned int

The view-id.

GICv3_Redistributor.MemoryMapped_MultipleViews_Access

DISPLAY GICv3 %{OPERATION} memory access on address %{ADDRESS} from view %{VIEW}. Fields:

ADDRESS unsigned int

The address of the register being accessed.

OPERATION string

read/write operation.

VIEW unsigned int

The view being accessed.

GICv3_Redistributor.MemoryMapped_Read

Trace read from a GICv3 memory-mapped register. Fields:

A0 unsigned int

Affinity level 0 (for banked registers).

A1 unsigned int

Affinity level 1 (for banked registers).

A2 unsigned int

Affinity level 2 (for banked registers).

A3 unsigned int

Affinity level 3 (for banked registers).

BANKED bool

Whether the access is to a banked register.

NS bool

Access is non-secure.

OFFSET unsigned int

Offset of address within the GICv3 module.

REG_NAME string

Name of the register being read.

VALUE unsigned int

Value returned.

GICv3_Redistributor.MemoryMapped_Read64

Trace read from a GICv3 memory-mapped register. Fields:

A0 unsigned int

Affinity level 0 (for banked registers).

A1 unsigned int

Affinity level 1 (for banked registers).

A2 unsigned int

Affinity level 2 (for banked registers).

A3 unsigned int

Affinity level 3 (for banked registers).

BANKED bool

Whether the access is to a banked register.

NS bool

Access is non-secure.

OFFSET unsigned int

Offset of address within the GICv3 module.

REG_NAME string

Name of the register being read.

VALUE unsigned int

Value returned.

GICv3_Redistributor.MemoryMapped_Write

Trace write to a GICv3 memory-mapped register. Fields:

A0 unsigned int

Affinity level 0 (for banked registers).

A1 unsigned int

Affinity level 1 (for banked registers).

A2 unsigned int

Affinity level 2 (for banked registers).

A3 unsigned int

Affinity level 3 (for banked registers).

BANKED bool

Whether the access is to a banked register.

NS bool

Access is non-secure.

OFFSET unsigned int

Offset of address within the GICv3 module.

REG_NAME string

Name of the register being written.

UPDATED_VALUE unsigned int

New value read back from the register; for write-only registers this will be equal to VALUE.

VALUE unsigned int

Value written.

GICv3_Redistributor.MemoryMapped_Write64

Trace write to a GICv3 memory-mapped register. Fields:

A0 unsigned int

Affinity level 0 (for banked registers).

A1 unsigned int

Affinity level 1 (for banked registers).

A2 unsigned int

Affinity level 2 (for banked registers).

A3 unsigned int

Affinity level 3 (for banked registers).

BANKED bool

Whether the access is to a banked register.

NS bool

Access is non-secure.

OFFSET unsigned int

Offset of address within the GICv3 module.

REG_NAME string

Name of the register being written.

UPDATED_VALUE unsigned int

New value read back from the register; for write-only registers this will be equal to VALUE.

VALUE unsigned int

Value written.

GICv3_Redistributor.MemoryMapped_WritePort

Trace write to a GICv3 memory-mapped register acting as imp def port. Fields:

FIRST_BYTE_IN_BURST unsigned int

First byte written in the burst.

NS bool

Access is non-secure.

OFFSET unsigned int

Offset of address within the GICv3 module.

REG_NAME string

Name of the register being written.

GICv3_RedistributorPowerManagementByPWRR

DISPLAY Redistributor power management by PWRR is %{PWRR_PM:(disabled|enabled)}.

Fields:

PWRR_PM bool

Whether redistributor power is managed by GICR_PWRR (1 enabled, 0 disabled).

GICv3_RedistributorSettingNewPowerState

DISPLAY Redistributor moving power state from %{OLD_POWER_STATE} to %{NEW_POWER_STATE} due to the following reason: %{REASON}. Fields:

NEW_POWER_STATE string

New power state.

OLD_POWER_STATE string

Old power state.

REASON string

Reason for moving to the new power state.

GICv3_RedistributorSettingNewProcessorPowerState

DISPLAY Redistributor moving processor power state from %{OLD_PROCESSOR_POWER_STATE} to %{NEW_PROCESSOR_POWER_STATE} due to the following reason: %{REASON}. Fields:

NEW_PROCESSOR_POWER_STATE string

New processor power state.

OLD_PROCESSOR_POWER_STATE string

Old processor power state.

REASON string

Reason for moving to the new processor power state.

GICv3_Redistributor_Comms_ReceiveAbove

GICv3 internal communications packet that has been received and is traveling towards the CPU. Fields:

COMMAND enum

The command type for this packet.

DATA unsigned int

The data (including header) contained within this packet.

DATA_LENGTH unsigned int

Number of bytes in the data (including header).

GICv3_Redistributor_Comms_ReceiveBelow

GICv3 internal communications packet that has been received and is traveling towards the top-level. Fields:

COMMAND enum

The command type for this packet.

DATA unsigned int

The data (including header) contained within this packet.

DATA_LENGTH unsigned int

Number of bytes in the data (including header).

INDEX unsigned int

Index for the redistributor at the lower level from which this packet is being sent. Not applicable for sends from CPU interface to RD0.

GICv3_Redistributor_Comms_SendAbove

GICv3 internal communications packet that is being sent towards the top-level. Fields:

COMMAND enum

The command type for this packet.

DATA unsigned int

The data (including header) contained within this packet.

DATA_LENGTH unsigned int

Number of bytes in the data (including header).

GICv3_Redistributor_Comms_SendBelow

GICv3 internal communications packet that is being sent towards the CPU. Fields:

COMMAND enum

The command type for this packet.

DATA unsigned int

The data (including header) contained within this packet.

DATA_LENGTH unsigned int

Number of bytes in the data (including header).

INDEX unsigned int

Index for the redistributor at the lower level from which this packet is being sent. Not applicable for sends from CPU interface to RD0.

GICv3_RegUpdated32

DISPLAY %{REG_NAME} has been updated to %{VALUE} by a system event. Fields:

REG_NAME string

The name of the register being updated.

VALUE unsigned int

The updated value.

GICv3_RegUpdated64

DISPLAY %{REG_NAME} has been updated to %{VALUE} by a system event. Fields:

REG_NAME string

The name of the register being updated.

VALUE unsigned int

The updated value.

GICv3_TraceWakeRequest

DISPLAY wake_request signal for interface %INTERFACE has been %{SETnCLEAR:(set|cleared)}. Fields:

INTERFACE unsigned int

Index of the interface.

SETnCLEAR bool

Whether the signal is set(true) or cleared (false).

2.117 GICv5

This section describes the trace sources.

ArchMsg.Error.gicv5_unable_to_locate_pe_routing_mode_info_for_domain

DISPLAY Unable to find RoutingMode info for PE with affinity %{AFFINITY_ID} for %{IRSDomain} domain. Fields:

AFFINITY_ID unsigned int

Affinity ID of PE.

IRSDOMAIN enum

IRSDomain.

ArchMsg.Warning.gicv5_access_configuration_of_non_existing_pe

DISPLAY Accessing the configuration of non-existing PE at channel index %{CHANNEL_INDEX} on IRS %{IRSID}. Fields:

CHANNEL_INDEX unsigned int

Channel Index of PE.

IRSID unsigned int

IRS id.

ArchMsg.Warning.gicv5_failed_to_locate_PE_inprogress_interrupt_object

DISPLAY IRS '%{CALLER_IRSID}' failed to locate PE in-progress interrupts object for PE with affinity %{AFFINITY} on %{IRSDomain} domain. Fields:

AFFINITY unsigned int

Remote IRSID.

CALLER_IRSID unsigned int

Local IRSID.

IRSDOMAIN enum

IRSDomain.

ArchMsg.Warning.gicv5_ignore_accessing_unreachable_spi

DISPLAY IRS '%{IRSID}' ignore accessing the unreachable SPI %{SPI_ID} as it is %{REASON}. Fields:

IRSID unsigned int

The IRSID requesting the SPI.

REASON string

Reason to having this SPI unreachable.

SPI_ID unsigned int

SPI ID.

ArchMsg.Warning.gicv5_pe_affinity_not_found

DISPLAY PE affinity not found for channel %{CHANNEL_INDEX}. Fields:

CHANNEL_INDEX unsigned int

Channel Index of PE.

ArchMsg.Warning.gicv5_pe_not_found

DISPLAY PE not found for affinity %{AFFINITY_ID} on IRS %{IRSID}. Fields:

AFFINITY_ID unsigned int

Affinity ID of PE.

IRSID unsigned int

IRS id.

ArchMsg.Warning.gicv5_release_last_outstanding_interrupt_while_no_outstanding_interrupts_exist

DISPLAY IRS trying to release non existing outstanding interrupt.

ArchMsg.Warning.gicv5_skip_release_request_as_no_outstanding_interrupt_exists

DISPLAY Unexpected release to interrupt %{INTERRUPT_ID} which is not outstanding for %{IRSDOMAIN} domain. Fields:

INTERRUPT_ID unsigned int

Interrupt ID.

IRSDOMAIN enum

IRSDomain.

ArchMsg.Warning.gicv5_skip_removing_non_existing_pending_interrupt

DISPLAY Failed to remove interrupt %{INTERRUPT_ID} with this priority %{PID}, as we didn't find it in pending interrupts list for %{IRSDOMAIN} domain. Fields:

INTERRUPT_ID unsigned int

Interrupt ID.

IRSDOMAIN enum

IRSDomain.

PID unsigned int

Priority ID.

ArchMsg.Warning.gicv5_unable_to_access_irs_SPIs

DISPLAY IRS '%{IRSID}' is unable to find SPIs owned by it. Fields:

IRSID unsigned int

The IRSID requesting the SPI.

GICV5_ACKNOWLEDGE_AND_RELEASE_OUTSTANDING_INTERRUPT

DISPLAY IRS do acknowledge and release outstanding interrupt %{INTERRUPT_ID} on PE with affinity %{AFFINITY_ID} for %{IRSDOMAIN} domain. Fields:

AFFINITY_ID unsigned int

Affinity ID of PE.

INTERRUPT_ID unsigned int

Interrupt ID.

IRSDOMAIN enum

IRSDomain.

GICV5_FOUND_REMOTE_IRS_OWNING_PE

DISPLAY IRS '%{CALLER_IRSID}' managed to find the IRS '%{REMOTE_IRSID}' which is connected to PE with affinity %{AFFINITY}. Fields:

AFFINITY unsigned int

PE's affinity.

CALLER_IRSID unsigned int

The local IRS.

REMOTE_IRSID unsigned int

The remote IRSID owning the PE.

GICV5_HPPI_AVAILABLE

DISPLAY IRS found HPPI %{INTERRUPT_TYPE} %{INTERRUPT_ID} to be delivered to the PE with affinity %{AFFINITY} for %{IRSDOMAIN} domain. Fields:

AFFINITY unsigned int

The affinity of the PE.

INTERRUPT_ID unsigned int

Interrupt ID.

INTERRUPT_TYPE enum

Interrupt type.

IRSDOMAIN enum

IRSDomain.

GICV5_IRS_ADD_INTERRUPT_TO_PENDING_LIST

DISPLAY IRS added interrupt %{INTERRUPT_ID} of type %{INTERRUPT_TYPE} with priority %{PRIORITY} to pending list of %{PE_INFO} for %{IRSDOMAIN} domain. Fields:

INTERRUPT_ID unsigned int

Interrupt ID.

INTERRUPT_TYPE enum

Interrupt type.

IRSDOMAIN enum

IRSDomain.

PE_INFO string

PE or VPE info.

PRIORITY unsigned int

Interrupt priority.

GICV5_IRS_REMOVE_INTERRUPT_FROM_PENDING_LIST

DISPLAY IRS removed interrupt %{INTERRUPT_ID} of type %{INTERRUPT_TYPE} with priority %{PRIORITY} from pending list of %{PE_INFO} for %{IRSDOMAIN} domain. Fields:

INTERRUPT_ID unsigned int

Interrupt ID.

INTERRUPT_TYPE enum

Interrupt type.

IRSDOMAIN enum

IRSDomain.

PE_INFO string

PE or VPE info.

PRIORITY unsigned int

Interrupt priority.

GICV5_IRS_SELECTED_HPPI

DISPLAY IRS selected interrupt %{INTERRUPT_ID} of type %{INTERRUPT_TYPE} to be HPPI from PE with affinity %{AFFINITY} for %{IRSDOMAIN} domain. Fields:

AFFINITY unsigned int

The affinity of the PE.

INTERRUPT_ID unsigned int

Interrupt ID.

INTERRUPT_TYPE enum

Interrupt type.

IRSDOMAIN enum

IRSDomain.

GICV5_IRS_SELECTED_VIRTUAL_HPPI

DISPLAY IRS selected virtual interrupt %{INTERRUPT_ID} of type %{INTERRUPT_TYPE} to be HPPI from VPE %{VPE_ID} on VM %{VM_ID} for %{IRSDOMAIN} domain. Fields:

INTERRUPT_ID unsigned int

Interrupt ID.

INTERRUPT_TYPE enum

Interrupt type.

IRSDOMAIN enum

IRSDomain.

VM_ID unsigned int

VM ID.

VPE_ID unsigned int

VPE ID.

GICV5_NO_HPPI_AVAILABLE

DISPLAY IRS didn't find any HPPI available to be delivered to the PE with affinity %{AFFINITY} for %{IRSDOMAIN} domain. Fields:

AFFINITY unsigned int

The affinity of the PE.

IRSDOMAIN enum

IRSDomain.

GICV5_NO_VIRTUAL_HPPI_AVAILABLE

DISPLAY IRS didn't find any virtual HPPI available to be delivered to VPE '%{VPE_ID}' on VM '%{VM_ID}' for %{IRSDOMAIN} domain. Fields:

IRSDOMAIN enum

IRSDomain.

VM_ID unsigned int

VM Id.

VPE_ID unsigned int

VPE Id.

GICV5_PE_INPROGRESS_INTERRUPT_OBJECT_CREATED

DISPLAY IRS '%{CALLER_IRSID}' created PE in-progress interrupts object for PE with affinity %{AFFINITY} on %{IRSDomain} domain. Fields:

AFFINITY unsigned int

Remote IRSID.

CALLER_IRSID unsigned int

Local IRSID.

IRSDOMAIN enum

IRSDomain.

GICV5_SKIP_INTERRUPT_FROM_HPPI_SELECTION

DISPLAY IRS didn't consider %{INTERRUPT_TYPE} interrupt %{INTERRUPT_ID} to be HPPI from PE with affinity %{AFFINITY} for %{IRSDOMAIN} domain because %{REASON}. Fields:

AFFINITY unsigned int

The affinity of the PE.

INTERRUPT_ID unsigned int

Interrupt ID.

INTERRUPT_TYPE enum

Interrupt type.

IRSDOMAIN enum

IRSDomain.

REASON string

The reason of not considering that interrupt.

GICV5_SKIP_VIRTUAL_INTERRUPT_FROM_HPPI_SELECTION

DISPLAY IRS didn't consider virtual %{INTERRUPT_TYPE} interrupt %{INTERRUPT_ID} to be HPPI from VPE %{VPE_ID} on VM %{VM_ID} for %{IRSDOMAIN} domain because %{REASON}. Fields:

INTERRUPT_ID unsigned int

Interrupt ID.

INTERRUPT_TYPE enum

Interrupt type.

IRSDOMAIN enum

IRSDomain.

REASON string

The reason of not considering that interrupt.

VM_ID unsigned int

VM ID.

VPE_ID unsigned int

VPE ID.

GICV5_UNABLE_TO_LOCATE_VPE_INPROGRESS_INTERRUPT_OBJECT

DISPLAY IRS '%{CALLER_IRSID}' failed to locate VPE in-progress interrupts object for vPE with VPE_ID %{VPE_ID} and VM_ID %{VM_ID} on %{IRSDomain} domain. Fields:

CALLER_IRSID unsigned int

Local IRSID.

IRSDOMAIN enum

IRSDomain.

VM_ID unsigned int

VM Id.

VPE_ID unsigned int

VPE Id.

GICV5_VIRTUAL_HPPI_AVAILABLE

DISPLAY IRS found HPPI %{INTERRUPT_TYPE} %{INTERRUPT_ID} to be delivered to VPE %{VPE_ID} on VM %{VM_ID} for %{IRSDOMAIN} domain. Fields:

INTERRUPT_ID unsigned int

Interrupt ID.

INTERRUPT_TYPE enum

Interrupt type.

IRSDOMAIN enum

IRSDomain.

VM_ID unsigned int

VM Id.

VPE_ID unsigned int

VPE Id.

GICV5_VPE_INPROGRESS_INTERRUPT_OBJECT_CREATED

DISPLAY IRS '%{CALLER_IRSID}' created VPE in-progress interrupts object for vPE with VPE_ID %{VPE_ID} and VM_ID %{VM_ID} on %{IRSDomain} domain. Fields:

CALLER_IRSID unsigned int

Local IRSID.

IRSDOMAIN enum

IRSDomain.

VM_ID unsigned int

VM Id.

VPE_ID unsigned int

VPE Id.

2.118 IRS

This section describes the trace sources.

ArchMsg.Error.gicv5_block_delivering_command_to_cpuid

Display IRS blocked delivering the command %{COMMAND} to CPUID due to stream connection restrictions. Fields:

COMMAND string

Command to be delivered.

ArchMsg.Error.gicv5_failed_to_wake_up_PE

DISPLAY GICv5 failed to wake up the PE on channel %{CHANNEL_INDEX}. Fields:

CHANNEL_INDEX unsigned int

Channel index.

ArchMsg.Error.gicv5_ignore_upstream_cmd

DISPLAY IRS ignored the CPUIF upstream command on %{IRSDOMAIN} domain from PE on channel index %{CHANNEL_INDEX} because %{REASON}. Fields:

CHANNEL_INDEX unsigned int

Channel for PE.

IRSDOMAIN enum

IRSDomain.

REASON string

The reason of ignoring CPUIF upstream command.

ArchMsg.Error.gicv5_invalid_or_missing_domain

DISPLAY IRS cmd ignored as command targetting wrong or missing domain %{DOMAIN}. Fields:

DOMAIN string

Interrupt domain.

ArchMsg.Error.gicv5_skip_sending_irs_downstream_command

DISPLAY Skip sending irs downstream command: %{COMMAND} for channel %{CHANNEL_INDEX} due to reason: %{SKIP_REASON} . Fields:

CHANNEL_INDEX unsigned int

Channel for PE.

COMMAND string

Downstream command send by IRS.

SKIP_REASON string

Reason to skip sending command.

ArchMsg.Error.gicv5_stream_connection_command_not_allowed_by_IRS

DISPLAY command %{CMD} not forwarded by IRS at channel index %{CHANNEL_INDEX}. Fields:

CHANNEL_INDEX unsigned int

Channel index.

CMD string

Command executed by IRS.

ArchMsg.Error.gicv5_stream_connection_downstream_command_acknowledged_without_being_sent

DISPLAY Received acknowledgement for command %{CMD} but not sent at channel %{CHANNEL_INDEX}. Fields:

CHANNEL_INDEX unsigned int

Channel index of PE.

CMD string

Command executed by IRS Domain.

ArchMsg.Warning.gicv5_irs_ignore_remote_request

DISPLAY Remote IRS %{REMOTE_IRSID} ignored a remote request because %{REASON}.
Fields:

REASON string

The reason of ignoring the remote request.

REMOTE_IRSID unsigned int

IRS ID.

GICV5_IGNORED_EDGE_TRIGGERED_SPI_CLEAR_EVENT

DISPLAY IRS does not generate a CLEAR event for SPIs when the Trigger mode is edge-triggered and SPI signal is de-asserted, which happened for spi wire %{SPI_WIRE}. Fields:

SPI_WIRE unsigned int

SPI wire number.

GICV5_IGNORED_SPI_WIRE_DURING_RESET

DISPLAY IRS received and ignored a spi event for spi wire %{SPI_WIRE} while in reset state.
Fields:

SPI_WIRE unsigned int

SPI Wire number.

GICV5_IGNORE_REASSIGNING_SPI_DOMAIN

DISPLAY Ignore Updating domain of SPI : %{SPI_ID} as current spi domain is non existent.
Fields:

SPI_ID unsigned int

SPI ID.

GICV5_INPUT_WIRE_STATE_CHANGED

SPI input wire state changed. Fields:

STATE enum

The state it changed to.

WIRE_NUM unsigned int

Number of the input wire.

GICV5_IRS_DOWNSTREAM_COMMAND

DISPLAY Send IRS downstream command: %{COMMAND} for PE %{AFFINITY}. Fields:

AFFINITY unsigned int

PE affinity.

COMMAND string

Downstream command send by IRS.

GICV5_IRS_UPSTREAM_COMMAND_RECEIVED

DISPLAY Send IRS upstream command: %{COMMAND} from PE with affinity %{AFFINITY}.
Fields:

AFFINITY unsigned int

PE affinity.

COMMAND string

Upstream command send by IRS.

GICV5_MEMORY_MAPPING

DISPLAY print the memory mapping for IRS '%{FRAME_NAME}' frame on '%{IRSDOMAIN}' domain Match first %{MF}, Match last %{ML}, Remap first {RF}, Remap last %{RL}. Fields:

FRAME_NAME string

IRS frame name.

IRSDOMAIN enum

IRSDomain.

MF unsigned int

Match first for map range.

ML unsigned int

Match last for map range.

RF unsigned int

ReMap first for map range.

RL unsigned int

ReMap last for map range.

GICV5_RESET_ENDED

Reset ended for IRS.

GICV5_RESET_STARTED

Starting reset for IRS.

GICV5_SET_PE_CONNECTION_ONLINE

DISPLAY Set PE connection status online for channel index %{CHANNEL_INDEX}. Fields:

CHANNEL_INDEX unsigned int

Channel Index of PE.

GICV5_SPIS_RESET_STARTED

Starting resetting SPIs.

GICV5_UPDATING_SPI_TO_NON_EXISTENT_DOMAIN

DISPLAY Updating SPI : %{SPI_ID} to non existent domain. Fields:

SPI_ID unsigned int

SPI ID.

2.119 IRSDomain

This section describes the trace sources.

ArchMsg.Error.gicv5_access_vm_out_of_range

DISPLAY A trial to access VM %{VM_ID} which is out of range, the maximum number of supported VM ID bits are %{SUPPORTED_VM_BITS}. Fields:

SUPPORTED_VM_BITS unsigned int

Supported VM bits.

VM_ID unsigned int

VM_ID ID.

ArchMsg.Error.gicv5_block_delivering_command_to_cpuid

DISPLAY IRS blocked delivering the command %{COMMAND} to CPUID due to stream connection restrictions. Fields:

COMMAND string

Command to be delivered.

ArchMsg.Error.gicv5_ignore_access_irs_xxt_cfgr_bitfields

DISPLAY IRS ignore updating '%{BITFIELD_NAME}' to '%{BITFIELD_SW_VALUE}' because %{REASON}. Fields:

BITFIELD_NAME string

The name of the register bitfield SW trying to write '1' on.

BITFIELD_SW_VALUE unsigned int

The value SW trying to write but got ignored.

REASON string

The reason for ignore updating the bitfield.

ArchMsg.Error.gicv5_invalid_ist_split_value_provided

DISPLAY An invalid IST 'Split' value provided as %{REASON}. Fields:

REASON string

The reason of the invalid 'Split' value.

ArchMsg.Error.gicv5_invalid_iste_size_provided

DISPLAY An invalid ISTE size provided as the minimum allowed ISTE size is %{MIN_ISTE_SIZE}, but now it is %{ISTE_SIZE} which is smaller than the minimal allowed. Fields:

ISTE_SIZE unsigned int

Current ISTE size.

MIN_ISTE_SIZE unsigned int

Min allowed size for ISTE.

ArchMsg.Error.gicv5_irs_received_virtual_command_while_no_resident_vpe

DISPLAY IRS received a virtual command %{COMMAND} while no resident VPE on channel index %{CHANNEL_INDEX}. Fields:

CHANNEL_INDEX unsigned int

Channel index.

COMMAND string

CPUIF command.

ArchMsg.Error.gicv5_iste_not_aligned_to_ist

DISPLAY level %{LEVEL} Physical LPI IST is not aligned properly because ISTSZ=%{ISTSZ}, ID_BITS=%{ID_BITS}, L2SZ=%{L2SZ} and ADDRESS=%{ADDRESS}. Fields:

ADDRESS unsigned int

IST Address.

ID_BITS unsigned int

Physical LPI ID_BITS.

ISTSZ unsigned int

Physical LPI ISTSZ.

L2SZ unsigned int

Physical L2SZ field.

LEVEL unsigned int

IST Level.

ArchMsg.Error.gicv5_unable_to_save_vm_interrupts

DISPLAY Unable to save state of interrupts for VM %{VM_ID}. Fields:

VM_ID unsigned int

VM ID.

ArchMsg.Error.gicv5_unsupported_value_provided_for_ist_field

DISPLAY IRS ignore unsupported value '%{FIELD_SW_VALUE}' provided for '%{FIELD_NAME}' because %{REASON}. Fields:

FIELD_NAME string

The name of the IST field SW trying to write on.

FIELD_SW_VALUE unsigned int

The value SW trying to write but got ignored.

REASON string

The reason for ignore the value written.

ArchMsg.Error.gicv5_viste_not_aligned_to_vist

DISPLAY level %{LEVEL} %{TABLE_TYPE} IST is not aligned properly for VM %{VM_ID} because ISTSZ=%{ISTSZ}, ID_BITS=%{ID_BITS}, L2SZ=%{L2SZ} and ADDRESS=%{ADDRESS}. Fields:

ADDRESS unsigned int

IST Address.

ID_BITS unsigned int

SPI/LPI ID_BITS.

ISTSZ unsigned int

SPI/LPI ISTSZ.

L2SZ unsigned int

L2SZ field.

LEVEL unsigned int

VIST_LEVEL.

TABLE_TYPE enum

table type.

VM_ID unsigned int

VM ID.

ArchMsg.Error.gicv5_vmte_not_aligned_to_vmt

DISPLAY level %{LEVEL} VMT is not aligned properly because VM_ID_BITS=%{ID_BITS} and ADDRESS=%{ADDRESS}. Fields:

ADDRESS unsigned int

VMT Address.

ID_BITS unsigned int

VM ID_BITS.

LEVEL unsigned int

IST Level.

ArchMsg.Error.gicv5_vpete_not_aligned_to_vpe

DISPLAY level 2 VPETE array is not aligned to the size of the VPET array for VM: %{VM_ID} and VPE: %{VPE_ID}. Fields:

VM_ID unsigned int

VM ID.

VPE_ID unsigned int

VPE ID.

ArchMsg.Error.gicv5_wrong_id_bits_provided

DISPLAY IRS received wrong %{ID_BITS_VAR_NAME} value '%{VALUE}' as the highest allowed value is '%{MAX}'. Fields:

ID_BITS_VAR_NAME string

The data-structure holding the id bits.

MAX unsigned int

The reason for ignore updating the bitfield.

VALUE unsigned int

The value SW trying to write but got ignored.

ArchMsg.Error.gicv5_wrong_vpe_id_bits_provided

DISPLAY IRS received wrong VMTE.VPE_ID_Bits value '{VALUE}' as the highest allowed value is '{MAX}'. Fields:

MAX unsigned int

The reason for ignore updating the bitfield.

VALUE unsigned int

The value SW trying to write but got ignored.

ArchMsg.Warning.gicv5_access_IST_entry_without_configuring_base_address_register

DISPLAY Attempt to access IST without configuring '{Register}' for ID '{ID}'. Fields:

ID unsigned int

Interrupt/VM/VPE ID.

Register string

The register holding the base address of the table.

ArchMsg.Warning.gicv5_access_VMT_entry_without_configuring_base_address_register

DISPLAY Attempt to access VMT '{Reason}' without configuring '{Register}' for VM_ID '{VM_ID}'. Fields:

Reason string

Reason to access VMT Table.

Register string

The register holding the base address of the table.

VM_ID unsigned int

VM_ID.

ArchMsg.Warning.gicv5_access_VMT_entry_without_configuring_vpe_table

DISPLAY Attempt to access VPET without configuring VPE table for VM_ID: '{VM_ID}'. Fields:

VM_ID unsigned int

VM_ID.

ArchMsg.Warning.gicv5_access_non_existent_VIST_entry

DISPLAY Unable to access a '{INTERRUPT_TYPE}' ISTE for VM: '{VM_ID}'. Fields:

INTERRUPT_TYPE enum

Interrupt type.

VM_ID unsigned int

VM ID.

ArchMsg.Warning.gicv5_failed_to_get_vm_configuration

DISPLAY IRS failed to get the configuration of VM %{VM_ID}. Fields:

VM_ID unsigned int
VM ID.

ArchMsg.Warning.gicv5_failed_to_get_vpe_configuration

DISPLAY IRS failed to get the configuration of VPE %{VPE_ID} on VM %{VM_ID}. Fields:

VM_ID unsigned int
VM ID.

VPE_ID unsigned int
VPE ID.

ArchMsg.Warning.gicv5_fetch_invalid_vm_entry

DISPLAY IRS did fetch invalid VM entry. Fields:

VM_ID unsigned int
VM_ID ID.

ArchMsg.Warning.gicv5_ignore_DPS_write

DISPLAY IRS ignored writing '%{BITFIELD_NAME}' to 0 because '%{IRS_IDRO_BITFIELD}' is 0. Fields:

BITFIELD_NAME string
Could be IRS_PE_CRO or IRS_VPE_CRO.

IRS_IDRO_BITFIELD string
Could be IRS_IDRO.VIRT_ONE_N or IRS_IDRO.ONE_N.

ArchMsg.Warning.gicv5_ignore_processing_interrupt_event

DISPLAY IRS ignored processing %{INTERRUPT_TYPE} '%{INTERRUPT_ID}' because %{REASON}. Fields:

INTERRUPT_ID unsigned int
LPI ID.

INTERRUPT_TYPE enum
Interrupt type.

REASON string
The reason of ignoring the interrupt event.

ArchMsg.Warning.gicv5_ignore_read_access_due_to_valid_bit

DISPLAY Ignore reading register %{REGISTER_TO_ACCESS} because valid bit is not set on register %{CONTROLLER_REGISTER}. Fields:

CONTROLLER_REGISTER string
Name of the controller register.

REGISTER_TO_ACCESS string

Name of accessed register.

ArchMsg.Warning.gicv5_ignore_reassigning_statically_assigned_spi

DISPLAY Ignore updating the assigned domain for the statically assigned SPI: %**{**INTERRUPT_ID**}**. Fields:

INTERRUPT_ID unsigned int

Interrupt ID.

ArchMsg.Warning.gicv5_ignore_register_access

DISPLAY Access to register %**{**REG**}** has been ignored, and the reason is that %**{**REASON**}**. Fields:

REASON string

The reason of ignoring the access.

REG string

The accessed register.

ArchMsg.Warning.gicv5_ignore_updating_trigger_mode_to_unprogrammable_spi

DISPLAY Ignore updating trigger mode for unprogrammable SPI: %**{**SPI_ID**}**. Fields:

SPI_ID unsigned int

SPI ID.

ArchMsg.Warning.gicv5_ignore_write_access_due_to_valid_bit

DISPLAY Ignore writing value %**{**VALUE**}** to register %**{**REGISTER_TO_ACCESS**}** because valid bit is not set on register %**{**CONTROLLER_REGISTER**}**. Fields:

CONTROLLER_REGISTER string

Name of the controller register.

REGISTER_TO_ACCESS string

Name of accessed register.

VALUE unsigned int

Value of attempted write.

ArchMsg.Warning.gicv5_ignore_writing_to_irs_spi_resampler

DISPLAY Ignore writing to IRS_SPI_RESAMPLER for SPI Id %**{**SPI_ID**}**, Reason Message: %**{**REASON**}**. Fields:

REASON string

Reason to ignore Write.

SPI_ID unsigned int

SPI ID.

ArchMsg.Warning.gicv5_invalid_access_to_spi_register

DISPLAY Invalid access to %{REG_NAME} register for SPI %{SPI_ID} as it is assigned to %{SPI_DOMAIN} domain which is not equal to the accessor domain: %{ACCESSOR_DOMAIN}. Fields:

ACCESSOR_DOMAIN enum

ACCESSOR Domain.

REG_NAME string

Register Name.

SPI_DOMAIN enum

SPI Domain.

SPI_ID unsigned int

SPI ID.

ArchMsg.Warning.gicv5_invalid_interrupt_id_bits_provided

DISPLAY Invalid %{INTERRUPT_TYPE} Bits '%{INTERRUPT_Bits}' provided through IRS_IST_CFGR.LPI_ID_Bits (if LPI) or L2_VMTE.%{INTERRUPT_TYPE}_ID_Bits, as it should be between [%{MIN_INTERRUPT_BITS} - %{MAX_INTERRUPT_BITS}]. Fields:

INTERRUPT_Bits unsigned int

Interrupt ID bits supported.

INTERRUPT_TYPE enum

Interrupt type.

MAX_INTERRUPT_BITS unsigned int

Maximum Interrupt ID bits supported.

MIN_INTERRUPT_BITS unsigned int

Minimum Interrupt ID bits supported.

ArchMsg.Warning.gicv5_invalid_pe_config_selected

DISPLAY Failed to select a PE to access its configuration through IRS_PE_SEL.

ArchMsg.Warning.gicv5_irs_domain_disabled

DISPLAY IRS will not be able to forward HPPI to CPUIF (if any exists), as domain is disabled.

ArchMsg.Warning.gicv5_irs_ignore_cpuif_command

DISPLAY IRS ignored CPUIF command %{COMMAND} because %{REASON}. Fields:

COMMAND string

The ignored upstream command.

REASON string

The reason to ignore it.

ArchMsg.Warning.gicv5_irs_l1_table_entry_is_not_valid

DISPLAY Reading a %{TABLE_TYPE:(IST|VMT|VPET)}E id=%{INTERRUPT_ID} which is not valid. Before SW can read %{TABLE_TYPE:(IST|VMT|VPET)}E its Valid flag must be set to 1. Fields:

INTERRUPT_ID unsigned int

L1 Entry ID.

LEVEL unsigned int

The level of the table entry (1 or 2).

TABLE_TYPE enum

Type of table.

ArchMsg.Warning.gicv5_irs_l1_table_entry_is_valid

DISPLAY Reading a %{TABLE_TYPE:(IST|VMT|VPET)}E id=%{INTERRUPT_ID} which is valid. When accessed as a result of writing to register %{REG_NAME} %{TABLE_TYPE:(IST|VMT|VPET)}E should have Valid bit set to 0. Fields:

INTERRUPT_ID unsigned int

L1 Entry ID.

LEVEL unsigned int

The level of the table entry (1 or 2).

REG_NAME string

Name of the register being written.

TABLE_TYPE enum

Type of table.

ArchMsg.Warning.gicv5_irs_override_routing_mode

DISPLAY IRS override the routing mode provided to be TARGET_ONLY.

ArchMsg.Warning.gicv5_irs_table_entry_invalid_id

DISPLAY Reading interrupt %{INTERRUPT_ID} which is out of range. ID should be smaller than %{MAXID}. Fields:

INTERRUPT_ID unsigned int

Interrupt ID.

MAXID unsigned int

Maximum number of interrupts.

ArchMsg.Warning.gicv5_irs_table_read_failed

DISPLAY IRS could not read a %{TABLE_TYPE:(IST|VMT|VPET)}E at %{ADDRESS}. Fields:

ADDRESS unsigned int

ADDRESS of the table entry.

LEVEL unsigned int

The level of the table entry (1 or 2).

TABLE_TYPE enum

Type of table.

ArchMsg.Warning.gicv5_irs_table_write_failed

DISPLAY IRS could not a %{TABLE_TYPE:(IST|VMT|VPET)}E to %{ADDRESS}. Fields:

ADDRESS unsigned int

ADDRESS of the table entry.

LEVEL unsigned int

The level of the table entry (1 or 2).

TABLE_TYPE enum

Type of table.

ArchMsg.Warning.gicv5_irs_trimming_affinity_provided_by_cpuid

DISPLAY IRS trimming unsupported bits in %{FIELD} '%{AFFINITY}' provided by CPUID as it is exceeding the %{FIELD} supported bits which is '%{AFFINITY_BITS_SUPPORTED}'. Fields:

AFFINITY unsigned int

Affinity ID.

AFFINITY_BITS_SUPPORTED unsigned int

Affinity bits supported by IRS.

FIELD string

VPE_ID or AFFINITY.

ArchMsg.Warning.gicv5_memory_mapped_access_reserved

DISPLAY %{IS_WRITE:(Read|Write)} access to reserved offset %{OFFSET} within a register frame. Fields:

IS_WRITE bool

Whether this is a write access.

OFFSET unsigned int

Offset of address within the GICv5 register frame.

ArchMsg.Warning.gicv5_memory_mapped_access_wider_than_size

DISPLAY 64-bit %{IS_WRITE:(read|write)} access affects on one or more 32-bit registers at offset %{OFFSET}. Fields:

IS_WRITE bool

Whether this is a write access.

OFFSET unsigned int

Offset of address within the GICv5 register frame.

ArchMsg.Warning.gicv5_memory_mapped_illegal_access_width

DISPLAY %{IS_WRITE:(Read|Write)} access to memory-mapped interface using an illegal access width of %{WIDTH} bits. Fields:

IS_WRITE bool

Whether this is a write access.

WIDTH unsigned int

Access width in bits.

ArchMsg.Warning.gicv5_memory_mapped_read_write_only_reg

DISPLAY Attempt to read from write-only register %{REG_NAME}. Fields:

REG_NAME string

Name of the register being read.

ArchMsg.Warning.gicv5_memory_mapped_write64_read_only_reg

DISPLAY Attempt to write value %{VALUE} to read-only register %{REG_NAME}. Fields:

REG_NAME string

Name of the register being written.

VALUE unsigned int

Value written.

ArchMsg.Warning.gicv5_memory_mapped_write_read_only_reg

DISPLAY Attempt to write value %{VALUE} to read-only register %{REG_NAME}. Fields:

REG_NAME string

Name of the register being written.

VALUE unsigned int

Value written.

ArchMsg.Warning.gicv5_no_spis_implemented

DISPLAY No SPIs are implemented, IRS_IDR4.SPI_RANGE is 0.

ArchMsg.Warning.gicv5_override_ist_split_value

DISPLAY IRS is overriding IST 'Split' value to be %{IST_SPLIT}. Fields:

IST_SPLIT unsigned int

Current IST 'Split' value.

ArchMsg.Warning.gicv5_override_iste_size

DISPLAY IRS is overriding ISTE size to be %{ISTE_SIZE}. Fields:

ISTE_SIZE unsigned int

Current ISTE size.

ArchMsg.Warning.gicv5_pas_filtering

DISPLAY Ignore %{IS_WRITE:(Read|Write)} operation performed by software in %{TX_PAS} PAS to a memory mapped register which expect access from %{EX_PAS} PAS at address %{ADDR}. Fields:

ADDR unsigned int

Transaction Address.

EX_PAS string

Expected PAS.

IS_WRITE bool

Whether this is a write access.

TX_PAS string

PAS associated with Read or Write Transaction.

ArchMsg.Warning.gicv5_select_out_of_range_spi

DISPLAY Select SPI %{SPI} which is out of range, as SPI Base is %{SPI_BASE} and range is %{SPI_RANGE}. Fields:

SPI unsigned int

SPI Number.

SPI_BASE unsigned int

Min SPI IntID.

SPI_RANGE unsigned int

SPI Range.

ArchMsg.Warning.gicv5_skip_activating_already_active_interrupt

DISPLAY IRS skip activating interrupt %{INTERRUPT_ID} as it is already 'Active'. Fields:

INTERRUPT_ID unsigned int

Interrupt ID.

ArchMsg.Warning.gicv5_skip_deactivating_inactive_interrupt

DISPLAY IRS skip deactivating the inactive interrupt %{INTERRUPT_ID}. Fields:

INTERRUPT_ID unsigned int

Interrupt ID.

ArchMsg.Warning.gicv5_skip_moving_pending_hppi_1ofN_interrupt_on_new_target

DISPLAY skip removing pending, hppi and 1ofN interrupt %{INTERRUPT_ID} on new target %{AFFINITY} when receive SetTarget command. Fields:

AFFINITY unsigned int

Affinity ID.

INTERRUPT_ID unsigned int

Interrupt ID.

ArchMsg.Warning.gicv5_skip_processing_virtual_interrupt

DISPLAY IRS skipped processing virtual interrupt as %{REASON}. Fields:

REASON string

The reason of skipping the interrupt.

ArchMsg.Warning.gicv5_trace_event_access_invalid_vm

DISPLAY IRS access invalid VM: %{VM_ID}, where L2_VMTE.Valid=0. Fields:

VM_ID unsigned int

VM ID.

ArchMsg.Warning.gicv5_trace_event_access_invalid_vpe

DISPLAY IRS access invalid VPE: %{VPE_ID} on VM: %{VM_ID} as VPETE.Valid=0. Fields:

VM_ID unsigned int

VM ID.

VPE_ID unsigned int

VPE ID.

ArchMsg.Warning.gicv5_trying_to_select_pe_with_affinity_exceeding_the_supported_affinity_range

DISPLAY SW is trying to select a PE through 'IRS_PE_SEL' using affinity '%{AFFINITY}' which is exceeding the IRS's affinity supported bits '%{AFFINITY_BITS_SUPPORTED}'. Fields:

AFFINITY unsigned int

Affinity ID.

AFFINITY_BITS_SUPPORTED unsigned int

Affinity bits supported.

ArchMsg.Warning.gicv5_unable_to_fetch_unreachable_interrupt

DISPLAY Unable to fetch unreachable %{TARGET} %{INTERRUPT_TYPE} interrupt %{INTERRUPT_ID}%{PE_INFO}. Fields:

INFO string

Virtual interrupt info.

INTERRUPT_ID unsigned int

Interrupt ID.

INTERRUPT_TYPE enum

Interrupt type.

TARGET string

Physical or Virtual.

ArchMsg.Warning.gicv5_unable_to_find_target_channel

DISPLAY IRS unable to find target channel for interrupt %{INTERRUPT_ID} of type %{INTERRUPT_TYPE} targeting %{PE_INFO}. Fields:

INTERRUPT_ID unsigned int

Interrupt ID.

INTERRUPT_TYPE enum

Interrupt type.

PE_INFO string

PE or VPE info.

ArchMsg.Warning.gicv5_unable_to_get_channel_for_vpe

DISPLAY IRS is unable to get channel of residency for VPE %{VPE_ID} on VM %{VM_ID} because %{REASON}. Fields:

REASON string

Reason behind failure in getting vpe physical channel.

VM_ID unsigned int

VM ID.

VPE_ID unsigned int

VPE ID.

ArchMsg.Warning.gicv5_unable_to_get_residency_info

DISPLAY IRS is unable to get the residency info on PE with affinity %{AFFINITY}. Fields:

AFFINITY unsigned int

Affinity ID.

ArchMsg.Warning.gicv5_unable_to_retrieve_vm_table_entry

DISPLAY Unable to retrieve vm table entry for VM: %{VM_ID}. Fields:

VM_ID unsigned int

VM ID.

ArchMsg.Warning.gicv5_valid_irs_ist_baser_already_exists

DISPLAY When IRS_IST_BASER.Valid == 1, access to IRS_IST_BASER.Addr is RO, Currently address %{ADDR} is in use. Fields:

ADDR unsigned int

IST Address.

ArchMsg.Warning.gicv5_valid_irs_vmt_baser_already_exists

DISPLAY When IRS_VMT_BASER.Valid == 1, access to this IRS_VMT_BASER.Addr is RO, Currently address %{ADDR} is in use. Fields:

ADDR unsigned int

VMT Address.

ArchMsg.Warning.gicv5_valid_irs_vmt_baser_valid_is_already_zero

DISPLAY When IRS_VMT_BASER.Valid == 0, no change in valid field.

ArchMsg.Warning.gicv5_vpe_is_already_non_resident

DISPLAY VPE %{VPE_ID} on VM %{VM_ID} cannot be made non-resident as it is already non-resident. Fields:

VM_ID unsigned int

VM ID.

VPE_ID unsigned int

VPE ID.

ArchMsg.Warning.gicv5_vpe_is_already_resident_on_another_pe

DISPLAY VPE %{VPE_ID} on VM %{VM_ID} cannot be made resident on PE %{AFFINITY}, as it is already resident on PE %{RESIDENT}. Fields:

AFFINITY unsigned int

PE's affinity.

RESIDENT unsigned int

Current residency of VPE.

VM_ID unsigned int

VM ID.

VPE_ID unsigned int

VPE ID.

GICV5_ASSIGN_SPI_TO_DOMAIN

DISPLAY Assign SPI %{SPI_ID} to %{DOMAIN} domain. Fields:

DOMAIN enum

SPI Domain.

SPI_ID unsigned int

SPI ID.

GICV5_CLEARED_OUTSTANDING_COMMAND

DISPLAY IRS clear the outstanding command for PE with channel index %{CHANNEL_INDEX}. Fields:

CHANNEL_INDEX unsigned int

Channel Index of PE.

GICV5_CLEAR_ASSIGNMENT_PHYSICAL_SPI_TO_VM

DISPLAY Clear the assignment of the selected SPI %{SPI} to VM %{VM_ID}. Fields:

SPI unsigned int

Physical SPI ID.

VM_ID unsigned int

Virtual machine ID.

GICV5_CLEAR_PENDING_STATE_OF_ALREADY_ACTIVE_INTERRUPT

DISPLAY IRS received clear pending request for interrupt ID %{INTERRUPT_ID} while being in an 'Active' state. IRS will only clear the interrupt's pending state and skip any further processing. Fields:

INTERRUPT_ID unsigned int

Interrupt ID.

GICV5_DOORBELL_CONDITION_MET

DISPLAY IRS found all doorbell conditions are met for doorbell %{INTERRUPT_ID} for VPE %{VPE_ID} on VM %{VM_ID}. Fields:

INTERRUPT_ID unsigned int

Interrupt Id.

VM_ID unsigned int

VM ID.

VPE_ID unsigned int

VPE ID.

GICV5_DO_INVALIDATE_IST

DISPLAY Invalidate IST with address %{ADDR}. Fields:

ADDR unsigned int

IST Address.

GICV5_DO_INVALIDATE_VMT

DISPLAY Invalidate VMT with address %{ADDR}. Fields:

ADDR unsigned int

VMT Address.

GICV5_FOUND_CHANNEL_FOR_RESIDENT_VPE

DISPLAY IRS channel index %{CHANNEL_INDEX} for resident VPE %{VPE_ID} on VM %{VM_ID}. Fields:

CHANNEL_INDEX unsigned int

The hannel index where the VPE resident.

VM_ID unsigned int

VM ID.

VPE_ID unsigned int

VPE ID.

GICV5_FOUND_CONFIGURED_AND_REQUESTED_DOORBELL

DISPLAY IRS found LPI %{LPI_ID} configured and requested as a doorbell for VPE %{VPE_ID} on VM %{VM_ID}. Fields:

LPI_ID unsigned int

Doorbell INTID.

VM_ID unsigned int

VM ID.

VPE_ID unsigned int

VPE ID.

GICV5_FOUND_PHYSICAL_AFFINITY_FOR_RESIDENT_VPE

DISPLAY IRS found Physical Affinity %{AFFINITY} for resident VPE %{VPE_ID} on VM %{VM_ID}. Fields:

AFFINITY unsigned int

The physical affinity where the VPE resident.

VM_ID unsigned int

VM ID.

VPE_ID unsigned int

VPE ID.

GICV5_IGNORE_CLEARING_PE_OUTSTANDING_INTERRUPTS

DISPLAY Ignore clearing outstanding interrupts for offline PE at channel index %**{CHANNEL_INDEX}**. Fields:

CHANNEL_INDEX unsigned int

Channel Index of PE.

GICV5_IRS_FOUND_VPE_NOT_RESIDENT

DISPLAY IRS found that VPE %**{VPE_ID}** on VM %**{VM_ID}** is not resident. Fields:

VM_ID unsigned int

VM ID.

VPE_ID unsigned int

VPE ID.

GICV5_IRS_HPPI_SELECTION_OUTCOME

DISPLAY IRS trying to find HPPI for PE with affinity %**{AFFINITY}** and the outcome is that '%**{OUTCOME}**'. Fields:

AFFINITY unsigned int

PE's affinity.

OUTCOME string

The outcome of IRS selection trial.

GICV5_IRS_IST_CONFIGURATION_SYNC_COMPLETED

DISPLAY Sync configuration for IRS IST with address %**{ADDR}** completed. Fields:

ADDR unsigned int

IST Address.

GICV5_IRS_NOMINATE_NEW_1ofN_VIRTUAL_INTERRUPT

DISPLAY found new one_of_n hppi: %**{NEW_HPPI}** to VPE %**{VPE_ID}** on VM %**{VM_ID}**. Fields:

NEW_HPPI unsigned int

New Interrupt ID.

VM_ID unsigned int

VM Id.

VPE_ID unsigned int

VPE Id.

GICV5_IRS_SKIP_FORWARDING_HPPI_AS_IT_WAS_ALREADY_FORWARDED_TO_CPUIF

DISPLAY IRS skip forwarding HPPI to the PE on channel index %{CHANNEL_INDEX} with affinity %{AFFINITY} because it was already forwarded to CPUIF. Fields:

AFFINITY unsigned int

The affinity of the PE.

CHANNEL_INDEX unsigned int

The channel index of the PE.

GICV5_IRS_SKIP_FORWARDING_VIRTUAL_HPPI_AS_IT_WAS_ALREADY_FORWARDED_TO_CPUIF

DISPLAY IRS skip forwarding virtual HPPI to VPE %{VPE_ID} on VM %{VM_ID} because it was already forwarded. Fields:

VM_ID unsigned int

VM Id.

VPE_ID unsigned int

VPE Id.

GICV5_IRS_SKIP_VPE_AS_IT_BELONGS_TO_ANOTHER_VM

DISPLAY The resident VPE %{VPE_ID} on channel '%{CHANNEL_INDEX}' belongs to VM-%{VM_ID}, but we are %{OPERATION} on VM-%{TARGET_VM_ID}. Fields:

CHANNEL_INDEX unsigned int

Channel index.

OPERATION string

RecallingInterrupts or ForwardingInterrupts.

TARGET_VM_ID unsigned int

VM ID.

VM_ID unsigned int

VM ID.

VPE_ID unsigned int

VPE ID.

GICV5_IRS_TABLES_CONFIGS_SYNC_COMPLETED

DISPLAY IRS %{TABLE_TYPE} register %{CFGR_REGISTER} and register %{BASER_REGISTER} of IRSID %{ID} is in sync with the value of %{CFGR_VALUE} and %{BASER_VALUE} respectively. Fields:

BASER_REGISTER string

irs ist or vmt baser register.

BASER_VALUE unsigned int

new irs ist or vmt baser register value after sync .

CFGR_REGISTER string

irs ist or vmt configuration register.

CFGR_VALUE unsigned int

new irs ist or vmt cfgr register value after sync .

ID unsigned int

irs id.

GICV5_IRS_TABLE_READ

DISPLAY IRS read an %{ENTRY} at %{ADDRESS}. Fields:

ADDRESS unsigned int

Address of the entry.

ENTRY string

IRS Table Entry.

GICV5_IRS_TABLE_WRITE

DISPLAY IRS write an %{ENTRY} at %{ADDRESS}. Fields:

ADDRESS unsigned int

Address of the entry.

ENTRY string

IRS Table Entry.

GICV5_IRS_TRIGGER_LPI_THROUGH_SETLPI_REGISTER

DISPLAY SW trigger LPI '%{LPI_ID}' through SETLPI register on IRS '%{IRSID}'. Fields:

IRSID unsigned int

IRS ID.

LPI_ID unsigned int

LPI ID.

GICV5_IRS_VMT_CONFIGURATION_SYNC_COMPLETED

DISPLAY Sync configuration for IRS VMT with address %{ADDR} completed. Fields:

ADDR unsigned int

VMT Address.

GICV5_IST_IN_USE

DISPLAY IST with address %{ADDR} is in use. Fields:

ADDR unsigned int

IST address.

GICV5_NO_CHANGE_TO_IRS_STATE

DISPLAY No change occurs to IRS state after writing on register IRS_CRO value %{VALUE}. Fields:

VALUE unsigned int

Value of attempted write.

GICV5_NO_EFFECT_ON_HPPI_SELECTION

DISPLAY IRS HPPI selection did not change as a result of change in the %{CHANGE_ASPECT} of interrupt %{INTERRUPT_ID}. Fields:

CHANGE_ASPECT string

Interrupt property.

INTERRUPT_ID unsigned int

Interrupt ID.

GICV5_NO_OUTSTANDING_INTERRUPTS_FOUND_FOR_THE_RESIDENT_VPE

DISPLAY IRS found no outstanding interrupts for VPE %{VPE_ID} on VM %{VM_ID}. Fields:

VM_ID unsigned int

VM ID.

VPE_ID unsigned int

VPE ID.

GICV5_NO_VPE_FOUND_ON_IRS

DISPLAY IRS did not find a VPE %{VPE_ID} on VM %{VM_ID} on this local IRS. Fields:

VM_ID unsigned int

VM Id.

VPE_ID unsigned int

VPE Id.

GICV5_NO_VPE_RESIDENT_ON_PE

DISPLAY IRS did not find a resident VPE on channel %{CHANNEL_INDEX}. Fields:

CHANNEL_INDEX unsigned int

Channel Index.

GICV5_PHYSICAL_SPI_IS_NOT_ASSIGNED_TO_VM

DISPLAY The selected SPI %{SPI} is not assigned to any VM. Fields:

SPI unsigned int

SPI ID.

GICV5_REGISTER_READ

DISPLAY %{IS_DEBUG:([DEBUG])} Read value %{VALUE} from 32-bit register %{REG_NAME}. Fields:

IS_DEBUG bool

Whether this access is from the model's debug interface (CADI/Iris).

REG_NAME string

Name of the register being read.

VALUE unsigned int

Value returned.

GICV5_REGISTER_READ64

DISPLAY %{IS_DEBUG:([DEBUG])} Read value %{VALUE} from 64-bit register %{REG_NAME}. Fields:

IS_DEBUG bool

Whether this access is from the model's debug interface (CADI/Iris).

REG_NAME string

Name of the register being read.

VALUE unsigned int

Value returned.

GICV5_REGISTER_UPDATE

DISPLAY 32-bit register %{REG_NAME} updated to %{NEW_VALUE}. Fields:

NEW_VALUE unsigned int

Value after the update.

REG_NAME string

Name of register that was updated.

GICV5_REGISTER_UPDATE64

DISPLAY 64-bit register %{REG_NAME} updated to %{NEW_VALUE}. Fields:

NEW_VALUE unsigned int

Value after the update.

REG_NAME string

Name of register that was updated.

GICV5_REGISTER_WRITE

DISPLAY %{IS_DEBUG:([DEBUG])} Write value %{VALUE} to %{REG_NAME}. Previous: %{PREV_VALUE}. New: %{UPDATED_VALUE}. Fields:

IS_DEBUG bool

Whether this access is from the model's debug interface (CADI/Iris).

PREV_VALUE unsigned int

Value in the register before the write.

REG_NAME string

Name of the register being written.

UPDATED_VALUE unsigned int

New value read back from the register; for write-only registers this will be equal to VALUE.

VALUE unsigned int

Value written.

GICV5_REGISTER_WRITE64

DISPLAY `%{IS_DEBUG:([[DEBUG]])}` Write value `%{VALUE}` to `%{REG_NAME}`. Previous: `%{PREV_VALUE}`. New: `%{UPDATED_VALUE}`. Fields:

IS_DEBUG bool

Whether this access is from the model's debug interface (CADI/Iris).

PREV_VALUE unsigned int

Value in the register before the write.

REG_NAME string

Name of the register being written.

UPDATED_VALUE unsigned int

New value read back from the register; for write-only registers this will be equal to `VALUE`.

VALUE unsigned int

Value written.

GICV5_RESET_PHYSICAL_SPI_AND_ASSIGN_IT_TO_VM

DISPLAY IRS reseted the selected physical SPI `%{SPI}` and assigned it successfully to VM `%{VM_ID}`. Fields:

SPI unsigned int

Physical SPI ID.

VM_ID unsigned int

Virtual machine ID.

GICV5_SKIP_INTERRUPT_ENABLE_REQUEST

DISPLAY Skip enable request, as there is no change to the enable state of interrupt `%{INTERRUPT_ID}` which is `%{OLD_ENABLE_STATE}`. Fields:

INTERRUPT_ID unsigned int

Interrupt ID.

OLD_ENABLE_STATE bool

Old interrupt enable state.

GICV5_SKIP_INTERRUPT_PENDING_STATE_CHANGE

DISPLAY Skip set-pending request, as there is no change to the pending state of interrupt `%{INTERRUPT_ID}` as a result of event `%{EVENT_TYPE}`, as the pending state is already `%{IS_PENDING}`. Fields:

EVENT_TYPE string

The type of interrupt event.

INTERRUPT_ID unsigned int

Interrupt ID.

IS_PENDING bool

Pending state of interrupt.

GICV5_SKIP_PROCESSING_INVALID_INTERRUPT_EVENT_TYPE

DISPLAY Skip processing interrupt %{INTERRUPT_ID} as the event-type is invalid. Fields:

INTERRUPT_ID unsigned int

Interrupt ID.

GICV5_SKIP_RECALLING_INTERRUPT

DISPLAY IRS skip recalling the outstanding %{INTERRUPT_TYPE} interrupt as it is recalling %{RECALLED_INTERRUPT_TYPE} interrupts only. Fields:

INTERRUPT_TYPE enum

Interrupt type.

RECALLED_INTERRUPT_TYPE enum

Interrupt type.

GICV5_SKIP_REMOVING_PENDING_HPPI_1ofN_INTERRUPT_ON_OLD_TARGET

DISPLAY skip removing pending, hppi and 1ofN interrupt %{INTERRUPT_ID} on old target %{AFFINITY} when receive SetTarget command. Fields:

AFFINITY unsigned int

Affinity ID.

INTERRUPT_ID unsigned int

Interrupt ID.

GICV5_SKIP_SENDING_1ofN_DOORBELL

DISPLAY IRS will skip sending 1ofN doorbell (if exists) because %{REASON} for VM_ID: %{VM_ID} . Fields:

REASON string

The reason of skipping doorbell.

VM_ID unsigned int

VM ID.

GICV5_SKIP_SENDING_VPE_DOORBELL

DISPLAY IRS will skip sending VPE's doorbell (if exists) because %{REASON}. Fields:

REASON string

The reason of skipping doorbell.

GICV5_SKIP_SET_TARGET_COMMAND

DISPLAY skip SetTarget command for %{INTERRUPT_TYPE} %{INTERRUPT_ID} as requested new affinity id and routing mode is same as before. Fields:

INTERRUPT_ID unsigned int

Interrupt ID.

INTERRUPT_TYPE enum

Interrupt type.

GICV5_SPI_SELECTION_DONE_SUCCESSFULLY

DISPLAY SPI %{SPI_ID} selection is done successfully. Fields:

SPI_ID unsigned int

SPI ID.

GICV5_START_HANDLING_ITS_EVENT

DISPLAY IRS start handling ITS event for INTID %{INTERRUPT_ID}, EventType %{EVENT_TYPE} and Virtual=\${VIRTUAL}. Fields:

EVENT_TYPE string

Event Type.

INTERRUPT_ID unsigned int

Interrupt ID.

VIRTUAL bool

Is virtual event.

GICV5_START_PROCESSING_INTERRUPT_EVENT

DISPLAY Start processing interrupt event '%{EVENT_TYPE}' for INTID %{INTERRUPT_ID} with type '%{TYPE}'. Fields:

EVENT_TYPE enum

The type of interrupt event.

INTERRUPT_ID unsigned int

Interrupt ID.

TYPE enum

Interrupt Type.

GICV5_TRANSITION_COMPLETE

DISPLAY The effects of updating the IRSEN field of IRS_CRO have completed and domain transitioned to %{STATE} state. Fields:

STATE string

New Transition state.

GICV5_TRIGGER_LPI_INTERRUPT_OUT_OF_RANGE

DISPLAY Triggering %{INTERRUPT_KIND} %{INTERRUPT_TYPE} %{INTERRUPT_ID} which is out of range. Fields:

INTERRUPT_ID unsigned int

Interrupt ID.

INTERRUPT_TYPE enum

Interrupt type.

KIND enum

Interrupt kind.

GICV5_UNABLE_TO_COMMUNICATE_WITH_NON_RESIDENT_VPE

DISPLAY IRS unable to communicate with VPE %{VPE_ID} on VM %{VM_ID} for %{CMD} command, as the VPE is not resident. Fields:

CMD string

Upstream command.

VM_ID unsigned int

VM ID.

VPE_ID unsigned int

VPE ID.

GICV5_UNABLE_TO_COMMUNICATE_WITH_NON_RESIDENT_VPE_TO_CHANGE_STATE

DISPLAY IRS unable to communicate with VPE %{VPE_ID} on VM %{VM_ID}, as the VPE is not resident. Fields:

VM_ID unsigned int

VM ID.

VPE_ID unsigned int

VPE ID.

GICV5_UNABLE_TO_FIND_VPE_SUPPORT_lofN_INTERRUPTS

DISPLAY IRS unable to find vpe on VM %{VM_ID} for interrupt %{INTERRUPT_ID}. Fields:

INTERRUPT_ID unsigned int

Interrupt ID.

VM_ID unsigned int

VM ID.

GICV5_UNABLE_TO_GET_PHYSICAL_AFFINITY_OF_RESIDENCY_FOR_VPE

DISPLAY IRS is unable to get physical affinity of residency for VPE %{VPE_ID} on VM %{VM_ID} because %{REASON}. Fields:

REASON string

Reason behind failure in getting physical affinity for VPE.

VM_ID unsigned int

VM ID.

VPE_ID unsigned int

VPE ID.

GICV5_UNABLE_TO_NOMINATE_lofN_VIRTUAL_INTERRUPT

DISPLAY IRS is unable to nominate one-of-n virtual interrupt for VPE %{VPE_ID} on VM %{VM_ID} because %{REASON}. Fields:

REASON string

Reason behind failure in getting vpe physical channel.

VM_ID unsigned int

VM ID.

VPE_ID unsigned int

VPE ID.

GICV5_UPSTREAM_COMMAND_TARGETING_NON_RESIDENT_VPE

DISPLAY IRS received %{CMD} command for %{INTERRUPT_TYPE} %{INTERRUPT_ID} targeting non-resident %{PE_INFO}. Fields:

CMD string

Upstream command.

INTERRUPT_ID unsigned int

Interrupt ID.

INTERRUPT_TYPE enum

Interrupt type.

PE_INFO string

PE or VPE info.

GICV5_VIRTUAL_INTERRUPT_PRIORITY_HIGHER_THAN_OR_EQUAL_DOORBELL_PRIORITY_MASK

DISPLAY IRS found virtual interrupt with priority '%{VIRT_INT_PRIORITY}' which is higher than or equal to doorbell priority mask '%{DOORBELL_PRIORITY_MASK}' for non-resident VPE %{VPE_ID} on VM %{VM_ID}. Fields:

DOORBELL_PRIORITY_MASK unsigned int

Doorbell priority mask.

VIRT_INT_PRIORITY unsigned int

Virtual interrupt priority.

VM_ID unsigned int

VM ID.

VPE_ID unsigned int

VPE ID.

GICV5_VIRTUAL_INTERRUPT_PRIORITY_LOWER_THAN_DOORBELL_PRIORITY_MASK

DISPLAY IRS found virtual %{INTERRUPT_TYPE} %{INTERRUPT_ID} with priority '%{VIRT_INT_PRIORITY}' which is lower than doorbell priority mask '%{DOORBELL_PRIORITY_MASK}' for non-resident VPE %{VPE_ID} on VM %{VM_ID}. Fields:

DOORBELL_PRIORITY_MASK unsigned int

Doorbell priority mask.

INTERRUPT_ID unsigned int

Interrupt ID.

INTERRUPT_TYPE enum

Interrupt type.

VIRT_INT_PRIORITY unsigned int

Virtual interrupt priority.

VM_ID unsigned int

VM ID.

VPE_ID unsigned int

VPE ID.

GICV5_VMT_IN_USE

DISPLAY VMT with address %{ADDR} is in use. Fields:

ADDR unsigned int

VMT address.

Reset

Component changed reset state. Fields:

RESET bool

Entered/left reset state.

2.120 ITS

This section describes the trace sources.

GICV5_MEMORY_MAPPING

DISPLAY print the memory mapping for ITS '%{FRAME_NAME}' frame on '%{ITSDOMAIN}' domain Match first %{MF}, Match last %{ML}, Remap first {RF}, Remap last %{RL}. Fields:

FRAME_NAME string

IRS frame name.

ITSDOMAIN enum

IRSDomain.

MF unsigned int

Match first for map range.

ML unsigned int

Match last for map range.

RF unsigned int

ReMap first for map range.

RL unsigned int

ReMap last for map range.

2.121 ITSDomain

This section describes the trace sources.

ArchMsg.Warning.gicv5_cross_domain_translation_failed_no_permission

DISPLAY Cross-domain translation of event with DeviceID %{DEVICE_ID}, EventID %{EVENT_ID} failed because the ITTE lacks required permission. Fields:

DEVICE_ID unsigned int

DeviceID of the interrupt event.

EVENT_ID unsigned int

EventID of the interrupt event.

ArchMsg.Warning.gicv5_cross_domain_translation_not_supported

DISPLAY Write to %{REGISTER} is ignored because Realm domain is not implemented or does not support translation of Non-Secure events. Fields:

REGISTER string

Name of the register being written.

ArchMsg.Warning.gicv5_cross_domain_translation_wrong_domain

DISPLAY Write to ITS_RL_TRANSLATER with DeviceID %{DEVICE_ID}, EventID %{EVENT_ID} is ignored because it targets domain other than Non-Secure. Fields:

DEVICE_ID unsigned int

DeviceID of the interrupt event.

EVENT_ID unsigned int

EventID of the interrupt event.

ArchMsg.Warning.gicv5_device_id_bits_forced_to_maximum

DISPLAY Effective value of DeviceID bits is %{EFFECTIVE} as reported in ITS_IDR1.DeviceID_Bits instead of %{CONFIGURED} as configured in ITS_DT_CFGR.DeviceID_Bits. Fields:

CONFIGURED unsigned int

DeviceID bits requested in ITS_DT_CFGR.

EFFECTIVE unsigned int

Maximum supported number of DeviceID bits as reported in ITS_IDR1.

ArchMsg.Warning.gicv5_event_forced_physical_in_el3

DISPLAY Treating event with DeviceID %{DEVICE_ID}, EventID %{EVENT_ID} as Physical because it targets EL3. Fields:

DEVICE_ID unsigned int

DeviceID of the interrupt event.

EVENT_ID unsigned int

EventID of the interrupt event.

ArchMsg.Warning.gicv5_event_id_bits_forced_to_maximum

DISPLAY Effective value of EventID bits for DeviceID %{DEVICE_ID} is %{EFFECTIVE} as reported in ITS_IDR2.EventID_Bits instead of %{CONFIGURED} as configured in L2_DTE.EventID_Bits. Fields:

CONFIGURED unsigned int

EventID bits requested in L2_DTE.

DEVICE_ID unsigned int

The DeviceID of the device whose ITT is affected.

EFFECTIVE unsigned int

Maximum supported number of EventID bits as reported in ITS_IDR2.

ArchMsg.Warning.gicv5_event_ignored_domain_disabled

DISPLAY ITS received event with DeviceID %{DEVICE_ID}, EventID %{EVENT_ID} targeting disabled interrupt domain. Fields:

DEVICE_ID unsigned int

DeviceID of the interrupt event.

EVENT_ID unsigned int

EventID of the interrupt event.

ArchMsg.Warning.gicv5_ignore_register_access

DISPLAY Access to register %{REG} has been ignored, and the reason is that %{REASON}. Fields:

REASON string

The reason of ignoring the access.

REG string

The accessed register.

ArchMsg.Warning.gicv5_its_bad_l2_split

DISPLAY ITS %{TABLE_TYPE:(DT|ITT)} at %{ADDRESS} received a reserved or unsupported Split value: %{VALUE}. Fields:

ADDRESS unsigned int

ADDRESS base of the table.

TABLE_TYPE enum

Type of table.

VALUE unsigned int

The invalid Split value.

ArchMsg.Warning.gicv5_its_table_address_alignment

DISPLAY ITS %{TABLE_TYPE:(DT|ITT)} configured to be at %{CONF_ADDR} but effective address is %{EFF_ADDR} due to alignment requirements. Fields:

CONF_ADDR unsigned int

Configured base address of the table.

EFF_ADDR unsigned int

Effective base address of the table.

TABLE_TYPE enum

Type of table.

ArchMsg.Warning.gicv5_its_table_read_failed

DISPLAY ITS could not read a %{TABLE_TYPE:(DT|ITT)} entry at %{ADDRESS}. Fields:

ADDRESS unsigned int

ADDRESS of the table entry.

TABLE_TYPE enum

Type of table.

ArchMsg.Warning.gicv5_memory_mapped_access_reserved

DISPLAY %{IS_WRITE:(Read|Write)} access to reserved offset %{OFFSET} within a register frame. Fields:

IS_WRITE bool

Whether this is a write access.

OFFSET unsigned int

Offset of address within the GICv5 register frame.

ArchMsg.Warning.gicv5_memory_mapped_access_wider_than_size

DISPLAY 64-bit %{IS_WRITE:(read|write)} access affects on one or more 32-bit registers at offset %{OFFSET}. Fields:

IS_WRITE bool

Whether this is a write access.

OFFSET unsigned int

Offset of address within the GICv5 register frame.

ArchMsg.Warning.gicv5_memory_mapped_illegal_access_width

DISPLAY %{IS_WRITE:(Read|Write)} access to memory-mapped interface using an illegal access width of %{WIDTH} bits. Fields:

IS_WRITE bool

Whether this is a write access.

WIDTH unsigned int

Access width in bits.

ArchMsg.Warning.gicv5_memory_mapped_read_write_only_reg

DISPLAY Attempt to read from write-only register %{REG_NAME}. Fields:

REG_NAME string

Name of the register being read.

ArchMsg.Warning.gicv5_memory_mapped_write64_read_only_reg

DISPLAY Attempt to write value %{VALUE} to read-only register %{REG_NAME}. Fields:

REG_NAME string

Name of the register being written.

VALUE unsigned int

Value written.

ArchMsg.Warning.gicv5_memory_mapped_write_read_only_reg

DISPLAY Attempt to write value %{VALUE} to read-only register %{REG_NAME}. Fields:

REG_NAME string

Name of the register being written.

VALUE unsigned int

Value written.

ArchMsg.Warning.gicv5_pas_filtering

DISPLAY Ignore %{IS_WRITE:(Read|Write)} operation performed by software in %{TX_PAS} PAS to a memory mapped register which expect access from %{EX_PAS} PAS at address %{ADDR}. Fields:

ADDR unsigned int

Transaction Address.

EX_PAS string

Expected PAS.

IS_WRITE bool

Whether this is a write access.

TX_PAS string

PAS associated with Read or Write Transaction.

ArchMsg.Warning.gicv5_translation_failed_invalid_entry

DISPLAY Translation of event with DeviceID %{DEVICE_ID}, EventID %{EVENT_ID} failed. L %{LEVEL:d}_%{TABLE_TYPE:(DTE|ITTE)} is invalid. Fields:

DEVICE_ID unsigned int

DeviceID of the interrupt event.

EVENT_ID unsigned int

EventID of the interrupt event.

LEVEL unsigned int

The level of the table entry (1 or 2).

TABLE_TYPE enum

Type of table.

ArchMsg.Warning.gicv5_use_of_reserved_value

DISPLAY Value %{VAL} written in %{REG}. %{FIELD} is reserved. Fields:

FIELD string

Impacted register field.

REG string

Impacted register name.

VAL unsigned int

Value written.

ArchMsg.Warning.gicv5_write_ignored_domain_disabled

DISPLAY Write with value %{VALUE} to %{REGISTER} ignored because the domain was disabled. Fields:

REGISTER string

Name of accessed register.

VALUE unsigned int

Value of attempted write.

ArchMsg.Warning.gicv5_write_ignored_domain_enabled

DISPLAY Write with value %{VALUE} to %{REGISTER} ignored because the domain was enabled. Fields:

REGISTER string

Name of accessed register.

VALUE unsigned int

Value of attempted write.

GICV5_ITS_TABLE_CACHE_ADD

DISPLAY ITS table %{TABLE_TYPE:(DT|ITT)} at %{BASE_ADDR} cached L%{LEVEL:d}_%{TABLE_TYPE:(DT|ITT)}E with ID %{ID}. Fields:

BASE_ADDR unsigned int

The start of the whole table.

ID unsigned int

ID the entry describes.

LEVEL unsigned int

The level of the table entry (1 or 2).

TABLE_TYPE enum

Type of table.

GICV5_ITS_TABLE_CACHE_DROP

DISPLAY ITS table %{TABLE_TYPE:(DT|ITT)} at %{BASE_ADDR} dropped L%{LEVEL:d}_%{TABLE_TYPE:(DT|ITT)}E with ID %{ID}. Fields:

BASE_ADDR unsigned int

The start of the whole table.

ID unsigned int

ID the entry describes.

LEVEL unsigned int

The level of the table entry (1 or 2).

TABLE_TYPE enum

Type of table.

GICV5_ITS_TABLE_READ

DISPLAY ITS read an %{ENTRY} at %{ADDRESS}. Fields:

ADDRESS unsigned int

Address of the entry.

ENTRY string

IRS Table Entry.

GICV5_REGISTER_READ

DISPLAY %{IS_DEBUG:([DEBUG])} Read value %{VALUE} from 32-bit register %{REG_NAME}. Fields:

IS_DEBUG bool

Whether this access is from the model's debug interface (CADI/Iris).

REG_NAME string

Name of the register being read.

VALUE unsigned int

Value returned.

GICV5_REGISTER_READ64

DISPLAY %{IS_DEBUG:([DEBUG])} Read value %{VALUE} from 64-bit register %{REG_NAME}. Fields:

IS_DEBUG bool

Whether this access is from the model's debug interface (CADI/Iris).

REG_NAME string

Name of the register being read.

VALUE unsigned int

Value returned.

GICV5_REGISTER_UPDATE

DISPLAY 32-bit register %{REG_NAME} updated to %{NEW_VALUE}. Fields:

NEW_VALUE unsigned int

Value after the update.

REG_NAME string

Name of register that was updated.

GICV5_REGISTER_UPDATE64

DISPLAY 64-bit register %{REG_NAME} updated to %{NEW_VALUE}. Fields:

NEW_VALUE unsigned int

Value after the update.

REG_NAME string

Name of register that was updated.

GICV5_REGISTER_WRITE

DISPLAY %{IS_DEBUG:([DEBUG])} Write value %{VALUE} to %{REG_NAME}. Previous: %{PREV_VALUE}. New: %{UPDATED_VALUE}. Fields:

IS_DEBUG bool

Whether this access is from the model's debug interface (CADI/Iris).

PREV_VALUE unsigned int

Value in the register before the write.

REG_NAME string

Name of the register being written.

UPDATED_VALUE unsigned int

New value read back from the register; for write-only registers this will be equal to VALUE.

VALUE unsigned int

Value written.

GICV5_REGISTER_WRITE64

DISPLAY %{IS_DEBUG:([DEBUG])} Write value %{VALUE} to %{REG_NAME}. Previous: %{PREV_VALUE}. New: %{UPDATED_VALUE}. Fields:

IS_DEBUG bool

Whether this access is from the model's debug interface (CADI/Iris).

PREV_VALUE unsigned int

Value in the register before the write.

REG_NAME string

Name of the register being written.

UPDATED_VALUE unsigned int

New value read back from the register; for write-only registers this will be equal to VALUE.

VALUE unsigned int

Value written.

GICV5_SEND_PHYSICAL_EVENT_TO_IRS

DISPLAY Sending a physical event %{INTERRUPT_ID} of type %{EVENT_TYPE} to the IRS.
Fields:

EVENT_TYPE enum

The type of interrupt event.

INTERRUPT_ID unsigned int

ID of the interrupt event.

GICV5_SEND_VIRTUAL_EVENT_TO_IRS

DISPLAY Sending a virtual event %{INTERRUPT_ID} of type %{EVENT_TYPE} with VMID \${VM_ID} to the IRS. Fields:

EVENT_TYPE enum

The type of interrupt event.

INTERRUPT_ID unsigned int

ID of the interrupt event.

VM_ID unsigned int

The Virtual Machine Identifier.

Reset

Component changed reset state. Fields:

RESET bool

Entered/left reset state.

2.122 IWB

This section describes the trace sources.

ArchMsg.Warning.gicv5_enabled_wire_trigger_change

Wire's trigger mode changed while it was enabled. Fields:

WIRE_NUM unsigned int

Number of the input wire.

ArchMsg.Warning.gicv5_ignore_iwb_wresampler_access

DISPLAY Access to IWB_WRESAMPLER register ignored for wire number %{WIRE_NUM}, and the reason is that %{REASON}. Fields:

DOMAIN enum

Interrupt domain.

PAS enum

Physical address space.

REASON string

The reason to ignore the access.

WIRE_NUM unsigned int

Number of the input wire.

ArchMsg.Warning.gicv5_invalid_pas_access

Register accessed using an unsupported PAS. Fields:

PAS enum

Physical address space.

READ/WRITE enum

Access is read or write.

REG_NAME string

Name of the accessed register.

ArchMsg.Warning.gicv5_invalid_wire_domain

Wire was assigned an unsupported domain. Fields:

DOMAIN enum

Interrupt domain.

WIRE_NUM unsigned int

Number of the input wire.

ArchMsg.Warning.gicv5_memory_mapped_access_reserved

DISPLAY %{IS_WRITE:(Read|Write)} access to reserved offset %{OFFSET} within a register frame. Fields:

IS_WRITE bool

Whether this is a write access.

OFFSET unsigned int

Offset of address within the GICv5 register frame.

ArchMsg.Warning.gicv5_memory_mapped_access_wider_than_size

DISPLAY 64-bit %{IS_WRITE:(read|write)} access affects on one or more 32-bit registers at offset %{OFFSET}. Fields:

IS_WRITE bool

Whether this is a write access.

OFFSET unsigned int

Offset of address within the GICv5 register frame.

ArchMsg.Warning.gicv5_memory_mapped_illegal_access_width

DISPLAY %{IS_WRITE:(Read|Write)} access to memory-mapped interface using an illegal access width of %{WIDTH} bits. Fields:

IS_WRITE bool

Whether this is a write access.

WIDTH unsigned int

Access width in bits.

ArchMsg.Warning.gicv5_memory_mapped_read_write_only_reg

DISPLAY Attempt to read from write-only register %{REG_NAME}. Fields:

REG_NAME string

Name of the register being read.

ArchMsg.Warning.gicv5_memory_mapped_write64_read_only_reg

DISPLAY Attempt to write value %{VALUE} to read-only register %{REG_NAME}. Fields:

REG_NAME string

Name of the register being written.

VALUE unsigned int

Value written.

ArchMsg.Warning.gicv5_memory_mapped_write_read_only_reg

DISPLAY Attempt to write value %{VALUE} to read-only register %{REG_NAME}. Fields:

REG_NAME string

Name of the register being written.

VALUE unsigned int

Value written.

ArchMsg.Warning.gicv5_pas_filtering

DISPLAY Ignore %{IS_WRITE:(Read|Write)} operation performed by software in %{TX_PAS} PAS to a memory mapped register which expect access from %{EX_PAS} PAS at address %{ADDR}. Fields:

ADDR unsigned int

Transaction Address.

EX_PAS string

Expected PAS.

IS_WRITE bool

Whether this is a write access.

TX_PAS string

PAS associated with Read or Write Transaction.

GICV5_IGNORED_WIRE_DISABLED

IWB ingored a wire event because the wire is disabled. Fields:

WIRE_NUM unsigned int

Number of the input wire.

GICV5_IGNORED_WIRE_DURING_RESET

IWB received and ingored a wire event while in reset state. Fields:

WIRE_NUM unsigned int

Number of the input wire.

GICV5_IGNORED_WIRE_IWB_DISABLED

IWB ingored a wire event because the IWB is disabled. Fields:

WIRE_NUM unsigned int

Number of the input wire.

GICV5_INPUT_WIRE_CHANGED_STATE

IWB input wire changed state. Fields:

STATE enum

The state it changed to.

WIRE_NUM unsigned int

Number of the input wire.

GICV5_IWB_ENABLED

IWB changed enabled state. Fields:

ENABLED bool

Changed to enabled or disabled.

GICV5_NON_MPPAS_ACCESS_REJECTED

PAS other than MPPAS was rejected an access. Fields:

PAS enum

Physical address space.

READ/WRITE enum

Access is read or write.

REG_NAME string

Name of the accessed register.

GICV5_REGISTER_READ

DISPLAY %{IS_DEBUG:([DEBUG])} Read value %{VALUE} from 32-bit register
%{REG_NAME}. Fields:

IS_DEBUG bool

Whether this access is from the model's debug interface (CADI/Iris).

REG_NAME string

Name of the register being read.

VALUE unsigned int

Value returned.

GICV5_REGISTER_READ64

DISPLAY %{IS_DEBUG:([[DEBUG]])} Read value %{VALUE} from 64-bit register %{REG_NAME}. Fields:

IS_DEBUG bool

Whether this access is from the model's debug interface (CADI/Iris).

REG_NAME string

Name of the register being read.

VALUE unsigned int

Value returned.

GICV5_REGISTER_UPDATE

DISPLAY 32-bit register %{REG_NAME} updated to %{NEW_VALUE}. Fields:

NEW_VALUE unsigned int

Value after the update.

REG_NAME string

Name of register that was updated.

GICV5_REGISTER_UPDATE64

DISPLAY 64-bit register %{REG_NAME} updated to %{NEW_VALUE}. Fields:

NEW_VALUE unsigned int

Value after the update.

REG_NAME string

Name of register that was updated.

GICV5_REGISTER_WRITE

DISPLAY %{IS_DEBUG:([[DEBUG]])} Write value %{VALUE} to %{REG_NAME}. Previous: %{PREV_VALUE}. New: %{UPDATED_VALUE}. Fields:

IS_DEBUG bool

Whether this access is from the model's debug interface (CADI/Iris).

PREV_VALUE unsigned int

Value in the register before the write.

REG_NAME string

Name of the register being written.

UPDATED_VALUE unsigned int

New value read back from the register; for write-only registers this will be equal to VALUE.

VALUE unsigned int

Value written.

GICV5_REGISTER_WRITE64

DISPLAY %{IS_DEBUG:([[DEBUG]])} Write value %{VALUE} to %{REG_NAME}. Previous: %{PREV_VALUE}. New: %{UPDATED_VALUE}. Fields:

IS_DEBUG bool

Whether this access is from the model's debug interface (CADI/Iris).

PREV_VALUE unsigned int

Value in the register before the write.

REG_NAME string

Name of the register being written.

UPDATED_VALUE unsigned int

New value read back from the register; for write-only registers this will be equal to VALUE.

VALUE unsigned int

Value written.

GICV5_SEND_EVENT_TO_ITS

IWB sent an interrupt event to ITS. Fields:

DEVICE_ID unsigned int

Device ID of the IWB.

DOMAIN enum

Interrupt domain.

EVENT_TYPE enum

The type of event.

WIRE_IDX unsigned int

Index of the input wire used as EventID.

Reset

Component changed reset state. Fields:

RESET bool

Entered/left reset state.

2.123 MMU_400_BASE

This section describes the trace sources.

aarch64_AddressSizeFault_base_address_out_of_range

An AArch64 context got an AddressSizeFault as the base address of the TTBR used is out of range. Fields:

CPAMax unsigned int

Configured PAMax.

PAMax unsigned int

Actual PAMax.

context_bank unsigned int

context_bank.

stage unsigned int

Stage.

aarch64_AddressSizeFault_in_final_block_address_trace

An AArch64 context got an AddressSizeFault in the final block descriptor. Fields:

CPAMax unsigned int

Configured PAMax.

PAMax unsigned int

Actual PAMax.

context_bank unsigned int

context_bank.

output_address unsigned int

Output address from stage.

stage unsigned int

Stage.

aarch64_AddressSizeFault_in_table_descriptor

An AArch64 context got an AddressSizeFault as a descriptor encoded a page and the table base address described was out of range. Fields:

CPAMax unsigned int

Configured PAMax.

PAMax unsigned int

Actual PAMax.

context_bank unsigned int

context_bank.

ipa_address unsigned int

IPA of descriptor.

ns bool

Descriptor address is non-secure.

pa_address unsigned int

PA of descriptor.

stage unsigned int

Stage.

aarch64_TranslationFault_as_block_too_large

The AArch64 block described by a descriptor is too large. Fields:

context_bank unsigned int

context_bank.

level unsigned int

level.

stage unsigned int

stage.

aarch64_TranslationFault_input_address_out_of_range

An AArch64 context received an address which was outside of the programmed TxSZ. Fields:

context_bank unsigned int

context_bank.

input_address unsigned int

Input address to stage.

ns bool

Security state of context bank.

stage unsigned int

Stage.

aarch64_TranslationFault_st2_starting_level_and_tsize_inconsistent

An AArch64 context got a TranslationFault as the specified starting level and TxSZ field are inconsistent. Fields:

context_bank unsigned int

context_bank.

stage unsigned int

Stage.

add_ns_context_fault

A context fault occurred and this is the record it wishes to generate. Fields:

SMMU_CBFrsYNRAn unsigned int

SMMU_CBFrsYNRAn value that it wishes to write.

SMMU_CBn_FAR unsigned int

SMMU_CBn_FAR value that it wishes to write.

SMMU_CBn_FSR unsigned int

SMMU_CBn_FSR value that it wishes to write.

SMMU_CBn_FSYNR0 unsigned int

SMMU_CBn_FSYNR0 value that it wishes to write.

SMMU_CBn_FSYNR1 unsigned int

SMMU_CBn_FSYNR1 value that it wishes to write.

SMMU_CbN_IPAFAR unsigned int

(SMMUV2 only) SMMU_CbN_IPAFAR value that it wishes to write. The value is **UNKNOWN** if this is a stage 1 context.

context_bank unsigned int

Context bank.

going_to_be_lost bool

This fault record is going to be lost as there is already an unhandled fault record.

add_ns_global_fault

Attempt to raise a Non-Secure global fault. Fields:

SMMU_GFAR unsigned int

The fault address that we are attempting to record.

SMMU_GFSR unsigned int

The bits that we are attempting to set in the SMMU_GFSR.

SMMU_GFSYNR0 unsigned int

The syndrome we are attempting to record.

SMMU_GFSYNR1 unsigned int

The syndrome we are attempting to record.

going_to_be_lost bool

This fault record is going to be lost as there is already an unhandled fault record.

add_s_context_fault

A context fault occurred and this is the record it wishes to generate. Fields:

SMMU_CBFERSYNRAn unsigned int

SMMU_CBFERSYNRAn value that it wishes to write.

SMMU_CbN_FAR unsigned int

SMMU_CbN_FAR value that it wishes to write.

SMMU_CbN_FSR unsigned int

SMMU_CbN_FSR value that it wishes to write.

SMMU_CbN_FSYNR0 unsigned int

SMMU_CbN_FSYNR0 value that it wishes to write.

SMMU_CbN_FSYNR1 unsigned int

SMMU_CbN_FSYNR1 value that it wishes to write.

SMMU_CbN_IPAFAR unsigned int

(SMMUV2) SMMU_CbN_IPAFAR value that it wishes to read. This becomes **UNKNOWN** as this is a secure bank.

context_bank unsigned int

Context bank.

going_to_be_lost bool

This fault record is going to be lost as there is already an unhandled fault record.

add_s_global_fault

Attempt to raise a Secure global fault. Fields:

SMMU_SGFAR unsigned int

The fault address that we are attempting to record.

SMMU_SGFSR unsigned int

The bits that we are attempting to set in the SMMU_SGFSR.

SMMU_SGFSYNR0 unsigned int

The syndrome we are attempting to record.

SMMU_SGFSYNR1 unsigned int

The syndrome we are attempting to record.

going_to_be_lost bool

This fault record is going to be lost as there is already an unhandled fault record.

ats_cb_atstr_read

A context bank Address Translation Status Register read occurred (Writes are reported through `ats_cb_operation_doesnt_exist_trace`). Fields:

ACTIVE bool

The ACTIVE bit returned.

context_id unsigned int

Context bank ID.

is_debug bool

The access was a debug access.

ats_cb_operation_doesnt_exist

An attempt was made to use a Context Bank Address Translation Operation that didn't exist for some reason. Fields:

context_id unsigned int

Context bank ID.

desc string

Description of why it doesn't exist.

ats_cb_operation_queued

A new Context Bank Address Translation Operation was queued. Fields:

VA unsigned int

Virtual Address.

ats_is_privileged bool

The ATS operation is asking for a 'privileged' access.

ats_is_read bool

The ATS operation is asking for a 'read' access.

context_id unsigned int

Context bank ID.

is_debug bool

Is for a debug request.

ats_cb_operation_received_while_ats_in_progress

Another Context Bank Address Translation operation was received whilst one was currently in progress. This is **UNPREDICTABLE**. Fields:

context_id unsigned int

Context bank ID.

desc string

Description of new ATS operation.

ats_global_atstr_read

A Global Address Translation Status Register read occurred (Writes are reported through `ats_cb_operation_doesnt_exist_trace`). Fields:

ACTIVE bool

The ACTIVE bit returned.

is_debug bool

The access was a debug access.

which enum

Which global ATSR register?.

ats_global_operation_doesnt_exist

An attempt was made to use a Context Bank Address Translation Operation that didn't exist for some reason. Fields:

desc string

Description of why it doesn't exist.

is_non_secure enum

Is the operation non-secure.

ats_global_operation_queued

A new Context Bank Address Translation Operation was queued. Fields:

VA unsigned int

Virtual Address.

ats_is_privileged bool

The ATS operation is asking for a 'privileged' access.

ats_is_read bool

The ATS operation is asking for a 'read' access.

context_id unsigned int

Context bank ID.

is_debug bool

Is for a debug request.

is_non_secure enum

Is the operation non-secure.

is_s12 bool

Is a stage 1 and stage 2 translation.

ats_global_operation_received_while_ats_in_progress

Another Global Address Translation operation was received whilst one was currently in progress. This is **UNPREDICTABLE**. Fields:

context_id unsigned int

Context bank ID of new operation.

desc string

Description of new ATS operation.

is_non_secure enum

Is the operation non-secure.

ats_op_started

A queued address translation operation has started. Fields:

VA unsigned int

Input virtual address.

context_bank unsigned int

Context bank if which == 'context_bank'.

is_debug bool

The operation is a debug ATS command.

is_privileged bool

Is a privileged operation.

is_read bool

Is a read operation.

is_s12 bool

Is for a stage 1 and 2 translation.

which enum

Which group of operations?.

cfg_cttw

The cfg_cttw signal level that indicates that the SoC support coherent page walks. Fields:

value bool

The signal level.

cfg_flt_irpt

Configuration Fault Interrupt. Fields:

asserted bool

The signal is asserted.

is_non_secure bool

cfg_flt_irpt_ns or cfg_flt_irpt_s.

changed_PAGESIZE

The PAGESIZE has been told to change. Fields:

is_64KiB bool

The PAGESIZE is 64 KiB.

changing_security_state_of_context_bank

The specified bank has changed its security state. Fields:

context_bank unsigned int

Context bank number.

now_is_non_secure bool

The context bank is now non-secure.

prev_is_non_secure bool

Prior to this change in the SMMU_SCR1.NSNUMCBO then the bank was non-secure.

cmo_permission_fault

A Cache Maintenance Operation (CMO) generated a permission fault. Fields:

abort_returned_to_device bool

An abort is returned to the device.

cmo enum

Cache Maintenance Operation.

context_id unsigned int

Context ID that receives the fault.

input_address unsigned int

The address of the CMO.

cmo_permission_model_being_applied

The cache maintenance operation permission model is being applied. Fields:

aarch64 bool

Is the initial context bank aarch64?.

fault_action enum

Whether there is a fault or not.

incoming_cmo_operation enum

The CMO operation.

instruction bool

Is the CMO instruction?.

outgoing_cmo_operation enum

The CMO operation.

privileged bool

Is the CMO privileged?.

stage1_perms enum

The permissions at stage 1.

stage2_perms enum

The permissions at stage 1.

comb_irpt

Combined interrupt signal. This is the OR of all of the non-secure interrupt signals. Fields:

asserted bool

The signal is asserted.

is_non_secure bool

Is the comb_irpt_ns signal or the comb_irpt_s signal.

configuration_access_fault_unimplemented_stream_mapping_register_group

This occurs when a configuration transaction attempts to access a stream mapping register group that doesn't exist and if the implementation treats this as a Configuration Access Fault. Fields:

address unsigned int

Address that was used.

group_number unsigned int

Group number if it were to exist/be accessible.

is_non_secure bool

Is the configuration access non-secure?.

is_read bool

Is the configuration access a read?.

context_bank_commentary

These messages are verbose commentary about what the context bank is doing. Fields:

output string

The stream output.

context_bank_mmu_off_stage1

This is the transformation applied by the stage 1 SMMU_CBn_SCTLR. Fields:

SMMU_CBn_SCTLR unsigned int

Value of the SMMU_CBn_SCTLR used.

context_id unsigned int

Context bank id the transaction has been mapped to.

trans_id unsigned int

Transaction ID.

context_bank_mmu_off_stage2

This is the transformation applied by the stage 2 SMMU_CbN_SCTLR. Fields:

SMMU_CbN_SCTLR unsigned int

Value of the SMMU_CbN_SCTLR used.

context_id unsigned int

Context bank id the transaction has been mapped to.

trans_id unsigned int

Transaction ID.

ctx_irpt_ns

Context Interrupt, this can only ever be non-secure. Fields:

asserted bool

The signal is asserted.

debug

This is the textual output of the debug messages of the SMMU model, this need not necessarily make much sense to the user, but is useful for sending to the model writers when reporting bugs. Fields:

output string

The stream output.

denied_access_due_to_imp_def_reason

The register access was denied because of some **IMPLEMENTATION DEFINED** reason due to the transaction attributes. Fields:

address unsigned int

Address of register access.

desc string

Description of register.

denied_access_due_to_smmu_scr1_gasrae

The register access was denied because SMMU_SCR1.GASRAE has locked down the register. Fields:

address unsigned int

Address of register access.

dvm_tlbinvalidate_complete

The DVM TLB Invalidate message completed.

dvm_tlbinvalidate_received

A DVM message for a TLB Invalidate has been received. Fields:

address unsigned int

The VA or IPA to use if match_address.

asid unsigned int

The ASID to match if match_asid.

by_ipa bool

The operation is for an IPA operation if match_address.

ignored enum

The DVM message was ignored.

last_level bool

The operation is for last level if supported.

match_address bool

Match the address field.

match_asid bool

Match the asid field.

match_vmid bool

Match the vmid field.

prot enum

The protection level for which this TLB Invalidate will operate on.

security_world enum

The security world that this will apply to.

stage1_only bool

The operation is for stage 1 only if supported.

vmid unsigned int

The VMID to match if match_vmid.

end_stall_transaction

Indicates a transaction unstalling. Fields:

context_id unsigned int

The context id that we are stalling against.

trans_id unsigned int

Transaction id of replaying transaction.

unstall_state enum

Describes why we unstalled. If we are head of line then we are either replay/terminate, otherwise we will be not be.

error

These messages are about activity occurring on the SMMU that is considered an error. Fields:

output string

The stream output.

error_contiguous_bit_is_bad

The contiguous bit is inconsistent with the descriptor. Fields:

B_mask unsigned int

The mask of the address that is zapped if we were to use the contiguous bit.

aarch64_contig_bit_too_high bool

AArch64 claims that the contig bit can't be set for this particular set of start levels and input address size.

input_address unsigned int

The input address that will be combined with the output address.

ln2_size_in_bytes_of_region_if_contiguous unsigned int

The number of address bits that form the 'offset' within the contiguous region.

out_of_range_of_input bool

The range mapped by a contiguous bit being on is larger than the input range for this stage.

output_address unsigned int

The output address as contained in the descriptor.

output_address_if_contig_bit_uses_input_address unsigned int

Output address if contig bit was fully taken (maximally sized region) and those bits came from the input address.

output_address_inconsistent bool

The output address in the descriptor is inconsistent with its position in the table.

stage_and_level unsigned int

Top 8 bits are the stage, bottom 8 bits are level.

error_invalid_smmu_cbar

A SMMU_CBAR has been encountered for a secure stream that is not of the Stage 1 with Stage 2 bypass form. Fields:

SMMU_CBARn unsigned int

The corresponding SMMU_CBARn for the context.

action enum

Action taken.

context_id unsigned int

Context ID.

trans_id unsigned int

Transaction id.

error_tlb_entries_overlap

A TLB entry was attempted to be inserted into the TLB and was determined that it overlaps an existing entry. This check is not perfect but will catch simple errors. Fields:

end_address_of_new_entry unsigned int

End address of new entry.

end_address_of_old_entry unsigned int

End address of old entry.

index_of_new_entry unsigned int

Index of new entry.

index_of_old_entry unsigned int

Index of old entry.

start_address_of_new_entry unsigned int

Start address of new entry.

start_address_of_old_entry unsigned int

Start address of old entry.

extended_attributes_for_stage

This gives the attributes that a particular stage gives prior to being applied to the incoming attributes. Fields:

inner enum

Inner cacheability.

ns bool

NS/S security state.

outer enum

Outer cacheability.

racfg enum

RACFG from descriptor (if exists).

shareability enum

Shareability.

stage unsigned int

Stage that this record is for.

trans_id unsigned int

Transaction ID this is for.

wacfg enum

WACFG from descriptor (if exists).

glblflt_irpt

Global Fault Interrupt. Fields:

asserted bool

The signal is asserted.

is_non_secure bool

g!blflt_irpt_ns or g!blflt_irpt_s.

imp_def_non_SMMU_CbN_RESUME_register_access_transaction_forces_replay

There is an IMP DEF option to replay stalled transactions in affected banks. This records that the implementation has made use of this option. Fields:

context_id unsigned int

Context ID of affected bank.

why enum

Why did this replay occur?.

implementation_postprocess

This describes the non-architectural transformation that the implementation makes. Fields:

end_adomain enum

End adomain value.

end_inner_acache enum

End inner acache value.

end_outer_acache enum

End outer acache value.

is_read bool

Is a read transaction.

start_adomain enum

Start adomain value.

start_inner_acache enum

Start inner acache value.

start_outer_acache enum

Start outer acache value.

info

These are information messages about what is happening in the SMMU. Fields:

output string

The stream output.

invalid_address_range

An input or output address range check failed. Input address range checks are reported as TranslationFault, output ones as AddressSizeFault. Fields:

context_bank unsigned int

Context bank.

offending_address unsigned int

The address that failed the address check.

why enum

The reason why the address was invalid.

ld_access_fault

An Access Fault occurred. Fields:

context_bank unsigned int

Context bank.

input_address unsigned int

The input address used for this descriptor.

is_non_secure bool

Is the input address non-secure?.

level unsigned int

Level of walk.

stage unsigned int

Stage of walk.

ld_descriptor_fetch_failed

Long Descriptor AArch32 or AArch64, level 'n' descriptor fetch failed. Fields:

context_bank unsigned int

Context bank.

ipa_address unsigned int

IPA of the descriptor address.

is_non_secure bool

Is the page walk and input addresses non-secure?.

level unsigned int

Level of walk.

pa_address unsigned int

PA of the descriptor address.

stage unsigned int

Stage of walk.

ld_descriptor_invalid

Long Descriptor AArch32 or AArch64 encodes an invalid entry. Fields:

context_bank unsigned int

Context bank.

ipa_address unsigned int

IPA of the descriptor address.

level unsigned int

Level of walk.

pa_address unsigned int

PA of the descriptor address.

ssd_ns bool

Is the transaction classified as non-secure.

stage unsigned int

Stage of walk.

ld_hyp_mode_address_out_of_range

A hypervisor input address is out of range of TOSZ. Fields:

context_bank unsigned int

The context bank.

input_address unsigned int

The VA input address used for this descriptor.

is_non_secure bool

Is this for the Non-secure side?.

ld_ipa_out_of_range

A stage 2 context bank was given an address that didn't match TOSZ. Fields:

context_bank unsigned int

The context bank.

input_address_ipa unsigned int

The IPA input address used for this descriptor.

is_non_secure bool

Is this for the Non-secure side?.

ld_ipa_out_of_range_for_stlptw_st2_mmu_off

(SMMUv2) The stage 2 context (nested with MMU off or bypass) for a stage 1 translation had an address that was out of the allowed range. Fields:

context_bank unsigned int

The context bank.

input_address_ipa unsigned int

The IPA input address used for this descriptor.

is_non_secure bool

Is this for the Non-secure side?.

ld_last_level_descriptor_does_not_encode_a_page

Long Descriptor, last level descriptor must encode a page but didn't. Fields:

context_bank unsigned int

Context bank.

ipa_address unsigned int

IPA of the descriptor address.

level unsigned int

Level of walk.

pa_address unsigned int

PA of the descriptor address.

ssd_ns bool

Is the transaction classified as non-secure?.

stage unsigned int

Stage of walk.

ld_misprogramming_fault_trace

A misprogramming occurred that the implementation faulted. Fields:

context_bank unsigned int

Context bank.

input_address unsigned int

Input address.

stage unsigned int

Stage.

why enum

Why the fault occurred.

ld_st2_produced_any_device_memory_for_st1ptw

A stage 2 translation process for a stage 1 page walk address produce a any-device memory type. Fields:

context_bank unsigned int

Context bank.

input_address unsigned int

The input address used for this descriptor.

level unsigned int

Level of walk.

ld_st2_slptw_has_no_access_rights

A stage 2 translation process for a stage 1 page walk address produce a permission fault. Fields:

context_bank unsigned int

Context bank.

input_address unsigned int

The input address used for this descriptor.

level unsigned int

Level of walk.

ld_stage1_non_hyp_no_ttbr_match

A stage 1 non-hyp entry failed to match TTBR0 or TTBR1. Fields:

context_bank unsigned int

The context bank.

input_address unsigned int

The input address used.

is_non_secure bool

Is this for the Non-secure side?.

ld_walk_process_is_disabled

A L1 or L2 short descriptor access flag has disabled access for this (super-)section/page. Fields:

context_bank unsigned int

The context bank.

input_address unsigned int

The input address used.

is_non_secure bool

Is this for the Non-secure side?.

stage unsigned int

Which change this corresponds to.

ttbr_id unsigned int

Which TTBR we are using (is 0 for a stage 2).

non_PTW_external_fault_recorded

A non Page Table Walk access was recorded when a translated transaction went downstream, but the downstream system aborted it. This is recorded as an asynchronous fault. In the model, we can provide more information about the faulting transaction. Fields:

downstream_address unsigned int

The downstream address.

downstream_ns enum

The downstream NS state. This is not the SSD.

is_read enum

Is the transaction a read?.

result_reported_upstream enum

The result that is reported upstream.

sGFSYNR0 unsigned int

The value of sGFSYNR0 that will attempt to be recorded (all 1s mean it will not be updated).

sGFSYNR1 unsigned int

The value of sGFSYNR1 that will attempt to be recorded (all 1s mean it will not be updated).

smmu_scr1_gefro bool

The value of SMMU_SCR1.GEFRO.

smmu_xcr0_gfre bool

The appropriate SMMU_sCR0.GFRE bit. If this is 0 then it squash the downstream abort.

ssd_ns enum

SSD.

upstream_address unsigned int

The upstream address.

upstream_ns enum

The upstream NS state. This is not the SSD.

post_s2crn_attributes

The attributes after the SMMU_S2CRn transformation has been applied. Fields:

adomain enum

The actual attributes of the access that was used, after IMP DEF mangling.

aprot enum

The actual attributes of the access that was used, after IMP DEF mangling.

inner_cache enum

The actual attributes of the access that was used, after IMP DEF mangling.

inner_transient bool

Is the transaction marked with the transient hint?.

outer_cache enum

The actual attributes of the access that was used, after IMP DEF mangling.

outer_transient bool

Is the transaction marked with the transient hint?.

trans_id unsigned int

Transaction id.

v8_dev_gathering bool

Is the transaction marked with the V8 Device Gathering property?.

v8_dev_reordering bool

Is the transaction marked with the V8 Device Reordering property?.

ptw_read

Page Table Walk (read). This is the physical access that the SMMU is making. Fields:

adomain enum

The actual attributes of the access that was used, after IMP DEF mangling.

aprot enum

The actual attributes of the access that was used, after IMP DEF mangling.

auser unsigned int

The AUSER fields.

data unsigned int

The data read in little-endian format.

hyp bool

Current mode is hyp.

inner_cache enum

The actual attributes of the access that was used, after IMP DEF mangling.

input_address unsigned int

Input address this lookup is for.

ns bool

This is for SSD Non-secure.

outer_cache enum

The actual attributes of the access that was used, after IMP DEF mangling.

pa_address unsigned int

Physical address.

result enum

The PTW result.

st2_st1_context_banks unsigned int

The stage 1 and 2 context bank that this walk is for (if there is one, otherwise 0xFF).
Stage 1 is the lowest byte, stage 2 the highest.

stage_and_level unsigned int

Stage and level that this walk is for. Top byte is stage, bottom is level.

trans_id unsigned int

Transaction id.

ptw_read_st1_invalid_long_descriptor

Page Table Walk (read). The descriptor that the PTW fetched is invalid. Fields:

hyp bool

Current mode is hyp.

input_address unsigned int

Input address this lookup is for.

level unsigned int

The page table level this walk is for.

ns bool

This is for SSD Non-secure.

pa_address unsigned int

Physical address.

st1_context_bank unsigned int

The stage 1 context bank that this walk is for (if there is one, otherwise 0xFF).

st2_context_bank unsigned int

The stage 2 context bank that this walk is for (if there is one, otherwise 0xFF).

trans_id unsigned int

Transaction id.

ptw_read_st1_leaf_long_descriptor

Page Table Walk (read). The descriptor that the PTW fetched is a block or page and this is the decode. Fields:

AF bool

Access Flag.

AP21 enum

The access permissions.

AttrIdx210 unsigned int

The attribute index into the MAIR0/1.

NS bool

The encoding is for non-secure if this is a secure fetch.

PXN bool

Privileged eXecute Never.

SH10 enum

The shareability.

XN bool

eXecute Never.

contiguous bool

Contiguous hint.

hyp bool

Current mode is hyp.

input_address unsigned int

Input address this lookup is for.

level unsigned int

The page table level this walk is for.

mair unsigned int

The MAIRn encoding attribute.

nG bool

not Global.

ns bool

The agent is Non-Secure.

output_address unsigned int

Output address.

pa_address unsigned int

Physical address.

st1_context_bank unsigned int

The stage 1 context bank that this walk is for (if there is one, otherwise 0xFF).

st2_context_bank unsigned int

The stage 2 context bank that this walk is for (if there is one, otherwise 0xFF).

trans_id unsigned int

Transaction id.

ptw_read_st1_table_long_descriptor

Page Table Walk (read). The descriptor that the PTW fetched is a Table, this decodes the fields. Fields:

APTable enum

Remove permissions independently of subsequent descriptors.

NSTable bool

The next level table descriptor is forced to non-secure.

PXNTable bool

Force PXN independently of subsequent descriptors.

TableAddress unsigned int

Address of the next table.

XNTable bool

Force XN independently of subsequent descriptors.

hyp bool

Current mode is hyp.

input_address unsigned int

Input address this lookup is for.

level unsigned int

The page table level this walk is for.

ns bool

This is for SSD Non-secure.

pa_address unsigned int

Physical address.

st1_context_bank unsigned int

The stage 1 context bank that this walk is for (if there is one, otherwise 0xFF).

st2_context_bank unsigned int

The stage 2 context bank that this walk is for (if there is one, otherwise 0xFF).

trans_id unsigned int

Transaction id.

ptw_read_st1_v7s_fault_descriptor

Page Table Walk (read). The descriptor encodes a fault descriptor. Fields:

level unsigned int

The level of the page table.

trans_id unsigned int

Transaction id.

ptw_read_st1_v7s_large_page_descriptor

Page Table Walk (read). The descriptor encodes a large page descriptor. Fields:

AP210 unsigned int

AP[2:0].

S bool

Shareable.

TEX210CB unsigned int

{TEX[2:0],C,B}.

XN bool

eXecute Never.

large_page_base_address unsigned int

Base address of the large page.

nG bool

not Global.

trans_id unsigned int

Transaction id.

ptw_read_st1_v7s_page_table_descriptor

Page Table Walk (read). The descriptor encodes a page table descriptor pointing to an L2 table. Fields:

NS bool

Encodes a non-secure entry.

PXN bool

Privileged eXecute Never.

domain unsigned int

Domain.

l2_table_address unsigned int

L2 page table address.

trans_id unsigned int

Transaction id.

ptw_read_st1_v7s_section_descriptor

Page Table Walk (read). The descriptor encodes a section descriptor. Fields:

AP210 unsigned int

AP[2:0].

NS bool

Encodes a non-secure entry.

PXN bool

Privileged eXecute Never.

S bool

Shareable.

TEX210CB unsigned int

{TEX[2:0],C,B}.

XN bool

eXecute Never.

domain unsigned int

Domain.

nG bool

not Global.

section_base_address unsigned int

Base address of section.

trans_id unsigned int

Transaction id.

ptw_read_st1_v7s_small_page_descriptor

Page Table Walk (read). The descriptor encodes a small page descriptor. Fields:

AP210 unsigned int

AP[2:0].

S bool

Shareable.

TEX210CB unsigned int

{TEX[2:0],C,B}.

XN bool

eXecute Never.

nG bool

not Global.

small_page_base_address unsigned int

Base address of the small page.

trans_id unsigned int

Transaction id.

ptw_read_st1_v7s_super_section_descriptor

Page Table Walk (read). The descriptor encodes a section descriptor. Fields:

AP210 unsigned int

AP[2:0].

NS bool

Encodes a non-secure entry.

PXN bool

Privileged eXecute Never.

S bool

Shareable.

TEX210CB unsigned int

{TEX[2:0],C,B}.

XN bool

eXecute Never.

nG bool

not Global.

super_section_base_address unsigned int

Base address of the super section.

trans_id unsigned int

Transaction id.

ptw_read_st2_invalid_descriptor

Page Table Walk (read). The descriptor that the PTW fetched is invalid. Fields:

hyp bool

Current mode is hyp.

input_address unsigned int

Input address this lookup is for.

level unsigned int

The page table level this walk is for.

ns bool

This is for SSD Non-secure.

pa_address unsigned int

Physical address.

st1_context_bank unsigned int

The stage 1 context bank that this walk is for (if there is one, otherwise 0xFF).

st2_context_bank unsigned int

The stage 2 context bank that this walk is for (if there is one, otherwise 0xFF).

trans_id unsigned int

Transaction id.

ptw_read_st2_leaf_descriptor

Page Table Walk (read). The descriptor that the PTW fetched is a block or page and this is the decode. Fields:

AF bool

Access Flag.

HAP21 enum

The access permissions.

MemAttr3_0 enum

The memory attributes.

RACFG enum

The extra SMMU RACFG field.

SH10 enum

The shareability.

WACFG enum

The extra SMMU WACFG field.

XN bool

eXecute Never.

contiguous bool

Contiguous hint.

hyp bool

Current mode is hyp.

input_address unsigned int

Input address this lookup is for.

level unsigned int

The page table level this walk is for.

ns bool

This is for SSD Non-secure.

output_address unsigned int

Output address.

pa_address unsigned int

Physical address.

st1_context_bank unsigned int

The stage 1 context bank that this walk is for (if there is one, otherwise 0xFF).

st2_context_bank unsigned int

The stage 2 context bank that this walk is for (if there is one, otherwise 0xFF).

trans_id unsigned int

Transaction id.

ptw_read_st2_table_descriptor

Page Table Walk (read). The descriptor that the PTW fetched is a Table, this decodes the fields. Fields:

APTable enum

Remove permissions independently of subsequent descriptors.

NSTable bool

The next level table descriptor is forced to non-secure.

PXNTable bool

Force PXN independently of subsequent descriptors.

TableAddress unsigned int

Address of the next table.

XNTable bool

Force XN independently of subsequent descriptors.

hyp bool

Current mode is hyp.

input_address unsigned int

Input address this lookup is for.

level unsigned int

The page table level this walk is for.

ns bool

This is for SSD Non-secure.

pa_address unsigned int

Physical address.

st1_context_bank unsigned int

The stage 1 context bank that this walk is for (if there is one, otherwise 0xFF).

st2_context_bank unsigned int

The stage 2 context bank that this walk is for (if there is one, otherwise 0xFF).

trans_id unsigned int

Transaction id.

raw_want_to_change_context_interrupt

This represents a context's bank intent to want to change an interrupt pin. However, this doesn't take into account the accessibility of the bank, nor if other banks are driving the pin.

Fields:

accessible bool

Is the interrupt ID accessible to this context bank?.

asserted bool

Is this context bank attempting to assert the interrupt?.

context_bank unsigned int

Context bank attempting to change the interrupt pin.

interrupt_id unsigned int

The interrupt ID it is attempting to change.

register_disallowed_read_string

A text representation of the read of a register that was disallowed. Fields:

out string

The text description of the register value read.

register_disallowed_write_string

A text representation of the write of a register write that was disallowed. Fields:

in string

The text description of the register value written.

register_read

Read of the register file. Fields:

address unsigned int

The address of the register.

context_id unsigned int

Context id bank (or 0xFF if not a context bank).

ln2_width enum

The log2 width of the access.

ns enum

The security state of the transaction addressing the register.

register_trans_id unsigned int

Register transaction ID (if top bit set then a debug transaction).

result enum

The result of the access.

value unsigned int

The byte-stream of all the data (little-endian).

register_read_reserved

A text representation of an access to a register address that is reserved. Fields:

in string

The text description of the register value.

register_read_string

A text representation of the read of a register. Fields:

out string

The text description of the register value read.

register_write

Write to the register file. Fields:

address unsigned int

The address of the register.

context_id unsigned int

Context id bank (or 0xFF if not a context bank).

ln2_width enum

The log2 width of the access.

ns enum

The security state of the transaction addressing the register.

register_trans_id unsigned int

Register transaction ID (if top bit set then a debug transaction).

result enum

The result of the access.

value unsigned int

The byte-stream of all the data (little-endian).

register_write_reserved

A text representation of an access to a register address that is reserved. Fields:

in string

The text description of the register value.

register_write_string

A text representation of the write of a register. Fields:

in string

The text description of the register value written.

reset_sequence

Marks the beginning and end of the reset sequence. Fields:

starting bool

True if starting the reset sequence, false when finished.

signal_reset

The reset signal level. Fields:

value bool

The signal level.

smmu_faulting_transaction

This is a trace of transaction that are not expected to fault. This does not include those transaction that are faulting by stage 2 or where the access flag or permission model is not allowing the transaction. DISPLAY trans_id:%{trans_id}-tbu:%{tbu} i %{original_attributes[3]}(w|W)%{original_attributes[2]}(r|R)%{original_attributes[1]}(m|M)%{original_attributes[0]}(b|B)%{original_attributes[14:13]}(\nGRE|?)_GRE)%{original_attributes[15]}(_T)}-o%{original_attributes[7]}(w|W)%{original_attributes[6]}(r|R)%{original_attributes[5]}(m|M)%{original_attributes[4]}(b|B)%{original_attributes[14:13]}(\nGRE|?)_GRE)%{original_attributes[16]}(_T)}-%{original_attributes[9:8]}(nsh|ish|osh|sys)}-%{original_attributes[10]}(p|P)%{original_attributes[11]}(S|N)%{original_attributes[12]}(D|I)%{original_attributes[17]}(|-excl)}%{original_attributes[18]}(|-debug)}%{original_attributes[19]}(|-cmo)}%{original_attributes[21.24]}(|-translated|-nonstallable|-transfaultflow)}%{original_attributes[22]}(|-non-address-based)}%{original_attributes[25]}(|-has-meta-data)}%{original_attributes[26]}(|-reserved-imp-def)}%{original_attributes[27]}(|-stream)}%{original_attributes[28]}(|-nse)}%{original_attributes[29]}(|-nse2)}%{original_attributes[31]}(|-has-mutating-meta-data-op)} -> i%{mapped_attributes[3]}(w|W)%{mapped_attributes[2]}(r|R)%{mapped_attributes[1]}(m|M)%{mapped_attributes[0]}(b|B)%{mapped_attributes[14:13]}(\nGRE|?)_GRE)%{mapped_attributes[15]}(_T)}-o%{mapped_attributes[7]}(w|W)%{mapped_attributes[6]}(r|R)%{mapped_attributes[5]}(m|M)%{mapped_attributes[4]}(b|B)%{mapped_attributes[14:13]}(\nGRE|?)_GRE)%{mapped_attributes[16]}(_T)}-%{mapped_attributes[9:8]}(nsh|ish|osh|sys)}-%{mapped_attributes[10]}(p|P)%{mapped_attributes[11]}(S|N)%{mapped_attributes[12]}(D|I)%{mapped_attributes[17]}(|-excl)}%{mapped_attributes[18]}(|-debug)}%{mapped_attributes[19]}(|-cmo)}%{mapped_attributes[21.24]}(|-translated|-nonstallable|-transfaultflow)}%{mapped_attributes[22]}(|-non-address-based)}%{mapped_attributes[25]}(|-has-meta-data)}%{mapped_attributes[26]}(|-reserved-imp-def)}%{mapped_attributes[27]}(|-stream)}%{mapped_attributes[28]}(|-nse)}%{mapped_attributes[29]}(|-nse2)}%{mapped_attributes[31]}(|-has-mutating-meta-data-op)}%{is_read:(W|R)}-stream_id:%{stream_id}-ssd_index:%{ssd_index}-ssd:%{ssd:(s-|ns-)}%{input_address:-> %}{output_address}. PRIMARY KEY address. Fields:

context_id unsigned int

The context id of the transaction used, if we failed to immediately hit in the TLB.

input_address unsigned int

The input address of the transaction group.

mapped_attributes unsigned int

The mapped attributes of the transaction.

original_attributes unsigned int

The original unmapped attributes of the transaction.

output_address unsigned int

The output address of the transaction group.

ssd_index unsigned int

The Security State Determination Index if applicable.

ssd_ns enum

The Security State Determination of the transaction.

state enum

The state of the transaction.

stream_id unsigned int

The stream id of the transaction.

tbu unsigned int

Translation Buffer Unit number.

tlb_entry_index_st1 unsigned int

If this was mapped by a TLB entry index then this is the one used for the stage 1 translation.

tlb_entry_index_st2 unsigned int

If this was mapped by a TLB entry index then this is the one used for the stage 2 translation.

trans_id unsigned int

Transaction id.

type_of_transaction enum

The type of transaction.

smmu_final_transaction

This is a trace of the SMMU transaction after it has been fully remapped. DISPLAY
 trans_id:%{trans_id}-tbu:%{tbu} i%{original_attributes[3]}(w|W)}%{original_attributes[2]}(r|R)}%{original_attributes[1]}(m|M)}%{original_attributes[0]}(b|B)}%{original_attributes[14:13]}(nGRE|?_GRE)}%{original_attributes[15]}(|_T)}-o%{original_attributes[7]}(w|W)}%{original_attributes[6]}(r|R)}%{original_attributes[5]}(m|M)}%{original_attributes[4]}(b|B)}%{original_attributes[14:13]}(nGRE|?_GRE)}%{original_attributes[16]}(|_T)}-%{original_attributes[9:8]}(nsh|ish|osh|sys)}-%{original_attributes[10]}(p|P)}%{original_attributes[11]}(S|N)}%{original_attributes[12]}(D|I)}%{original_attributes[17]}(|-excl)}%{original_attributes[18]}(|-debug)}%{original_attributes[19]}(|-cmo)}%{original_attributes[21:24]}(|-translated|-nonstallable|-transfaultflow)}%{original_attributes[22]}(|-non-address-based)}%{original_attributes[25]}(|-has-meta-data)}%{original_attributes[26]}(|-reserved-imp-def)}%{original_attributes[27]}(|-stream)}%{original_attributes[28]}(|-nse)}%{original_attributes[29]}(|-

```
nse2}}%{original_attributes[31]:(|-has-mutating-meta-data-op}} -> i%{mapped_attributes[3]:
(w|W}}%{mapped_attributes[2]:(r|R)}}%{mapped_attributes[1]:(m|M)}}%{mapped_attributes[0]:
(b|B)}}%{mapped_attributes[14:13]:(|nGRE|?|_GRE)}}%{mapped_attributes[15]:(|_T)}}-o
%{mapped_attributes[7]:(w|W)}}%{mapped_attributes[6]:(r|R)}}%{mapped_attributes[5]:
(m|M)}}%{mapped_attributes[4]:(b|B)}}%{mapped_attributes[14:13]:(|nGRE|?|
_GRE)}}%{mapped_attributes[16]:(|_T)}}-%{mapped_attributes[9:8]:(nsh|
ish|osh|sys)}}-%{mapped_attributes[10]:(p|P)}}%{mapped_attributes[11]:(S|
N)}}%{mapped_attributes[12]:(D|I)}}%{mapped_attributes[17]:(|-excl)}}%{mapped_attributes[18]:
(|-debug)}}%{mapped_attributes[19]:(|-cmo)}}%{mapped_attributes[21.24]:(|-
translated|-nonstallable|-transfaultflow)}}%{mapped_attributes[22]:(|-non-address-
based)}}%{mapped_attributes[25]:(|-has-meta-data)}}%{mapped_attributes[26]:(|-
reserved-imp-def)}}%{mapped_attributes[27]:(|-stream)}}%{mapped_attributes[28]:(|-
nse)}}%{mapped_attributes[29]:(|-nse2)}}%{mapped_attributes[31]:(|-has-mutating-meta-
data-op)}}%{is_read:(W:|R))-stream_id:%{stream_id}-ssd_index:%{ssd_index}-ssd:%{ssd: ( s-|
ns-)}}%{input_address:->%{output_address}. PRIMARY KEY address. Fields:
```

context_id unsigned int

The context id of the transaction used, if we failed to immediately hit in the TLB.

input_address unsigned int

The input address of the transaction group.

mapped_attributes unsigned int

The mapped attributes of the transaction.

original_attributes unsigned int

The original unmapped attributes of the transaction.

output_address unsigned int

The output address of the transaction group.

ssd_index unsigned int

The Security State Determination Index if applicable.

ssd_ns enum

The Security State Determination of the transaction.

state enum

The state of the transaction.

stream_id unsigned int

The stream id of the transaction.

tbu unsigned int

Translation Buffer Unit number.

tlb_entry_index_st1 unsigned int

If this was mapped by a TLB entry index then this is the one used for the stage 1 translation.

tlb_entry_index_st2 unsigned int

If this was mapped by a TLB entry index then this is the one used for the stage 2 translation.

trans_id unsigned int

Transaction id.

type_of_transaction enum

The type of transaction.

smmu_initial_transaction

This is the initial transaction group request to remap. This represents a bundle of transactions with the same attributes but different addresses within a certain range around the address. DISPLAY trans_id:%{trans_id}-tbu:%{tbu} i %{original_attributes[3]}(w|W)}%{original_attributes[2]}(r|R)}%{original_attributes[1]}(m|M)}%{original_attributes[0]}(b|B)}%{original_attributes[14:13]}(nGRE|?|_GRE)}%{original_attributes[15]}(|_T)}-o%{original_attributes[7]}(w|W)}%{original_attributes[6]}(r|R)}%{original_attributes[5]}(m|M)}%{original_attributes[4]}(b|B)}%{original_attributes[14:13]}(nGRE|?|_GRE)}%{original_attributes[16]}(|_T)}-%{original_attributes[9:8]}(nsh|ish|osh|sys)}-%{original_attributes[10]}(p|P)}%{original_attributes[11]}(S|N)}%{original_attributes[12]}(D|I)}%{original_attributes[17]}(|-excl)}%{original_attributes[18]}(|-debug)}%{original_attributes[19]}(|-cmo)}%{original_attributes[21.24]}(|-translated|-nonstallable|-transfaultflow)}%{original_attributes[22]}(|-non-address-based)}%{original_attributes[25]}(|-has-meta-data)}%{original_attributes[26]}(|-reserved-imp-def)}%{original_attributes[27]}(|-stream)}%{original_attributes[28]}(|-nse)}%{original_attributes[29]}(|-nse2)}%{original_attributes[31]}(|-has-mutating-meta-data-op)}%{is_read:(W|R)}-stream_id:%{stream_id}-ssd_index:%{ssd_index}-ssd:%{ssd} (s-|ns-)}%{input_address:}. PRIMARY KEY input_address. Fields:

input_address unsigned int

The input address of the transaction group.

kind_addressed enum

The kind of transaction this represents if it is an addressed transaction.

original_attributes unsigned int

The original unmapped attributes of the transaction.

ssd_index unsigned int

The Security State Determination Index if applicable.

ssd_ns enum

The Security State Determination of the transaction.

stream_id unsigned int

The stream id of the transaction.

tbu unsigned int

Translation Buffer Unit number.

trans_id unsigned int

Transaction id.

type_of_transaction enum

The type of transaction.

smmu_iprt_xgirpt

This trace source records the state of the SMMU_glrpt/SMMU_NSglrpt pins. Fields:

asserted bool

The interrupt is now asserted.

is_non_secure bool

Identifies the pin as SMMU_NSglrpt.

smmu_irpt_combined_context_interrupt_pin

This trace source records when there is at least one context bank in the specific world with an interrupt pin raised. Fields:

asserted bool

The interrupt is now asserted.

ns bool

Non-secure.

smmu_irpt_context_interrupt_pin

This trace source records when a context bank interrupt pin changes state. Fields:

asserted bool

The interrupt is now asserted.

interrupt_pin unsigned int

Interrupt pin number.

smmu_irpt_context_interrupt_pin_drivers

This trace source records when a context bank tries to assert an interrupt. Multiple banks can drive the same pin and so long as one is driving it then it is asserted. Fields:

context_bank unsigned int

The context bank number changing its request for the pin.

interrupt_pin unsigned int

Interrupt pin number.

new_count_asserting_pin unsigned int

The number of banks asserting the pin.

this_context_asserting bool

The request that this context is making – if this is true then the context bank is asserting a request.

smmu_irpt_xgcfglrpt

This trace source records the state of the SMMU_gCfgrpt/SMMU_NSgCfgrpt pins. Fields:

asserted bool

The interrupt is now asserted.

is_non_secure bool

Identifies the pin as SMMU_NSgCfgrpt.

smmu_st1_st2fault_transaction

This is a trace of transactions that map to stage 1 with stage 2 faulting. DISPLAY
trans_id: %{trans_id}-tbu: %{tbu} i: %{original_attributes[3]:(w|W)}%{original_attributes[2]:(r|R)}%{original_attributes[1]:(m|M)}%{original_attributes[0]:(b|B)}%{original_attributes[14:13]:(nGRE|?|_GRE)}%{original_attributes[15]:(|_T)}-o: %{original_attributes[7]:(w|W)}%{original_attributes[6]:(r|R)}%{original_attributes[5]:(m|M)}%{original_attributes[4]:(b|B)}%{original_attributes[14:13]:(nGRE|?|_GRE)}%{original_attributes[16]:(|_T)}-%{original_attributes[9:8]:(nsh|ish|osh|sys)}-%{original_attributes[10]:(p|P)}%{original_attributes[11]:(S|N)}%{original_attributes[12]:(D|I)}%{original_attributes[17]:(|-excl)}%{original_attributes[18]:(|-debug)}%{original_attributes[19]:(|-cmo)}%{original_attributes[21.24]:(|-translated|-nonstallable|-transfaultflow)}%{original_attributes[22]:(|-non-address-based)}%{original_attributes[25]:(|-has-meta-data)}%{original_attributes[26]:(|-reserved-imp-def)}%{original_attributes[27]:(|-stream)}%{original_attributes[28]:(|-nse)}%{original_attributes[29]:(|-nse2)}%{original_attributes[31]:(|-has-mutating-meta-data-op)} -> i: %{mapped_attributes[3]:(w|W)}%{mapped_attributes[2]:(r|R)}%{mapped_attributes[1]:(m|M)}%{mapped_attributes[0]:(b|B)}%{mapped_attributes[14:13]:(nGRE|?|_GRE)}%{mapped_attributes[15]:(|_T)}-o: %{mapped_attributes[7]:(w|W)}%{mapped_attributes[6]:(r|R)}%{mapped_attributes[5]:(m|M)}%{mapped_attributes[4]:(b|B)}%{mapped_attributes[14:13]:(nGRE|?|_GRE)}%{mapped_attributes[16]:(|_T)}-%{mapped_attributes[9:8]:(nsh|ish|osh|sys)}-%{mapped_attributes[10]:(p|P)}%{mapped_attributes[11]:(S|N)}%{mapped_attributes[12]:(D|I)}%{mapped_attributes[17]:(|-excl)}%{mapped_attributes[18]:(|-debug)}%{mapped_attributes[19]:(|-cmo)}%{mapped_attributes[21.24]:(|-translated|-nonstallable|-transfaultflow)}%{mapped_attributes[22]:(|-non-address-based)}%{mapped_attributes[25]:(|-has-meta-data)}%{mapped_attributes[26]:(|-reserved-imp-def)}%{mapped_attributes[27]:(|-stream)}%{mapped_attributes[28]:(|-nse)}%{mapped_attributes[29]:(|-nse2)}%{mapped_attributes[31]:(|-has-mutating-meta-data-op)}%{is_read:(W:|R)}-stream_id: %{stream_id}-ssd_index: %{ssd_index}-ssd: %{ssd: (s|ns-)}%{input_address:}->%{output_address}. PRIMARY KEY address. Fields:

context_id unsigned int

The context id of the transaction used, if we failed to immediately hit in the TLB.

input_address unsigned int

The input address of the transaction group.

mapped_attributes unsigned int

The mapped attributes of the transaction.

original_attributes unsigned int

The original unmapped attributes of the transaction.

output_address unsigned int

The output address of the transaction group.

ssd_index unsigned int

The Security State Determination Index if applicable.

ssd_ns enum

The Security State Determination of the transaction.

state enum

The state of the transaction.

stream_id unsigned int

The stream id of the transaction.

tbu unsigned int

Translation Buffer Unit number.

tlb_entry_index_st1 unsigned int

If this was mapped by a TLB entry index then this is the one used for the stage 1 translation.

tlb_entry_index_st2 unsigned int

If this was mapped by a TLB entry index then this is the one used for the stage 2 translation.

trans_id unsigned int

Transaction id.

type_of_transaction enum

The type of transaction.

smrg_matched

Stream Match Register Group matched. Fields:

desc string

A textual description of the register group.

n unsigned int

The SMRG that we matched against.

trans_id unsigned int

The transaction id that we are matching against.

start_ptw_read

Page Table Walk (read). This is the start of the physical access that the SMMU is making. Fields:

adomain enum

The actual attributes of the access that was used, after IMP DEF mangling.

aprot enum

The actual attributes of the access that was used, after IMP DEF mangling.

auser unsigned int

The AUSER fields.

hyp bool

Current mode is hyp.

inner_cache enum

The actual attributes of the access that was used, after IMP DEF mangling.

input_address unsigned int

Input address this lookup is for.

ns bool

This is for SSD Non-secure.

outer_cache enum

The actual attributes of the access that was used, after IMP DEF mangling.

pa_address unsigned int

Physical address.

st2_st1_context_banks unsigned int

The stage 1 and 2 context bank that this walk is for (if there is one, otherwise 0xFF).
Stage 1 is the lowest byte, stage 2 the highest.

stage_and_level unsigned int

Stage and level that this walk is for. Top byte is stage, bottom is level.

trans_id unsigned int

Transaction id.

start_stall_transaction

Indicating that a transaction is stalled. Fields:

context_id unsigned int

The context id that we are stalling against.

head_of_line bool

The transaction is the head-of-line to stall – it is the one that will pick up the value of SMMU_CBN_RESUME.

reason_for_stall enum

The state of the transaction as to why it stalled.

trans_id unsigned int

Transaction id of replaying transaction.

tlb_commentary

A commentary from the TLB about what is going on. Fields:

output string

The stream output.

tlb_entry_allocated

A TLB entry has been allocated. Fields:

index unsigned int

Index of TLB entry.

input_end_incl_address unsigned int

The end inclusive address of the input range that this matches.

input_start_address unsigned int

The start address of the input range that this matches.

output_end_incl_address unsigned int

The end inclusive address of the output range.

output_start_address unsigned int

The start address of the output range.

tlb_entry_invalidated

A TLB entry has been invalidated for some reason. Fields:

index unsigned int

Index of TLB entry.

reason enum

The reason why it was invalidated.

tlb_invalidate_command_executed

A TLB Invalidate command was executed. Fields:

address unsigned int

Address if appropriate.

asid unsigned int

ASID if appropriate.

command enum

Command queued.

context_bank unsigned int

Context bank number if queue_type is 'context_bank'.

last_level bool

Last level if appropriate.

ns bool

Security world if appropriate.

queue_type enum

Queue added to.

seq_id unsigned int

Sequence id for this TLB invalidate.

vmid unsigned int

VMID if appropriate.

tlb_invalidate_command_queued

A TLB Invalidate command was queued (but not synced). Fields:

address unsigned int

Address if appropriate.

asid unsigned int

ASID if appropriate.

command enum

Command queued.

context_bank unsigned int

Context bank number if queue_type is 'context_bank'.

last_level bool

Last level if appropriate.

ns bool

Security world if appropriate.

queue_type enum

Queue added to.

seq_id unsigned int

Sequence id for this TLB invalidate.

vmid unsigned int

VMID if appropriate.

tlb_sync

A TLB Sync command was issued. Fields:

context_bank unsigned int

Context bank number if queue_type is 'context_bank'.

ns bool

Security world.

queue_type enum

Queues that will be synced.

unpred_nested_st2_bank_is_not_marked_as_st2

The Stage 1 context bank is marked as Stage 1 + Stage 2 by SMMU_CBArn.TYPE; however, the Stage 2 context bank marked by SMMU_CBArn.CBNDX is not marked in it's SMMU_CBArn.TYPE as a Stage 2 context. Fields:

st1_context_bank_id unsigned int

The context bank id of the stage 1 bank.

st2_context_bank_id unsigned int

The context bank id of the stage 2 bank.

unpred_ptw_replay_asked_for_different_data

A PTW was in progress and was restarted (for some reason), but this time it asked for different data than last time. This means that the register file changed during the walk process and so the result of the translation for this transaction is unpredictable as to whether it would have seen the first or second walk data. Fields:

first_address unsigned int

The address of the first walk transaction that it performed.

first_ns bool

The NS state of the first walk transaction that it performed.

second_address unsigned int

The address that it is now walking for.

second_ns bool

The NS state that it is now walking for.

trans_id unsigned int

Transaction ID that was affected.

v7sd_access_fault

A L1 or L2 short descriptor access flag has disabled access for this (super-)section/page.

Fields:

context_bank unsigned int

Context bank.

input_address unsigned int

The input address used for this descriptor.

is_non_secure bool

Is this for the Non-secure side?.

level unsigned int

Level of walk.

v7sd_descriptor_faults

V7 Short Descriptor, level 'n' descriptor is marked as faulting. Fields:

context_bank unsigned int

Context bank.

input_address unsigned int

The input address used for this descriptor.

is_non_secure bool

Is this for the Non-secure side?.

level unsigned int

Level of walk.

v7sd_descriptor_fetch_failed

V7 Short Descriptor, level 'n' descriptor fetch failed. Fields:

context_bank unsigned int

Context bank.

ipa_address unsigned int

IPA of the descriptor address.

is_non_secure bool

Is the page walk and input addresses non-secure?.

level unsigned int

Level of walk.

pa_address unsigned int

PA of the descriptor address.

v7sd_walk_process_is_disabled

V7 Short Descriptor, walk process is disabled so producing a fault. Fields:

context_bank unsigned int

Context bank.

warning

These messages are about unusual (but not necessarily incorrect) activity occurring on the SMMU. Fields:

output string

The stream output.

warning_failed_to_clear_fsr_before_resuming

The programming model expects for the user to clear the fault in the FSR before calling SMMU_CBn_RESUME to unstage a transaction. Fields:

context_id unsigned int

Context ID of the FSR.

fault_flags_of_fsr unsigned int

Value of the fault bits of the FSR (excludes SS and the Format field).

warning_multiple_match_transaction

A transaction was encountered that multiple matched the stream id registers. Fields:

action enum

Action taken.

trans_id unsigned int

Transaction id.

warning_reg_after_doesnt_match_written_value

A write occurred that tried to set bits in a register, that for one reason or another, failed to get written. Fields:

desc string

The textual description of what happened.

2.124 MMU_600

This section describes the trace sources.

ArchMsg.Error.error

These messages are about activity occurring on the SMMU that is considered an error. Messages will only come out here if parameter `all_error_messages_through_trace` is true. `DISPLAY %{output}`. Fields:

output string

The stream output.

ArchMsg.Error.fetch_from_memory_type_not_supporting_httu

A descriptor fetch from an HTTU-enabled translation regime to an unsupported memory type was made. Whilst the fetch itself may succeed, if an update to the descriptor was attempted then it would fail. Fields:

address unsigned int

The address of the descriptor fetch.

desc_inner enum

Inner cacheability of descriptor.

desc_outer enum

Outer cacheability of descriptor.

desc_prot enum

The instruction/data, bus-ns/s and privileged/user marking.

desc_sh enum

Shareability of descriptor.

ssd enum

The SSD of the transaction.

ssd_ns bool

Is the Security State Determination of the transaction non-secure.

stage unsigned int

The stage at which we had a problem.

streamid unsigned int

The StreamID of the transaction.

substreamid unsigned int

The SubstreamID of the transaction (or ~0u if no substreamid).

trans_id unsigned int

Transaction id.

ArchMsg.Error.ns_cmdq

There is an error associated with the command queue. Fields:

cmd_id unsigned int

Command ID of the command in error.

cons unsigned int

CONS of the command.

what string

What is wrong.

ArchMsg.Error.priq_streamid_truncated

The SMMU received a PCIe PRI request with a StreamID) that was larger than that which the SMMU has been configured for. The StreamID that appears in the PRIQ entry will be truncated. Fields:

actual_streamid unsigned int

The actual StreamID that this request has.

sidsize unsigned int

The bit width of the SMMU for StreamIDs, as indicated by SMMU_IDR1.SIDSIZE.

trans_id unsigned int

The transaction ID that identifies this request.

trunc_streamid unsigned int

The truncated StreamID that will appear in the PRIQ entry.

ArchMsg.Error.priq_substreamid_truncated

The SMMU received a PCIe PRI request with a PASID prefix (SubstreamID) that was larger than that which the SMMU has been configured for. The SubstreamID that appears in the PRIQ entry will be truncated. Fields:

actual_substreamid unsigned int

The actual SubstreamID that this request has.

ssidsize unsigned int

The bit width of the SMMU for SubstreamIDs, as indicated by SMMU_IDR1.SSIDSIZE.

trans_id unsigned int

The transaction ID that identifies this request.

trunc_substreamid unsigned int

The truncated SubstreamID that will appear in the PRIQ entry.

ArchMsg.Error.s_cmdq

There is an error associated with the command queue. Fields:

cmd_id unsigned int

Command ID of the command in error.

cons unsigned int

CONS of the command.

what string

What is wrong.

ArchMsg.Error.tlb_entries_overlap

A TLB entry was attempted to be inserted into the TLB and was determined that it overlaps an existing entry. This check is not perfect but will catch simple errors. Fields:

do_f_tlb_conflict bool

Chosen to perform an F_TLB_CONFLICT.

end_address_of_new_entry unsigned int

End address of new entry.

end_address_of_old_entry unsigned int

End address of old entry.

index_of_new_entry unsigned int

Index of new entry.

index_of_old_entry unsigned int

Index of old entry.

start_address_of_new_entry unsigned int

Start address of new entry.

start_address_of_old_entry unsigned int

Start address of old entry.

why enum

Why it is an error for these two entries to overlap.

ArchMsg.Error.tlb_entry_not_invalidated_due_to_ril

An entry in the cache was not invalidated even though in the right address range because of the RIL fields of the command the entry do not match. Fields:

cmd_num unsigned int

The NUM field of the RIL part of the command.

cmd_ril_tg enum

The RIL_TG field of the RIL part of the command.

cmd_ril_ttl unsigned int

The TTL field of the RIL part of the command, zero means any level. 0x80 means match level 0 (and is from a DVM message).

cmd_scale unsigned int

The SCALE field of the RIL part of the command.

entry_id unsigned int

The entry id that is being invalidated.

tlb_entry string

The TLB entry.

ArchMsg.Error.vatos_sel_vmid_out_of_range

The SMMU_(S_)VATOS_SEL.VMID field was programmed with a VMID that was too wide for this implementation (SMMU_IDR0.VMID16 == 0). DISPLAY %{ssd_ns:(s-|ns-)}VMID:%{vmid} is out of range. Fields:

ssd_ns bool

The security state of the VATOS interface.

vmid unsigned int

The VMID programmed.

ArchMsg.Info.info

These are information messages about what is happening in the SMMU. DISPLAY %{output}. Fields:

output string

The stream output.

ArchMsg.Warning.CMD_RESUME_no_transaction_resumed

A CMD_RESUME was issued that matched no transaction. Fields:

stag unsigned int

STAG in the CMD_RESUME.

streamid unsigned int

StreamID in the CMD_RESUME.

streamid_ns bool

The StreamID was for the non-secure world.

ArchMsg.Warning.atc_inv_strange

Something was odd about the CMD_ATC_INV. DISPLAY CMD_ATC_INV strange as: %{why}. Fields:

cmd_id unsigned int

Command id.

ssd_of_cmdq enum

The SSD of the CMDQ.

why string

Why the CMD_ATC_INV was strange.

ArchMsg.Warning.bad_conf_reset_of_SMMU_S_GBPA_ABORT

The signal was not driven at reset by the external system, but was driven after we first started using the default value we had assumed. The external system, if it is going to drive the signal, must do so at first reset so that both sides of the connection can agree on the value of this signal. Fields:

default_value bool

The default value of the signal we have been assuming.

value bool

The value of the signal.

ArchMsg.Warning.bad_conf_system_supports_httu

The signal was not driven at reset by the external system, but was driven after we first started using the default value we had assumed. The external system, if it is going to drive the signal, must do so at first reset so that both sides of the connection can agree on the value of this signal. Fields:

default_value bool

The default value of the signal we have been assuming.

value bool

The value of the signal.

ArchMsg.Warning.bad_sec_override

The signal was not driven at reset by the external system, but was driven after we first started using the default value we had assumed. The external system, if it is going to drive the signal, must do so at first reset so that both sides of the connection can agree on the value of this signal. Fields:

default_value bool

The default value of the signal we have been assuming.

value bool

The value of the signal.

ArchMsg.Warning.bad_sup_btm

The signal was not driven at reset by the external system, but was driven after we first started using the default value we had assumed. The external system, if it is going to drive the signal, must do so at first reset so that both sides of the connection can agree on the value of this signal. Fields:

default_value bool

The default value of the signal we have been assuming.

value bool

The value of the signal.

ArchMsg.Warning.bad_sup_cohacc

The signal was not driven at reset by the external system, but was driven after we first started using the default value we had assumed. The external system, if it is going to drive the signal, must do so at first reset so that both sides of the connection can agree on the value of this signal. Fields:

default_value bool

The default value of the signal we have been assuming.

value bool

The value of the signal.

ArchMsg.Warning.bad_sup_sev

The signal was not driven at reset by the external system, but was driven after we first started using the default value we had assumed. The external system, if it is going to drive the signal, must do so at first reset so that both sides of the connection can agree on the value of this signal. Fields:

default_value bool

The default value of the signal we have been assuming.

value bool

The value of the signal.

ArchMsg.Warning.bad_tbu0_reset_in

The signal was not driven at reset by the external system, but was driven after we first started using the default value we had assumed. The external system, if it is going to drive the signal, must do so at first reset so that both sides of the connection can agree on the value of this signal. Fields:

default_value bool

The default value of the signal we have been assuming.

value bool

The value of the signal.

ArchMsg.Warning.bad_tcu_reset_in

The signal was not driven at reset by the external system, but was driven after we first started using the default value we had assumed. The external system, if it is going to drive the signal, must do so at first reset so that both sides of the connection can agree on the value of this signal. Fields:

default_value bool

The default value of the signal we have been assuming.

value bool

The value of the signal.

ArchMsg.Warning.contig_bit_gives_too_large_region_for_TxSZ

If the contig bit was used then the size of the contig region would be larger than that indicated by TxSZ. Fields:

desc_kind enum

Descriptor kind.

el enum

Exception level.

input_address unsigned int

Input address.

log2_contig_region_size unsigned int

Log2(contig region size in bytes).

output_address unsigned int

Output address.

stage_and_level unsigned int

The stage in bits [7:4] and the level in [3:0]. Level -1 is 0xF.

vmid unsigned int

VMID if appropriate.

ArchMsg.Warning.contig_bit_has_inconsistent_input_and_output_address

If the contig bit was used then the some bits of the output address held in the descriptor and the input address must match. Fields:

desc_kind enum

Descriptor kind.

el enum

Exception level.

input_address unsigned int

Input address.

log2_contig_region_size unsigned int

Log2(contig region size in bytes).

match_mask unsigned int

Bits that must match.

output_address unsigned int

Output address.

stage_and_level unsigned int

The stage in bits [7:4] and the level in [3:0]. Level -1 is 0xF.

vmid unsigned int

VMID if appropriate.

ArchMsg.Warning.fetch_from_memory_type_not_supporting_httu

A descriptor fetch from an HTTU-enabled translation regime to an unsupported memory type was made. Whilst the fetch itself may succeed, if an update to the descriptor was attempted then it would fail. Fields:

address unsigned int

The address of the descriptor fetch.

desc_inner enum

Inner cacheability of descriptor.

desc_outer enum

Outer cacheability of descriptor.

desc_prot enum

The instruction/data, bus-ns/s and privileged/user marking.

desc_sh enum

Shareability of descriptor.

ssd enum

The SSD of the transaction.

ssd_ns bool

Is the Security State Determination of the transaction non-secure.

stage unsigned int

The stage at which we had a problem.

streamid unsigned int

The StreamID of the transaction.

substreamid unsigned int

The SubstreamID of the transaction (or ~0u if no substreamid).

trans_id unsigned int

Transaction id.

ArchMsg.Warning.msi_address_truncated

An MSI was generated, but the address was silently truncated due to the limited downstream address bus width. Fields:

pmcg_index_and_counter unsigned int

If this is a PMCG interrupt, then the top 16 bits are the PMCG index, the lower 16 bits are counter index.

ras_error_group_id_node_id_and_record_index_in_node unsigned int

If this is a RAS interrupt then this is the error group record id in bits[23:16], the node_id in [15:8] and the record index in the node in [7:0].

truncated_address unsigned int

The truncated address.

untruncated_address unsigned int

The untruncated address.

which enum

Which MSI this is.

ArchMsg.Warning.msi_lost

An MSI was attempted to be sent, but couldn't be sent. Fields:

id unsigned int

ID of this interrupt transaction.

kind enum

What kind of interrupt.

why enum

Why this interrupt was denied.

ArchMsg.Warning.pmcg_non_secure_world

A commentary on potential mis-programming of the PMCG when transitioning the PMCG between security states. Fields:

commentary string

The commentary.

ArchMsg.Warning.pmcg_programming_violates_security

A commentary on potential mis-programming of the PMCG when transitioning the PMCG between different security states. This is under the assumption that an agent from another security state could be writing to the PMCG. Fields:

commentary string

The commentary.

ArchMsg.Warning.priq_auto_response_failed_to_find_STE

The PRIQ was going to generate an auto-response, but failed to find an STE and so is returning a Failure message to the EndPoint which should disable the PRI interface of the EndPoint. Fields:

trans_id unsigned int

The ID of the PPR that is being lost.

why enum

Why the PPR is being lost.

ArchMsg.Warning.priq_overflow_bad_acking

Indicates that an overflow condition was acknowledged by writing to:- SMMU_PRIQ_CONS.OVACKFLG but an overflow condition did not exist. Fields:

new_ovackflg bool

The new value of the SMMU_PRIQ_CONS.OVACKFLG.

ArchMsg.Warning.priq_smmuen_forces_effective_priqen_low

If SMMUEN == 0, then the effective value of PRIQEN is 0. This warning is triggered when PRIQEN == 1 && SMMUEN == 0; which may not be what was intended. The PRIQ cannot be active if SMMUEN == 0.

ArchMsg.Warning.sev_lost

A SEV was lost because it isn't supported according to SMMU_IDRO.SEV. DISPLAY SEV was lost because: %{why}. Fields:

why enum

Why the SEV was generated.

ArchMsg.Warning.smmu_pcie_rc_is_in_reset_ignoring_atc_invalidate

A CMD_PRI_RESP or CMD_ATC_INV was sent with a streamid that mapped to a PCIe Root Complex that is in reset. The message will be discarded and act as though completed successfully. Fields:

port_index unsigned int

The port index (node index) that is in reset but we would have sent it to.

streamid unsigned int

The StreamID that failed to map.

ArchMsg.Warning.smmu_pcie_rc_not_found_for_streamid

A CMD_PRI_RESP or CMD_ATC_INV was sent with a streamid that did not map to a PCIe Root Complex. This might be that the SW used an incorrect StreamID or it might be that the model has not been connected correctly. ATC Invalidate messages complete as though successful and PRI Requests are ignored. Fields:

streamid unsigned int

The StreamID that failed to map.

ArchMsg.Warning.smmu_pmusnapshot

Something strange happened on the pmusnapshot_req/pmusnapshot_ack interface. Fields:

pin_index unsigned int

If the PMCG index corresponds to an array of signals, this is the index in the array, or 0 otherwise.

pmcg_index enum

The PMCG index.

warning string

The warning message.

ArchMsg.Warning.suspicious_overlapping_entries

Two DPT TLB entries are overlapping but they differ in ways that are potentially a SW error. Fields:

new_AC enum

The value of the 'AC' field that controls access to this region.

new_FWB bool

The region is FWB.

new_VMID unsigned int

The VMID, if any, associated with this region.

new_entry_index unsigned int

The entry index of the new DPT TLB entry.

new_entry_is_vmsa_formed bool

The new entry was inserted as a result of VMSA information.

new_input_end_incl_address unsigned int

The end inclusive address of the input range that this matches.

new_input_start_address unsigned int

The start address of the input range that this matches.

new_output_pas enum

The output PAS of this DPT region.

new_ssd enum

The SSD of the streams this region captures.

new_writeable bool

True if this region is writeable.

old_AC enum

The value of the 'AC' field that controls access to this region.

old_FWB bool

The region is FWB.

old_VMID unsigned int

The VMID, if any, associated with this region.

old_entry_index unsigned int

The entry index of the old DPT TLB entry.

old_entry_is_vmsa_formed bool

The old entry was inserted as a result of VMSA information.

old_input_end_incl_address unsigned int

The end inclusive address of the input range that this matches.

old_input_start_address unsigned int

The start address of the input range that this matches.

old_output_pas enum

The output PAS of this DPT region.

old_ssd enum

The SSD of the streams this region captures.

old_writeable bool

True if this region is writeable.

pas_differ bool

The output PAS of the two entries differ.

vmids_differ bool

The VMIDs are used by at least one of the AC schemes and are different.

vmsa_formed_writeable_while_DPT_entry_is_not bool

The VMSA-formed entry is writeable but the DPT Entry says it is not writeable.

ArchMsg.Warning.warning

These messages are about unusual (but not necessarily incorrect) activity occurring on the SMMU. DISPLAY %{output}. Fields:

output string

The stream output.

ArchMsg.Warning.warning_effective_EOPD_differs_from_what_would_be_cached_in_TLB

Effective value of EOPD differs from what would be cached in the TLB DISPLAY transaction (%{transaction_id}), sid (%{sid}), ssid (%{ssid}), ssd (%{ssd}), effective EOPD (%{effective_EOPD}), cached EOPD (%{cached_EOPD}). Fields:

cached_EOPD bool

The EOPD value that would be cached in the TLB.

effective_EOPD bool

The effective value of EOPD.

ssd enum

SSD.

streamid unsigned int

StreamID or ~0ull if NoStreamID.

substreamid unsigned int

SubstreamID or ~0u if SSValid == false.

transaction_id unsigned int

The transaction ID.

DPTTLB.tlb_commentary

A commentary from the cache about what is going on. Fields:

output string

The stream output.

DPTTLB.tlb_entry_invalidated

An entry in the cache has been invalidated. Fields:

entry_id unsigned int

The entry id that is being invalidated.

reason enum

Why the entry is being invalidated.

EVENTQ_config

Configuration of interrupt updated. Fields:

address unsigned int

Address of the MSI.

data unsigned int

Data payload of the MSI.

irqen bool

Appropriate bit of SMMU_s_IRQ_CTRL controlling this interrupt.

memattr enum

Memory type.

ns bool

PAS of bus attribute is non-secure.

pas enum

PAS of the MSI.

ra bool

read-allocate attribute.

sh enum

Shareability.

smmu_idr0_msi bool

Are MSIs supported for this security world.

tr bool

transient attribute.

wa bool

write-allocate attributes.

EVENTQ_irqen

The IRQEN has been changed. Fields:

new_value bool

The new value of IRQEN.

EVENTQ_irqen_ack

The IRQEN change has been acked. Fields:

new_value bool

The new value of IRQEN.

GERROR_config

Configuration of interrupt updated. Fields:

address unsigned int

Address of the MSI.

data unsigned int

Data payload of the MSI.

irqen bool

Appropriate bit of SMMU_s_IRQ_CTRL controlling this interrupt.

memattr enum

Memory type.

ns bool

PAS of bus attribute is non-secure.

pas enum

PAS of the MSI.

ra bool

read-allocate attribute.

sh enum

Shareability.

smmu_idr0_msi bool

Are MSIs supported for this security world.

tr bool

transient attribute.

wa bool

write-allocate attributes.

GERROR_irqen

The IRQEN has been changed. Fields:

new_value bool

The new value of IRQEN.

GERROR_irqen_ack

The IRQEN change has been acked. Fields:

new_value bool

The new value of IRQEN.

HACDBS_PROCESSING_COMPLETE_config

Configuration of interrupt updated. Fields:

address unsigned int

Address of the MSI.

data unsigned int

Data payload of the MSI.

irqen bool

Appropriate bit of SMMU_s_IRQ_CTRL controlling this interrupt.

memattr enum

Memory type.

ns bool

PAS of bus attribute is non-secure.

pas enum

PAS of the MSI.

ra bool

read-allocate attribute.

sh enum

Shareability.

smmu_idr0_msi bool

Are MSIs supported for this security world.

tr bool

transient attribute.

wa bool

write-allocate attributes.

HACDBS_PROCESSING_COMPLETE_irqen

The IRQEN has been changed. Fields:

new_value bool

The new value of IRQEN.

HACDBS_PROCESSING_COMPLETE_irqen_ack

The IRQEN change has been acked. Fields:

new_value bool

The new value of IRQEN.

HDBSS_TABLE_FULL_config

Configuration of interrupt updated. Fields:

address unsigned int

Address of the MSI.

data unsigned int

Data payload of the MSI.

irqen bool

Appropriate bit of SMMU_s_IRQ_CTRL controlling this interrupt.

memattr enum

Memory type.

ns bool

PAS of bus attribute is non-secure.

pas enum

PAS of the MSI.

ra bool

read-allocate attribute.

sh enum

Shareability.

smmu_idr0_msi bool

Are MSIs supported for this security world.

tr bool

transient attribute.

wa bool

write-allocate attributes.

HDBSS_TABLE_FULL_irqen

The IRQEN has been changed. Fields:

new_value bool

The new value of IRQEN.

HDBSS_TABLE_FULL_irqen_ack

The IRQEN change has been acked. Fields:

new_value bool

The new value of IRQEN.

PRIQ_config

Configuration of interrupt updated. Fields:

address unsigned int

Address of the MSI.

data unsigned int

Data payload of the MSI.

irqen bool

Appropriate bit of SMMU_s_IRQ_CTRL controlling this interrupt.

memattr enum

Memory type.

ns bool

PAS of bus attribute is non-secure.

pas enum

PAS of the MSI.

ra bool

read-allocate attribute.

sh enum

Shareability.

smmu_idr0_msi bool

Are MSIs supported for this security world.

tr bool

transient attribute.

wa bool

write-allocate attributes.

PRIQ_irqen

The IRQEN has been changed. Fields:

new_value bool

The new value of IRQEN.

PRIQ_irqen_ack

The IRQEN change has been acked. Fields:

new_value bool

The new value of IRQEN.

SMMU_CR0ACK_SMMUEN_hazarded_by_priq

The SMMU_r_CR0ACK.SMMUEN cannot acknowledge the change to SMMUEN because there are outstanding PRIQ writes.

SMMU_CR0ACK_SMMUEN_update

The acknowledge to SMMU_CR0.SMMUEN was performed. Fields:

new_value bool

The new value of the register.

SMMU_CR0_SMMUEN_old_set_complete

A set of transactions associated with the old value of SMMU_CR0.SMMUEN completed. Fields:

last bool

This is the last set completing.

old_value bool

The old value of the set completing.

SMMU_CR0_SMMUEN_write

A write to SMMU_CR0.SMMUEN occurred. Fields:

new_value bool

The new value of the register.

old_value bool

The old value of the register.

SMMU_GBPA_old_set_complete

A set of transactions associated with the old value of SMMU_GBPA completed. Fields:

last bool

This is the last set completing.

old_value unsigned int

The old value of the set completing.

SMMU_GBPA_update

The Update flag to SMMU_GBPA was lowered. Fields:

new_value unsigned int

The new value of the register.

SMMU_GBPA_write

A write to SMMU_GBPA occurred. Fields:

new_value unsigned int

The new value of the register.

old_value unsigned int

The old value of the register.

SMMU_S_CR0ACK_SMMUEN_update

The acknowledge to SMMU_S_CR0.SMMUEN was performed. Fields:

new_value bool

The new value of the register.

SMMU_S_CR0_SMMUEN_old_set_complete

A set of transactions associated with the old value of SMMU_S_CR0.SMMUEN completed. Fields:

last bool

This is the last set completing.

old_value bool

The old value of the set completing.

SMMU_S_CR0_SMMUEN_write

A write to SMMU_S_CR0.SMMUEN occurred. Fields:

new_value bool

The new value of the register.

old_value bool

The old value of the register.

SMMU_S_GBPA_old_set_complete

A set of transactions associated with the old value of SMMU_S_GBPA completed. Fields:

last bool

This is the last set completing.

old_value unsigned int

The old value of the set completing.

SMMU_S_GBPA_update

The Update flag to SMMU_S_GBPA was lowered. Fields:

new_value unsigned int

The new value of the register.

SMMU_S_GBPA_write

A write to SMMU_S_GBPA occurred. Fields:

new_value unsigned int

The new value of the register.

old_value unsigned int

The old value of the register.

S_EVENTQ_config

Configuration of interrupt updated. Fields:

address unsigned int

Address of the MSI.

data unsigned int

Data payload of the MSI.

irqen bool

Appropriate bit of SMMU_s_IRQ_CTRL controlling this interrupt.

memattr enum

Memory type.

ns bool

PAS of bus attribute is non-secure.

pas enum

PAS of the MSI.

ra bool

read-allocate attribute.

sh enum

Shareability.

smmu_idr0_msi bool

Are MSIs supported for this security world.

tr bool

transient attribute.

wa bool

write-allocate attributes.

S_EVENTQ_irqen

The IRQEN has been changed. Fields:

new_value bool

The new value of IRQEN.

S_EVENTQ_irqen_ack

The IRQEN change has been acked. Fields:

new_value bool

The new value of IRQEN.

S_ERROR_config

Configuration of interrupt updated. Fields:

address unsigned int

Address of the MSI.

data unsigned int

Data payload of the MSI.

irqen bool

Appropriate bit of SMMU_s_IRQ_CTRL controlling this interrupt.

memattr enum

Memory type.

ns bool

PAS of bus attribute is non-secure.

pas enum

PAS of the MSI.

ra bool

read-allocate attribute.

sh enum

Shareability.

smmu_idr0_msi bool

Are MSIs supported for this security world.

tr bool

transient attribute.

wa bool

write-allocate attributes.

S_ERROR_irqen

The IRQEN has been changed. Fields:

new_value bool

The new value of IRQEN.

S_ERROR_irqen_ack

The IRQEN change has been acked. Fields:

new_value bool

The new value of IRQEN.

S_HACDBS_PROCESSING_COMPLETE_config

Configuration of interrupt updated. Fields:

address unsigned int

Address of the MSI.

data unsigned int

Data payload of the MSI.

irqen bool

Appropriate bit of SMMU_s_IRQ_CTRL controlling this interrupt.

memattr enum

Memory type.

ns bool

PAS of bus attribute is non-secure.

pas enum

PAS of the MSI.

ra bool

read-allocate attribute.

sh enum

Shareability.

smmu_idr0_msi bool

Are MSIs supported for this security world.

tr bool

transient attribute.

wa bool

write-allocate attributes.

S_HACDBS_PROCESSING_COMPLETE_irqen

The IRQEN has been changed. Fields:

new_value bool

The new value of IRQEN.

S_HACDBS_PROCESSING_COMPLETE_irqen_ack

The IRQEN change has been acked. Fields:

new_value bool

The new value of IRQEN.

S_HDBSS_TABLE_FULL_config

Configuration of interrupt updated. Fields:

address unsigned int

Address of the MSI.

data unsigned int

Data payload of the MSI.

irqen bool

Appropriate bit of SMMU_s_IRQ_CTRL controlling this interrupt.

memattr enum

Memory type.

ns bool

PAS of bus attribute is non-secure.

pas enum

PAS of the MSI.

ra bool

read-allocate attribute.

sh enum

Shareability.

smmu_idr0_msi bool

Are MSIs supported for this security world.

tr bool

transient attribute.

wa bool

write-allocate attributes.

S_HDBSS_TABLE_FULL_irqen

The IRQEN has been changed. Fields:

new_value bool

The new value of IRQEN.

S_HDBSS_TABLE_FULL_irqen_ack

The IRQEN change has been acked. Fields:

new_value bool

The new value of IRQEN.

TLB.tlb_commentary

A commentary from the cache about what is going on. Fields:

output string

The stream output.

TLB.tlb_entry_invalidated

An entry in the cache has been invalidated. Fields:

entry_id unsigned int

The entry id that is being invalidated.

reason enum

Why the entry is being invalidated.

atc_inv_nop

The CMD_ATC_INV command is ignored as a **NOP**. This may be emitted multiple times if the CMD_ATC_INV is being ignored for multiple reasons. Fields:

address unsigned int

Untranslated Address to invalidate.

cmd_id unsigned int

Command id.

cons unsigned int

CONS of the command.

global bool

Global flag.

log2_size_in_bytes unsigned int

Log 2 of the size in bytes of the region to invalidate.

ssd_of_queue enum

The SSD of the CMDQ.

ssd_of_streamid enum

The SSD of the StreamID.

streamid unsigned int

StreamID in the invalidate command.

substreamid unsigned int

SubstreamID or ~0u if SSValid == false.

why enum

Why was NOPped.

atos_complete_fault

The ATOS operation completed with a fault. Fields:

effective_stltranslate bool

Because of the settings and/or the ATOS type then the effective stltranslate can be different.

fault_faddr unsigned int

The fault FADDR.

fault_faultcode enum

The fault code.

fault_reason enum

The fault reason.

ssd_ns bool

This is a non-secure ATOS operation.

stltranslate bool

The StreamID/SubstreamID combination should have been translated by stage 1.

st2translate bool

The translation suffered a stage 2 translation.

state enum

The final transaction state of the ATOS operation.

vatos_vmid unsigned int

The VMID of the VATOS operation, or ~0u if not a VATOS operation.

atos_complete_fault_inv_req

The ATOS operation completed, faulted and generated an INV_REQ response. Fields:

ssd_ns bool

This is a non-secure ATOS operation.

vatos_vmid unsigned int

The VMID of the VATOS operation, or ~0u if not a VATOS operation.

why enum

Why the ATOS operation generated an INV_REQ.

atos_complete_success

The ATOS operation completed successfully. Fields:

base_addr unsigned int

The actual base address of region.

effective_st1translate bool

Because of the settings and/or the ATOS type then the effective st1translate can be different.

par_addr unsigned int

The PAR.ADDR field.

par_mair unsigned int

The memory attributes encoded as a MAIR.

par_ns bool

The PAR.NS field, for an SSD-ns request then this will always be 0.

par_sh enum

Shareability.

par_size bool

The PAR.Size field.

size_in_bytes unsigned int

The actual size in bytes of the region.

ssd_ns bool

This is a non-secure ATOS operation.

st1translate bool

The StreamID/SubstreamID combination should have been translated by stage 1.

st2translate bool

The translation suffered a stage 2 translation.

state enum

The final transaction state of the ATOS operation.

vatos_vmid unsigned int

The VMID of the VATOS operation, or ~0u if not a VATOS operation.

atos_run_set

The SMMU_s_GATOS_CTRL.RUN field was set to start the ATOS operation. Fields:

ssd_ns bool

This is a non-secure ATOS operation.

vatos bool

This is a VATOS operation.

atos_starting

The ATOS operation is starting. Fields:

addr unsigned int

The input address to the ATOS operation.

httui bool

Inhibit HTTU update.

ind bool

Instruction Data.

pnu bool

Privileged not User.

rnw bool

Read not Write.

ssd_ns bool

This is a non-secure ATOS operation.

ssec bool

If this is a secure ATOS operation then this is if it is secure or not.

streamid unsigned int

The StreamID requested.

substreamid unsigned int

The SubstreamID requested, or ~0u if no SubstreamID.

type enum

The requested ATOS type.

vatos_vmid unsigned int

The VMID of the VATOS operation, or ~0u if not a VATOS operation.

cd_cc.CD_commentary

A commentary from the cache about what is going on. Fields:

output string

The stream output.

cd_cc.CD_entry_invalidated

An entry in the cache has been invalidated. Fields:

entry_id unsigned int

The entry id that is being invalidated.

reason enum

Why the entry is being invalidated.

cd_entry_allocated

An CD entry has been allocated. Fields:

AssuredTranslation bool

The CD (and any L1CD) was fetched from an AssuredOnly stage 2 page and so might be capable of creating AssuredTranslations.

cd string

A textual description of the CD.

entry_id unsigned int

The id of the cache entry. Top bit is set if is a debug entry.

ns enum

For the non-secure world.

ssd enum

The SSD of the entry.

streamid unsigned int

The StreamID of the entry this will match.

substreamid unsigned int

The SubstreamID of the entry this will match. This may be zero for transactions without a SubstreamID.

conf_reset_of_SMMU_S_GBPA_ABORT

The pin is driven. This is the reset value of SMMU_S_GBPA.ABORT. This pin is sampled on the negedge of the reset pin and has to be driven beforehand for the value to take effect. Fields:

value bool

The value of the signal.

conf_system_supports_httu

The pin is driven. This indicates the system supports HTTU. This pin is sampled on the negedge of the reset pin and has to be driven beforehand for the value to take effect. Fields:

value bool

The value of the signal.

dpttlb_entry_allocated

A DPT TLB entry has been allocated. Fields:

AC enum

The value of the 'AC' field that controls access to this region.

FWB bool

The region is FWB.

VMID unsigned int

The VMID, if any, associated with this region.

index unsigned int

Index of the TLB entry.

input_end_incl_address unsigned int

The end inclusive address of the input range that this matches.

input_start_address unsigned int

The start address of the input range that this matches.

output_pas enum

The output PAS of this DPT region.

ssd enum

The SSD of the streams this region captures.

trans_id unsigned int

The trans_id of the transaction that caused this allocation.

vmsa_formed bool

The entry was formed from VMSA information rather than from a DPT walk.

writable bool

True if this region is writeable.

dpttlb_invalidate_intersects_but_does_not_cover_entry_range

ENCODED_SIZE < size of the region covered by the DPT entry so invalidation is not architecturally guaranteed. No invalidation is performed. Fields:

dpttlb_entry_id unsigned int

ID of the TLB entry.

entry_end_incl_address unsigned int

Last address covered by the TLB entry.

entry_start_address unsigned int

First address covered by the TLB entry.

invalidate_end_incl_address unsigned int

Last address covered by the invalidation range.

invalidate_start_address unsigned int

First address covered by the invalidation range.

ssd enum

The security state of the TLB entry.

dpttlb_overlapping_entries

Two DPT TLB entries are overlapping. Fields:

new_AC enum

The value of the 'AC' field that controls access to this region.

new_FWB bool

The region is FWB.

new_VMID unsigned int

The VMID, if any, associated with this region.

new_entry_index unsigned int

The entry index of the new DPT TLB entry.

new_entry_is_vmsa_formed bool

The new entry was inserted as a result of VMSA information.

new_input_end_incl_address unsigned int

The end inclusive address of the input range that this matches.

new_input_start_address unsigned int

The start address of the input range that this matches.

new_output_pas enum

The output PAS of this DPT region.

new_ssd enum

The SSD of the streams this region captures.

new_writeable bool

True if this region is writeable.

old_AC enum

The value of the 'AC' field that controls access to this region.

old_FWB bool

The region is FWB.

old_VMID unsigned int

The VMID, if any, associated with this region.

old_entry_index unsigned int

The entry index of the old DPT TLB entry.

old_entry_is_vmsa_formed bool

The old entry was inserted as a result of VMSA information.

old_input_end_incl_address unsigned int

The end inclusive address of the input range that this matches.

old_input_start_address unsigned int

The start address of the input range that this matches.

old_output_pas enum

The output PAS of this DPT region.

old_ssd enum

The SSD of the streams this region captures.

old_writeable bool

True if this region is writeable.

dvm_tlbinvalidate_complete

The DVM TLB Invalidate message completed. Fields:

id unsigned int

The unique id of this DVM message.

ok bool

The DVM message was OK.

dvm_tlbinvalidate_received

A DVM message for a TLB Invalidate has been received. Fields:

address unsigned int

The VA or IPA to use if match_address.

asid unsigned int

The ASID to match if match_asid.

by_ipa bool

The operation is for an IPA operation if match_address.

id unsigned int

The unique id of this DVM message.

ignored enum

The DVM message was ignored.

last_level bool

The operation is for last level if supported.

match_address bool

Match the address field.

match_asid bool

Match the asid field.

match_vmid bool

Match the vmid field.

num unsigned int

If a range operation, the NUM field. If a single-address operation this is 0.

prot enum

The protection level for which this TLB Invalidate will operate on.

security_world enum

The security world that this will apply to.

smmu_scale unsigned int

If a range operation, then the SCALE field with the meaning in the SMMU architecture which is different to the PE architecture. If a single-address operation this is 0.

stage1_only bool

The operation is for stage 1 only if supported.

tg enum

If a single-address or address-range operation, then the Translation Granule hint. Address-range operations always supply a Translation Granule.

translation_table_level enum

The leaf level of the translation table.

vmid unsigned int

The VMID to match if match_vmid.

found_tlb_entry_has_different_aset

Architecturally, a particular ASID either should be ASET0 or ASET1. However, we have managed to find a TLB entry that has a different ASET than that which we were searching for. This indicates a programming error. You should examine all contexts with this particular ASID/VMID and ensure they are consistent. Fields:

aset unsigned int

The ASET of this entry.

asid unsigned int

ASID if appropriate.

index unsigned int

Index of TLB entry.

input_start_address unsigned int

The start address of the input range that this matches.

scheme enum

The tagging scheme used.

ssd enum

The SSD of the tagging scheme.

ssd_ns enum

Security State.

vmid unsigned int

VMID if appropriate.

httu_update_abandoned_update

The HTTU update of a descriptor in memory was potentially possible, but it was behind an update that failed to apply cleanly. Fields:

AF enum

Whether an AF update should/could occur.

DBM enum

Whether a DBM update should/could occur.

address unsigned int

Address of HTTU update.

httu_inner enum

Inner cacheability of descriptor to update.

httu_outer enum

Outer cacheability of descriptor to update.

httu_prot enum

The instruction/data, bus-ns/s and privileged/user marking.

httu_sh enum

Shareability of descriptor to update.

mpam_pmg_and_partid unsigned int

Bits[23:16] are the MPAM PMG. Bits[15:0] are the physical PARTID.

mpam_sp enum

The MPAM_SP of the transaction.

original_descriptor unsigned int

The original descriptor that the model observed.

stage_and_level unsigned int

The stage (bits[7:4]) and level (bits[3:0]). A level of 0xF means -1.

trans_id unsigned int

The transaction id that this HTTU update is associated with.

what enum

What this descriptor represents.

httu_update_end_update

The attempted update of a descriptor in memory has occurred. Fields:

is_big_endian bool

The descriptor is big-endian in memory.

original_descriptor unsigned int

The original descriptor value.

result enum

The result of the attempt to update.

trans_id unsigned int

The transaction id that this HTTU update is associated with.

try_to_change_to_descriptor unsigned int

The new descriptor value that tried to replace the original.

value_that_was_in_memory unsigned int

The value that the compare-and-swap operation returned as the value that was in memory.

httu_update_not_done

A discretionary HTTU update could occur and the implementation choose not to do it. A mandatory_do_if_linked_performed means that this is a stage 2 leaf descriptor that maps a subsequent stage 1 leaf descriptor whose update is discretionary and if that subsequent discretionary update is going to occur then this update becomes mandatory. Fields:

AF enum

Whether an AF update should/could occur.

DBM enum

Whether a DBM update should/could occur.

address unsigned int

Address of HTTU update.

httu_inner enum

Inner cacheability of descriptor to update.

httu_outer enum

Outer cacheability of descriptor to update.

httu_prot enum

The instruction/data, bus-ns/s and privileged/user marking.

httu_sh enum

Shareability of descriptor to update.

mpam_pmg_and_partid unsigned int

Bits[23:16] are the MPAM PMG. Bits[15:0] are the physical PARTID.

mpam_sp enum

The MPAM_SP of the transaction.

original_descriptor unsigned int

The original descriptor that the model observed.

stage_and_level unsigned int

The stage (bits[7:4]) and level (bits[3:0]). A level of 0xF means -1.

trans_id unsigned int

The transaction id that this HTTU update is associated with.

what enum

What this descriptor represents.

httu_update_start_update

An HTTU update could occur and the implementation chose to try it. A mandatory_do_if_linked_performed means that this is a stage 2 leaf descriptor that maps a subsequent stage 1 leaf descriptor whose update is discretionary and if that subsequent discretionary update is going to occur then this update becomes mandatory. Fields:

AF enum

Whether an AF update should/could occur.

DBM enum

Whether a DBM update should/could occur.

address unsigned int

Address of HTTU update.

httu_inner enum

Inner cacheability of descriptor to update.

httu_outer enum

Outer cacheability of descriptor to update.

httu_prot enum

The instruction/data, bus-ns/s and privileged/user marking.

httu_sh enum

Shareability of descriptor to update.

is_big_endian bool

The descriptor will be written to memory as big-endian.

mecid unsigned int

The masked MECID used for the update transaction, or ~0u if not appropriate.

mpam_pmg_and_partid unsigned int

Bits[23:16] are the MPAM PMG. Bits[15:0] are the physical PARTID.

mpam_sp enum

The MPAM_SP of the transaction.

original_descriptor unsigned int

The original descriptor that the model observed.

stage_and_level unsigned int

The stage (bits[7:4]) and level (bits[3:0]). A level of 0xF means -1.

trans_id unsigned int

The transaction id that this HTTU update is associated with.

try_to_change_to_descriptor unsigned int

The new descriptor value that is going to try replace the original.

what **enum**

What this descriptor represents.

will_do_AF **bool**

What the implementation chose to do for the AF update.

will_do_DBM **bool**

What the implementation chose to do for the DBM update.

integration_mode_end_ras_level_interrupt_restored

RAS level sensitive interrupt restored due to integration mode ending. Fields:

is_tcu **bool**

Is TCU integration mode.

ras_interrupt **enum**

RAS interrupt being restored.

tbu_index **unsigned int**

TBU index. ~0u if it's TCU.

value **bool**

Level of the interrupt signal.

integration_mode_pmcg_interrupt_lost

PMCG interrupt lost due to being in integration mode. Fields:

is_tcu **enum**

Is TCU integration mode.

pmcg_interrupt **enum**

PMCG interrupt being dropped.

tbu_index **unsigned int**

TBU index. ~0u if it's TCU.

integration_mode_pmusnapshot_ack_lost

pmusnapshot_ack lost due to being in integration mode. Fields:

is_tcu **enum**

Is TCU integration mode.

tbu_index **unsigned int**

TBU index. ~0u if it's TCU.

integration_mode_ras_interrupt_lost

RAS interrupt lost due to being in integration mode. Fields:

is_tcu **enum**

Is TCU integration mode.

ras_interrupt **enum**

RAS interrupt being dropped.

tbu_index unsigned int

TBU index. ~0u if it's TCU.

integration_mode_ras_level_interrupt_lost

RAS level sensitive interrupt lost due to being in integration mode. Fields:

is_tcu bool

Is TCU integration mode.

ras_interrupt enum

RAS interrupt being dropped.

tbu_index unsigned int

TBU index. ~0u if it's TCU.

value bool

Level of the interrupt signal.

integration_mode_start_ras_level_interrupt_cleared

RAS level sensitive interrupt cleared due to integration mode starting. Fields:

is_tcu bool

Is TCU integration mode.

ras_interrupt enum

RAS interrupt being cleared.

tbu_index unsigned int

TBU index. ~0u if it's TCU.

integration_mode_tcu_evento_lost

Evento lost due to being in integration mode.

integration_mode_tcu_interrupt_lost

Interrupt lost due to being in integration mode. Fields:

interrupt enum

Interrupt that is being dropped.

integration_register_change

An integration register write occurred and it's modified, which drives a signal. Fields:

is_tcu enum

Is TCU integration mode.

new_value bool

New value of the signal.

old_value bool

Old value of the signal.

register enum

The register being modified.

register_value unsigned int

Value written to the register.

signal string

Signal affected by the register write.

tbu_index unsigned int

TBU index. ~0u if it's TCU.

interrupt_returned

An interrupt/MSI returned from downstream. Fields:

InD_NS_PnU enum

Instruction/Data, Non-secure/Secure/Root/Realm, Privileged/User. Bit[1] = PAS[0], and bit[3] = PAS[1].

address unsigned int

Address of the MSI.

data unsigned int

Data payload of the MSI.

dcmdqcp_and_index unsigned int

If this not a DCMDQ MSI CMD_SYNC, then this is 0xFFFFFFFF. If it is then the DCMDQ QCP index is in bits[31:16] and the index in the QCP is in [7:0].

gpc_denied_msi bool

True if the MSI was denied as it failed its GPC checks. Thus the field 'ok' will be false.

id unsigned int

ID of this interrupt transaction.

inner_cache enum

The actual attributes of the access.

kind enum

What kind of interrupt.

mecid unsigned int

The MECID of the MSI, if applicable.

mpam_ns bool

For a non-RME system, this is true if the MPAM space is Non-secure. For an RME system, this is MPAM_SP[0].

mpam_pmg_and_partid unsigned int

Bits[23:16] are the MPAM PMG. Bits[15:0] are the physical PARTID.

mpam_sp enum

The MPAM_SP security state of the MPAM partition. 0 – secure, 1 – non-secure, 2 – root, 3 – realm.

msi_valid bool

MSI will attempt to be sent.

ok bool

Did the access return OK or an abort?.

outer_cache enum

The actual attributes of the access.

pas enum

The PAS of the MSI, if applicable.

pmcg_index_and_counter unsigned int

If this is a PMCG interrupt, then the top 16 bits are the PMCG index, the lower 16 bits are counter index.

ras_error_group_id_node_id_and_record_index_in_node unsigned int

If this is a RAS interrupt then this is the error group record id in bits[23:16], the node_id in [15:8] and the record index in the node in [7:0].

sh enum

Shareability.

interrupt_sent

An interrupt is raised. If it sends an MSI then this is *after* any device-dependent transform on the architectural attributes and so may differ from what is programmed. Fields:

InD_NS_PnU enum

Instruction/Data, Non-secure/Secure/Root/Realm, Privileged/User. Bit[1] = PAS[0], and bit[3] = PAS[1].

address unsigned int

Address of the MSI.

data unsigned int

Data payload of the MSI.

dcmdq_device_id unsigned int

If this is a DCMDQ then this is the DeviceID of the MSI write. Otherwise, 0xFFFF'ffff.

dcmdq_qcp_and_index unsigned int

If this not a DCMDQ MSI CMD_SYNC, then this is 0xFFFF'ffff. If it is then the DCMDQ QCP index is in bits[31:16] and the index in the QCP is in [7:0].

id unsigned int

ID of this interrupt transaction.

inner_cache enum

The actual attributes of the access.

kind enum

What kind of interrupt.

mecid unsigned int

The MECID of the MSI, if applicable.

mpam_ns bool

For a non-RME system, this is true if the MPAM space is Non-secure. For an RME system, this is MPAM_SP[0].

mpam_pmg_and_partid unsigned int

Bits[23:16] are the MPAM PMG. Bits[15:0] are the physical PARTID.

mpam_sp enum

The MPAM_SP security state of the MPAM partition. 0 – secure, 1 – non-secure, 2 – root, 3 – realm.

msi_valid bool

MSI will attempt to be sent.

outer_cache enum

The actual attributes of the access.

pas enum

The PAS of the MSI, if applicable.

pmcg_index_and_counter unsigned int

If this is a PMCG interrupt, then the top 16 bits are the PMCG index, the lower 16 bits are counter index.

qSID unsigned int

If this is a DCMDQ then this is the qSID of the MSI write. Otherwise, 0xFFFF'ffff.

ras_error_group_id_node_id_and_record_index_in_node unsigned int

If this is a RAS interrupt then this is the error group record id in bits[23:16], the node_id in [15:8] and the record index in the node in [7:0].

sh enum

Shareability.

l1cd_cc.L1CD_commentary

A commentary from the cache about what is going on. Fields:

output string

The stream output.

l1cd_cc.L1CD_entry_invalidated

An entry in the cache has been invalidated. Fields:

entry_id unsigned int

The entry id that is being invalidated.

reason enum

Why the entry is being invalidated.

l1cd_entry_allocated

An L1 CD entry has been allocated. Fields:

AssuredTranslation bool

The L1CD was fetched from an AssuredOnly stage 2 page and so might be capable of creating AssuredTranslations.

entry_id unsigned int

The id of the cache entry. Top bit is set if is a debug entry.

normalised_substreamid unsigned int

The first SubstreamID of the range of SubstreamIDs that this L1CD entry will match.

ns enum

For the non-secure world.

pa_l2 unsigned int

The PA of the L2 CD table. This is L2Ptr << 12.

ssd enum

The SSD of the entry.

streamid unsigned int

The StreamID this CD is for.

valid bool

Is the entry valid.

l1ste_cc.L1STE_commentary

A commentary from the cache about what is going on. Fields:

output string

The stream output.

l1ste_cc.L1STE_entry_invalidated

An entry in the cache has been invalidated. Fields:

entry_id unsigned int

The entry id that is being invalidated.

reason enum

Why the entry is being invalidated.

l1ste_entry_allocated

An L1 STE entry has been allocated. Fields:

entry_id unsigned int

The id of the cache entry. Top bit is set if is a debug entry.

first_streamid_of_range unsigned int

The first StreamID of the range of StreamIDs that this L1STE entry will match.

ns enum

For the non-secure world.

num_entries_in_l2 unsigned int

The number of entries in the L2 table. This is $2^{**}(\text{Span}-1)$ or 0 if invalid.

pa_l2 unsigned int

The PA of the L2 ST table. This is L2Ptr << 6 and aligned to the size of the table.

ssd enum

The SSD of the entry.

level_interrupt_sent

A level interrupt changed state. Fields:

kind enum

What kind of interrupt.

ras_error_group_id_node_id_and_record_index_in_node unsigned int

If this is a level RAS interrupt then this is the error group record id in bits[23:16], the node_id in [15:8] and the record index in the node in [7:0].

set_high bool

Level interrupt state.

ns_cmd_sync_completed_irq

“Non-secure” CMD_SYNC completed its IRQ action. Fields:

cmd_id unsigned int

The ID of the command being issued.

cons_incl_wrap unsigned int

CONS pointer of CMD_SYNC.

is_synthetic bool

This is a synthetic sync. The cons_incl_wrap is 0xFFFF'ffff.

ns_cmd_sync_completed_no_action

Non-secure CMD_SYNC completed, there was no CMD_SYNC action (SIG_NONE). Fields:

cmd_id unsigned int

The ID of the command being issued.

cons_incl_wrap unsigned int

CONS pointer of CMD_SYNC.

is_synthetic bool

This is a synthetic sync. The cons_incl_wrap is 0xFFFF'ffff.

ns_cmd_sync_completed_sev

Non-secure CMD_SYNC completed its action of SIG_SEV. Fields:

cmd_id unsigned int

The ID of the command being issued.

cons_incl_wrap unsigned int

CONS pointer of CMD_SYNC.

is_synthetic bool

This is a synthetic sync. The cons_incl_wrap is 0xFFFF'ffff.

ns_cmd_sync_completed_with_error

Non-secure CMD_SYNC completed, however, there was an error associated with the completion of the CMD_SYNC. Fields:

error enum

The error associated with the CMD_SYNC.

cmd_id unsigned int

The ID of the command being issued.

cons_incl_wrap unsigned int

CONS pointer of CMD_SYNC.

is_synthetic bool

This is a synthetic sync. The cons_incl_wrap is 0xFFFF'ffff.

ns_cmd_sync_issued

Non-secure CMD_SYNC has now completed issuing and we now know if something is hazarding the CMD_SYNC from completing. Fields:

cmd_id unsigned int

The ID of the command being issued.

cons_incl_wrap unsigned int

CONS pointer of CMD_SYNC.

hazarded bool

The CMD_SYNC is currently hazarded from completing.

is_synthetic bool

This is a synthetic sync. The cons_incl_wrap is 0xFFFF'ffff.

ns_cmd_sync_issuing

Non-secure CMD_SYNC is starting to issue, we are going to start accounting for which things have to be part of the prior set of things that have to complete before the CMD_SYNC can complete. Fields:

cmd_id unsigned int

The ID of the command being issued.

cons_incl_wrap unsigned int

CONS pointer of CMD_SYNC.

is_synthetic bool

This is a synthetic sync. The cons_incl_wrap is 0xFFFF'ffff.

ns_cmd_sync_starting_completion_action_irq

Non-secure CMD_SYNC starting IRQ completion action. Fields:

cmd_id unsigned int

The ID of the command being issued.

cons_incl_wrap unsigned int

CONS pointer of CMD_SYNC.

is_synthetic bool

This is a synthetic sync. The cons_incl_wrap is 0xFFFF'ffff.

msiaddress unsigned int

The MSI address to use (or 0 if not sending an MSI).

msidata unsigned int

The MSI data to send.

ns_cmd_sync_starting_completion_action_sev

Non-secure CMD_SYNC starting its completion action of SIG_SEV. Fields:

cmd_id unsigned int

The ID of the command being issued.

cons_incl_wrap unsigned int

CONS pointer of CMD_SYNC.

is_synthetic bool

This is a synthetic sync. The cons_incl_wrap is 0xFFFF'ffff.

ns_cmdq_cmd_sync_error

The CMD_SYNC can fault if an ATC Invalidate timeout occurs. The timeout will be attached to the next CMD_SYNC on the queue on which the invalidate was launched. Fields:

cerror enum

The CERROR to attach.

cmd_queue unsigned int

The top bits indicate the CMDQ id. These bits are attached to all cmd_ids generated for this queue. The bits [63:44] contain the queue ID. [63:62] 0: s-ssd, 1: reserved, 2: ns-ssd. 3: rl-ssd [61] is ECMDQ [59:52] QCP index [51:44] ECMDQ index in QCP.

herror enum

The HERROR to attach, or HERROR_NONE if we are attaching a CERROR.

ns_cmdq_cmd_sync_error_delivered

The CMD_SYNC can fault if an ATC Invalidate timeout occurs. This is the CMD_SYNC delivering that fault. Fields:

cerror enum

The CERROR to attach.

cmd_id unsigned int

Command ID of the CMD_SYNC.

ns_cmdq_ecmdq_enack

Trace the changes to ENACK. Fields:

error enum

The error being exposed, if any.

cmd_queue unsigned int

The top bits indicate the CMDQ id. These bits are attached to all cmd_ids generated for this queue. The bits [63:44] contain the queue ID. [63:62] 0: s-ssd, 1: reserved, 2: ns-ssd. 3: rl-ssd [61] is ECMDQ [59:52] QCP index [51:44] ECMDQ index in QCP.

enack bool

The value of ENACK.

forced_error bool

The queue was enabled with ERR != ERRACK and so the implementation chooses to trigger the error reporting mechanism.

ns_cmdq_issue

A command is actually being executed. Fields:

cmd_id unsigned int

Command ID of the command being executed.

cons unsigned int

CONS of the command.

what string

What is being executed.

ns_cmdq_pointers_state

The command queue pointers. Fields:

cmd_queue unsigned int

The top bits indicate the CMDQ id. These bits are attached to all cmd_ids generated for this queue. The bits [63:44] contain the queue ID. [63:62] 0: s-ssd, 1: reserved, 2: ns-ssd. 3: rl-ssd [61] is ECMDQ [59:52] QCP index [51:44] ECMDQ index in QCP.

disable_fetch bool

The model has decided that as you programmed PROD/CONS inconsistently that it would not fetch as it has no confidence as to which are commands and which are not.

q_issuable_and_issued_cons unsigned int

The CONS pointer of the region that is issuable and/or issued (including wrap).

q_issuable_and_issued_prod unsigned int

The PROD pointer of the region that is issuable and/or issued (including wrap).

q_programmer_cons unsigned int

The CONS pointer as seen by the programmer (including wrap).

q_programmer_prod unsigned int

The PROD pointer as seen by the programmer (including wrap).

q_unfetched_cons unsigned int

The CONS pointer of the region unfetched (including wrap).

q_unfetched_prod unsigned int

The PROD pointer of the region unfetched (including wrap).

ns_cmdq_state

A description of what the CMDQ can do now. Fields:

can_cerror bool

There is a CERROR that can be recognised.

can_cmd_sync bool

There is a CMD_SYNC that can be recognised.

can_fetch bool

The SMMU can fetch.

can_herror bool

There is an HERROR that can be recognised.

can_issue bool

There are fetched commands that are eligible to be issued.

can_update_cons bool

The programmer view of CMDQ_CONS can be moved to say some have been consumed.

cmd_queue unsigned int

The top bits indicate the CMDQ id. These bits are attached to all cmd_ids generated for this queue. The bits [63:44] contain the queue ID. [63:62] 0: s-ssd, 1: reserved, 2: ns-ssd. 3: rl-ssd [61] is ECMDQ [59:52] QCP index [51:44] ECMDQ index in QCP.

cmd_sync enum

Why a CMD_SYNC cannot be issued.

cmdqen bool

CMDQEN, when enabled then the CMDQ can fetch commands.

cmdqen_1_to_0_unacknowledged bool

CMDQEN has been set to 0, but we have yet to acknowledge it.

current-real-cerror enum

The current CERROR that would be visible to the programmer if there was an error.

fetch-disabled_due_to_programmer_error bool

The PROD/CONS pointer was inconsistent and the model decided to stop processing commands.

fetch-number_of_unfetched_commands unsigned int

The number of commands that have yet to be fetched from memory, or ~0u if CMDQEN == 0.

fetch-state enum

The state of the fetch state machine.

fetch-too_many_commands_outstanding_so_not_fetching_any_more bool

If there are too many issued commands, or pending to be issued then we won't fetch any more.

issue-number_of_issuable_commands unsigned int

The number of commands that have been fetched but not yet issued.

issue-number_of_unissuable_commands unsigned int

This might include commands behind an illegal command, or behind a CMD_SYNC. Under some circumstances this can include a command representing an external abort.

issue-state enum

The internal issued state machine state.

outstanding_fetches_since_reset unsigned int

The number of outstanding fetches.

synthetic_sync_can_be_completed bool

The synthetic sync that is outstanding can be completed.

waiting_for_synthetic_sync bool

There is a synthetic sync that is outstanding.

ns_eventq_adding_to_cmd_sync_prior_set

The event created by the specified transaction id must become visible to the programmer before the CMD_SYNC can complete. Fields:

trans_id unsigned int

The transaction id that caused the event.

ns_eventq_auto_retry_stalled_transaction

A transaction that was stalled but not yet reported on the Event Queue was retried. The retry can happen if the transaction didn't write because it was de-duplicated behind a reported stalling transaction and that stalling transaction was CMD_RESUMEd, or if the queue was unwritable and the queue is now writable. Fields:

trans_id unsigned int

Transaction id of the auto-retried transaction.

why enum

Why the transaction retried.

ns_eventq_cmd_sync_unhazardous

The CMD_SYNC has been unhazardous as the prior set is empty and as much of it as required is visible in the programmers' view. Fields:

prod_incl_wrap unsigned int

The prod index including wrap bit that was required to become visible.

prog_prod_incl_wrap unsigned int

The current programmer-visible prod including wrap bit.

ns_eventq_eventqen_and_eventqenack

Trace the values of EVENTQEN and EVENTQENACK as they change. Fields:

about enum

Some more information about the queue state.

cons_incl_wrap unsigned int

The CONS pointer (including wrap bits).

eventqen bool

EVENTQEN.

eventqenack bool

EVENTQENACK.

prod_incl_wrap unsigned int

The PROD pointer (including wrap bits).

ns_eventq_external_abort

A particular eventq record got an external abort. The event will subsequently appear in the ns/s_eventq_lost_event_records trace. Fields:

prod unsigned int

The prod pointer (excluding wrap bit) of the event record that aborted.

trans_id unsigned int

The transaction id that produced the event record that aborted.

ns_eventq_lost_event_records

Event records that are lost and never record appear on this trace. Fields:

dword0 unsigned int

Double word 0 of event.

dword1 unsigned int

Double word 1 of event.

dword2 unsigned int

Double word 2 of event.

dword3 unsigned int

Double word 3 of event.

event enum

Event code that was thrown away.

prod unsigned int

The PROD index of the event if appropriate or ~0u otherwise.

reason enum

The reason the record was lost.

trans_id unsigned int

The transaction id that this corresponds to.

ns_eventq_overflow_acknowledged

When events are lost then they appear on the eventq_lost_event_records_* trace. This will activate the overflow flag in the corresponding SMMU_s_EVENTQ_PROD register. Software can then acknowledge that flag using the overflow acknowledge flag in SMMU_s_EVENTQ_CONS. Fields:

bad bool

If true then the programmer has attempted to acknowledge an overflow condition that didn't exist.

count_of_events_lost_due_to_overflow unsigned int

This is the number of events that were lost because the event queue was full. This does not include any lost because of an EVENTQ_ABT_ERR.

ns_eventq_pending_event_records

Event records that are pending to be dealt with by the eventq state machine. A specific event can be traced multiple times if it becomes part of the prior set of transactions of a CMD_SYNC that has to complete before the CMD_SYNC is allowed to be completed. Fields:

dword0 unsigned int

Double word 0 of event.

dword1 unsigned int

Double word 1 of event.

dword2 unsigned int

Double word 2 of event.

dword3 unsigned int

Double word 3 of event.

event enum

Event code.

is_prior_set bool

The transaction forms part of the prior set of a CMD_SYNC that have to become visible (or thrown away if necessary) before the CMD_SYNC can complete.

trans_id unsigned int

The transaction id that this corresponds to.

ns_eventq_prod_cons_updated

This records when the programmer visible PROD/CONS pointer is updated. More can be visible in memory than are advertised in the programmer-visible PROD pointer. This also traces the latest PROD that could have been advertised. Fields:

mem_prod unsigned int

The PROD pointer value for all records that are visible in the memory.

num_events_available unsigned int

The number of events available to the programmer.

prog_cons unsigned int

The programmer-visible CONS pointer.

prog_prod unsigned int

The programmer-visible PROD pointer.

what enum

What changed?.

ns_eventq_raising_eventq_abt_err

EVENTQ_ABT_ERR can now be raised.

ns_eventq_removing_from_cmd_sync_prior_set

The event captured by a CMD_SYNC and created by the specified transaction id is now in memory. Fields:

prod unsigned int

The prod index (no wrap bit) that must become visible to the programmer to see this record. If not head_of_line then the record will not become visible and this field is meaningless.

reason enum

The reason the event was removed.

trans_id unsigned int

The transaction id that caused the event.

ns_eventq_stashing_unreported_stalled_transaction_for_auto_retry

A stall event record would have been lost, instead it is stashed away to auto-retry when it can. Fields:

dword0 unsigned int

Double word 0 of event.

dword1 unsigned int

Double word 1 of event.

dword2 unsigned int

Double word 2 of event.

dword3 unsigned int

Double word 3 of event.

event enum

Event code that was thrown away.

reason enum

The reason the record was lost.

trans_id unsigned int

The transaction id that this corresponds to.

ns_eventq_write_event

An event is committed to being written. Fields:

dword0 unsigned int

Double word 0 of event.

dword1 unsigned int

Double word 1 of event.

dword2 unsigned int

Double word 2 of event.

dword3 unsigned int

Double word 3 of event.

event enum

Event code that was thrown away.

prod unsigned int

The PROD pointer (without wrap) that this event will be written to.

stall_event bool

Describes an event for a stalled transaction.

trans_id unsigned int

Transaction ID that generated this event.

ns_eventq_write_event_visible_in_memory

An event is now visible in memory. Fields:

abort bool

The write aborted.

dword0 unsigned int

Double word 0 of event.

dword1 unsigned int

Double word 1 of event.

dword2 unsigned int

Double word 2 of event.

dword3 unsigned int

Double word 3 of event.

event enum

Event code that was thrown away.

prod unsigned int

The PROD pointer (without wrap) that this event will be written to.

stall_event bool

Describes an event for a stalled transaction.

trans_id unsigned int

Transaction ID that generated this event.

ns_gerror_inverted

A GERROR bit was attempted to be inverted to record a fault. Fields:

already_different_to_gerrorn bool

True if the bit was already different to the corresponding bit in GERRORN and so it was not inverted.

gerrorn unsigned int

GERRORN register value.

interrupt_potentially_sent bool

An interrupt is potentially sent, depending on IRQEN.
GERROR(N).MSI_GERROR_ABT_ERR does not send an interrupt.

new_gerror unsigned int

GERROR register value after any inversion.

which enum

Which bit was attempted to be inverted.

ns_gerrorn_acknowledge

A GERROR was acknowledged by SW. Fields:

which enum

Which GERROR was acknowledged.

pmcg_irq_config

The interrupt configuration of the Performance Monitor Counter Group (PMCG) changed.
Fields:

address unsigned int

Address of the MSI.

data unsigned int

Data payload of the MSI.

irqen bool

SMMU_PMCG_CTRL.IRQEN.

memattr enum

Memory type.

mpam_ns bool

The NS state of the MPAM PARTID and MPAM PMG.

mpam_pmg_and_partid unsigned int

Bits[23:16] are the MPAM PMG. Bits[15:0] are the physical PARTID.

msi_supported bool

Are MSIs supported for this security world.

ns bool

Non-secure bus attribute.

number_of_interrupts_in_flight unsigned int

The number of interrupts that have been committed to be produced or in flight.

pmcg_index unsigned int

Index of the PMCG.

sh enum

Shareability.

smmu_pmcg_gmpam_Update bool

The SMMU_PMCG_GMPAM.Update flag. Only when this is zero are writes predictable.

ssd_ns bool

True if this PMCG controlled by the non-secure world.

pmcg_irqen

A trace of SMMU_PMCG_IRQ_CTRL.IRQEN. Fields:

new_value bool

The new value of IRQEN.

pmcg_index unsigned int

Index of the PMCG.

ssd_ns bool

True if this PMCG controlled by the non-secure world.

pmcg_irqen_ack

A trace of SMMU_PMCG_IRQ_CTRLACK.IRQEN. Fields:

new_value bool

The new value of IRQEN.

pmcg_index unsigned int

Index of the PMCG.

ssd_ns bool

True if this PMCG controlled by the non-secure world.

pmcg_merging_interrupts

An interrupt was wanted to be generated, but one was already pending so the two were merged together. Fields:

pmcg_index unsigned int

Index of the PMCG.

ssd_ns bool

True if this PMCG controlled by the non-secure world.

pmcg_trigger

A PMCG counter has been triggered. Fields:

NoStreamID bool

True if the transaction is a NoStreamID transaction.

counter_index unsigned int

The index of the counter within the PMCG.

event_id unsigned int

The event id that has been triggered.

ns_event bool

Is the event associated with non-secure state.

pmcg_index unsigned int

Index of the PMCG.

prior_counter_value unsigned int

The Counter value *before* the event has incremented it.

ssd enum

The security state associated with the event.

streamid unsigned int

The StreamID associated with the event, if there is one.

tbu_index unsigned int

The TBU index of the transaction, or ~0u if not applicable.

pmu_active_counter

Traces what active counters are in a PMCG and what StreamIDs it might filter on. Those counters that trace StreamIDs for multiple security states, or those that are not filtered by StreamID, will appear multiple times, once for each security state. All active counters for a PMCG are traced one after another. Fields:

NoStreamID bool

True if NoStreamID transactions will be traced.

begin_streamid unsigned int

The start StreamID to filter on.

counter_index unsigned int

The counter index within the PMCG.

end_incl_streamid unsigned int

The end inclusive StreamID to filter on.

evcnt unsigned int

The current count.

event_id unsigned int

The event ID to filter.

ns bool

Are the StreamIDs non-secure?.

pmcg_index unsigned int

The index of the PMCG that this counter belongs to.

ssd enum

SSD of the StreamID.

tbu_index_to_match unsigned int

The TBU index that must match, or ~0u if no matching applicable.

pmu_all_counters_in_pmcg_became_inactive

The PMCG was tracing some events and now is not tracing any. Fields:

pmcg_index unsigned int

The index of the PMCG.

pmu_capture

For some reason, a capture event occurred. Fields:

pmcg_index unsigned int

The index of the PMCG that the capture occurred on.

why enum

Why did the capture occur?.

pmu_counter_configured_to_use_unsupported_event

An enabled counter was configured to use a unsupported event. Fields:

counter_index unsigned int

The counter index within the PMCG.

event_id unsigned int

The unsupported event id.

pmcg_index unsigned int

The index of the PMCG that this counter belongs to.

pmu_counter_overflowed

A counter in a particular PMCG overflowed. Fields:

already_overflowed bool

True if the overflow flag was already set.

capture bool

True if it captured the other counter values.

counter_index unsigned int

The counter index within the PMCG.

interrupt bool

True if going to attempt to generate an interrupt.

interrupt_action enum

The interrupt action that is going to occur.

pmcg_index unsigned int

The index of the PMCG that this counter belongs to.

ssd_ns bool

The PMCG is controlled by the Non-secure security state.

priq_auto_response

A PRIQ auto response is generated. Fields:

prgindex unsigned int

The PRG Index of the response.

resp enum

The response to send.

streamid unsigned int

The StreamID of the response.

substreamid unsigned int

The PASID/SubstreamID attached to the response, or ~0u if there is none.

trans_id unsigned int

The ID of the PPR that is being lost.

why enum

Why the PPR is being lost.

priq_lost_ppr

The PRIQ is not reporting a PPR into the queue. It may or may not generate an auto-response. Fields:

is_stop_marker bool

The PPR being lost is a Stop Marker.

resp enum

What, if any, auto-response will be.

trans_id unsigned int

The ID of the PPR that is being lost.

why enum

Why the PPR is being lost.

priq_overflow_acking

Indicates that an overflow condition was acknowledged by writing to:- SMMU_PRIQ_CONS.OVACKFLG. Fields:

new_ovackflg bool

The new value of the SMMU_PRIQ_CONS.OVACKFLG.

priq_overflow_asserting

Indicates that we are toggling the SMMU_PRIQ_PROD.OVFLG because we lost a PRI request due to the PRIQ being full and an existing overflow condition does not already exist. Fields:

new_ovflg bool

The new value of the SMMU_PRIQ_PROD.OVFLG.

trans_id unsigned int

The transaction ID of the PPR that caused the overflow.

priq_received

A PRIQ Request has been received and is queued waiting processing. This does not mean that it has been written to the PRIQ but has been placed in a pending queue awaiting a decision about what to do about it. Fields:

L bool

The Last in the Page Request Group (PRG).

R bool

Read permission is requested.

W bool

Read permission is requested.

prgindex unsigned int

The Page Request Group (PRG) index that is being used to identify this request to the EndPoint.

stop_marker bool

If LRW == 0b100 then this indicates this is a stop marker.

streamid unsigned int

The StreamID this request is associated with.

substreamid unsigned int

The SubstreamID (PASID) this request is associated with, of ~0u if no SubstreamID.

substreamid_P bool

If has a SubstreamID (!= ~0u) then Privileged Mode Requested.

substreamid_X bool

If has a SubstreamID (!= ~0u) then Execution Requested (R must also be 1).

trans_id unsigned int

The transaction ID that identifies this request.

untranslated_address unsigned int

The Untranslated Address that this is requesting.

priq_state

The trace of various fields that indicate the state of the PRIQ. Fields:

cons_incl_wrap unsigned int

The value of SMMU_PRIQ.CON.SRD_and_RD_wrap.

number_of_pprs unsigned int

The number of PPRs as indicated by the CONS/PROD.

number_of_pprs_still_to_deal_with unsigned int

This is the number of PPRs that are currently waiting to either be written to the PRIQ, or auto-responded to.

number_of_priq_writes_in_flight unsigned int

The number of writes to the PRIQ that are currently in flight.

ovackflg bool

The OVACKFLG which if different to OVFLG is used to indicate that the PRIQ overflowed.

ovflg bool

The OVFLG which if different to OVACKFLG is used to indicate that the PRIQ overflowed.

priq_abt_err bool

There is an active SMMU_GERROR{N}.PRIQ_ABT_ERR.

priqen bool

The value of SMMU_CRO.PRIQEN. The *effective* value is 0 if SMMUEN == 0.

priqenack bool

The value of SMMU_CROACK.PRIQEN.

prod_incl_wrap unsigned int

The value of SMMU_PRIQ.CONS.WR_and_WR_wrap.

queue_disabled_due_to_prior_programming_error bool

The queue was disabled as the programmer got CONS/PROD into an inconsistent state. The model will disable the PRIQ until SW disables and re-enables the queue via SMMU_CRO.PRIQEN.

smmuen bool

The value of SMMU_CRO.SMMUEN. If this is 0 then the effective PRIQEN is 0.

smmuenack bool

The value of SMMU_CRO.SMMUEN. If this is 0 then the effective PRIQEN is 0.

table_size_in_elements unsigned int

The size of the table in the number of items it can hold.

priq_write_aborted

A PRIQ write aborted. The PRIQ now goes into an error state and will start auto-responding to PRI requests. Fields:

prod_incl_wrap unsigned int

The PROD including the Wrap bit where we were trying to write.

trans_id unsigned int

The transaction ID of the PPR that aborted.

priq_write_ok

A PRIQ write completed OK. Fields:

prod_incl_wrap unsigned int

The PROD including the Wrap bit where we wrote..

trans_id unsigned int

The transaction ID of the PPR.

priq_write_start

A PRIQ request has been received and is going to be attempt to be written to the queue.

Fields:

L bool

The Last in the Page Request Group (PRG).

R bool

Read permission is requested.

W bool

Read permission is requested.

prgindex unsigned int

The Page Request Group (PRG) index that is being used to identify this request to the EndPoint.

prod_incl_wrap unsigned int

PROD position including the wrap bit.

stop_marker bool

If LRW == 0b100 then this indicates this is a stop marker.

streamid unsigned int

The StreamID this request is associated with.

substreamid unsigned int

The SubstreamID (PASID) this request is associated with, of ~0u if no SubstreamID.

substreamid_P bool

If has a SubstreamID (!= ~0u) then Privileged Mode Requested.

substreamid_X bool

If has a SubstreamID (!= ~0u) then Execution Requested (R must also be 1).

trans_id unsigned int

The transaction ID that identifies this request.

untranslated_address unsigned int

The Untranslated Address that this is requesting.

ptw_read

Page Table Walk (read). This is the result of the physical access that the SMMU is making.

Fields:

abort enum

Non-zero if the access aborted/failed.

adomain enum

The actual attributes of the access that was used, after IMP DEF mangling.

aprot enum

The actual attributes of the access that was used, after IMP DEF mangling. PAS[1] (bit[3]), Instruction/Data (bit[2]), PAS[0] (bit[1]), Privileged/User (bit[0]).

data unsigned int

The data fetch if it didn't abort.

desckind enum

The kind of descriptor.

inner_cache enum

The actual attributes of the access that was used, after IMP DEF mangling.

input_address unsigned int

Input address this lookup is for (after TBI normalisation).

mecid unsigned int

The MECID of the TTD fetch.

ns enum

Non-Secure on the bus.

outer_cache enum

The actual attributes of the access that was used, after IMP DEF mangling.

pa_address unsigned int

Physical address.

pas enum

The PAS of the TTD fetch.

ssd enum

SSD of the transaction.

ssd_ns enum

Security state of stream.

streamid unsigned int

The StreamID this is for.

substreamid unsigned int

The SubstreamID this is for.

trans_id unsigned int

Transaction id.

ttb_grain_stage_and_level unsigned int

Which TTB used in bits[31:24]. Which Grain size used in bit[23:16]:- 0 – 4 KiB 1 – 16 KiB 2 – 64 KiB 3 – 64 KiB ARMv8.2 where descriptor bits [15:12] for bits [51:48] of

output address 4 – 4 KiB FEAT_LPA2 where address is {bits[9:8], bits[49:12]} 5 – 16 KiB FEAT_LPA2 where address is {bits[9:8], bits[49:14], 2'b00} 6 – 64 KiB ARMv8.2 where descriptor bits [15:12] are ignored Stage in bits[15:8]. Level in bits[7:0], level -1 is 0xF.

ptw_read_stl_invalid_long_descriptor

Page Table Walk (read). The descriptor that the PTW fetched is invalid. Fields:

desckind enum

The kind of descriptor.

input_address unsigned int

Input address this lookup is for (after TBI normalisation).

mecid unsigned int

The MECID of the TTD fetch.

ns enum

Non-Secure on the bus.

pa_address unsigned int

Physical address.

pas enum

The PAS of the TTD fetch.

ssd enum

SSD of the transaction.

ssd_ns enum

Security state of stream.

streamid unsigned int

The StreamID this is for.

substreamid unsigned int

The SubstreamID this is for.

trans_id unsigned int

Transaction id.

ttb_grain_stage_and_level unsigned int

Which TTB used in bits[31:24]. Which Grain size used in bit[23:16]:- 0 – 4 KiB 1 – 16 KiB 2 – 64 KiB 3 – 64 KiB ARMv8.2 where descriptor bits [15:12] for bits [51:48] of output address 4 – 4 KiB FEAT_LPA2 where address is {bits[9:8], bits[49:12]} 5 – 16 KiB FEAT_LPA2 where address is {bits[9:8], bits[49:14], 2'b00} 6 – 64 KiB ARMv8.2 where descriptor bits [15:12] are ignored Stage in bits[15:8]. Level in bits[7:0], level -1 is 0xF.

ptw_read_stl_leaf_long_descriptor

Page Table Walk (read). The descriptor that the PTW fetched is a block or page and this is the decode. Fields:

AF bool

Access Flag.

AMEC enum

Use Primary or Alternative MECID.

AP21 enum

The access permissions.

AttrIndx210 unsigned int

The attribute index into the MAIR0/1. If AIE is implemented then this is the full index AttrIndx[3:0].

DBM bool

Dirty Bit Modifier. This only has meaning if running in AArch64 mode.

NS bool

The encoding is for non-secure if this is a secure fetch.

PIE_Dirty enum

The Dirty bit if S2PIE is in use.

PIE_PIIIndex unsigned int

The S2PIIndex if S2PIE is in use, or 0xFFFF if not.

PXN bool

Privileged eXecute Never.

Protected enum

Is the descriptor producing an AssuredTranslation.

SH10 enum

The shareability.

XN bool

eXecute Never.

contiguous bool

Contiguous hint.

desckind enum

The kind of descriptor.

hwu_pbha unsigned int

Top four bits are appropriate CD.HWU, *bottom bits[62:59] of descriptor. Page Based Hardware Attributes: only valid on a bit-per-bit basis enabled by CD.HWU0/CD.HWU1**.

input_address unsigned int

Input address this lookup is for (after TBI normalisation).

mecid unsigned int

The MECID of the TTD fetch.

nG bool

not Global.

nT bool

SMMUv3.2: If true then do not cache this entry in such a way that it can cause a TLB conflict abort (F_TLB_CONFLICT). The entry must produce a consistent result.

ns enum

Non-Secure on the bus.

output_address unsigned int

Output address.

pa_address unsigned int

Physical address.

pas enum

The PAS of the TTD fetch.

ssd enum

SSD of the transaction.

ssd_ns enum

Security state of stream.

streamid unsigned int

The StreamID this is for.

substreamid unsigned int

The SubstreamID this is for.

trans_id unsigned int

Transaction id.

ttb_grain_stage_and_level unsigned int

Which TTB used in bits[31:24]. Which Grain size used in bit[23:16]:- 0 – 4 KiB 1 – 16 KiB 2 – 64 KiB 3 – 64 KiB ARMv8.2 where descriptor bits [15:12] for bits [51:48] of output address 4 – 4 KiB FEAT_LPA2 where address is {bits[9:8], bits[49:12]} 5 – 16 KiB FEAT_LPA2 where address is {bits[9:8], bits[49:14], 2'b00} 6 – 64 KiB ARMv8.2 where descriptor bits [15:12] are ignored Stage in bits[15:8]. Level in bits[7:0], level -1 is 0xF.

ptw_read_st1_table_long_descriptor

Page Table Walk (read). The descriptor that the PTW fetched is a Table, this decodes the fields. Fields:

AF enum

.

APTable enum

Remove permissions independently of subsequent descriptors.

NSTable bool

The next level table descriptor is forced to non-secure.

PXNTable bool

Force PXN independently of subsequent descriptors.

Protected enum

Is the descriptor capable of producing an AssuredTranslation.

TableAddress unsigned int

Address of the next table.

XNTable bool

Force XN independently of subsequent descriptors.

desckind enum

The kind of descriptor.

input_address unsigned int

Input address this lookup is for (after TBI normalisation).

mecid unsigned int

The MECID of the TTD fetch.

ns enum

Non-Secure on the bus.

pa_address unsigned int

Physical address.

pas enum

The PAS of the TTD fetch.

ssd enum

SSD of the transaction.

ssd_ns enum

Security state of stream.

streamid unsigned int

The StreamID this is for.

substreamid unsigned int

The SubstreamID this is for.

trans_id unsigned int

Transaction id.

ttb_grain_stage_and_level unsigned int

Which TTB used in bits[31:24]. Which Grain size used in bit[23:16]:- 0 – 4 KiB 1 – 16 KiB 2 – 64 KiB 3 – 64 KiB ARMv8.2 where descriptor bits [15:12] for bits [51:48] of output address 4 – 4 KiB FEAT_LPA2 where address is {bits[9:8], bits[49:12]} 5 – 16 KiB FEAT_LPA2 where address is {bits[9:8], bits[49:14], 2'b00} 6 – 64 KiB ARMv8.2 where descriptor bits [15:12] are ignored Stage in bits[15:8]. Level in bits[7:0], level -1 is 0xF.

ptw_read_st2_invalid_descriptor

Page Table Walk (read). The descriptor that the PTW fetched is invalid. Fields:

desckind enum

The kind of descriptor.

input_address unsigned int

Input address this lookup is for (after TBI normalisation).

mecid unsigned int

The MECID of the TTD fetch.

ns enum

Non-Secure on the bus.

pa_address unsigned int

Physical address.

pas enum

The PAS of the TTD fetch.

ssd enum

SSD of the transaction.

ssd_ns enum

Security state of stream.

streamid unsigned int

The StreamID this is for.

substreamid unsigned int

The SubstreamID this is for.

trans_id unsigned int

Transaction id.

ttb_grain_stage_and_level unsigned int

Which TTB used in bits[31:24]. Which Grain size used in bit[23:16]:- 0 – 4 KiB 1 – 16 KiB 2 – 64 KiB 3 – 64 KiB ARMv8.2 where descriptor bits [15:12] for bits [51:48] of output address 4 – 4 KiB FEAT_LPA2 where address is {bits[9:8], bits[49:12]} 5 – 16 KiB FEAT_LPA2 where address is {bits[9:8], bits[49:14], 2'b00} 6 – 64 KiB ARMv8.2 where descriptor bits [15:12] are ignored Stage in bits[15:8]. Level in bits[7:0], level -1 is 0xF.

ptw_read_st2_leaf_descriptor

Page Table Walk (read). The descriptor that the PTW fetched is a block or page and this is the decode. Fields:

AF bool

Access Flag.

AMEC enum

Use Primary or Alternative MECID.

AssuredOnly enum

The descriptor is marked as AssuredOnly.

DBM bool

Dirty Bit Modifier. This only has meaning if running in AArch64 mode.

HAP21 enum

The access permissions.

MemAttr3_0 enum

The memory attributes.

NS enum

Whether this descriptor forces NS.

PIE_Dirty enum

The Dirty bit if S2PIE is in use.

PIE_PIIIndex unsigned int

The S2PIIndex if S2PIE is in use, or 0xFFFF if not.

POE_POIndex unsigned int

The S2POIndex if S2POE is in use, or 0xFFFF if not.

SH10 enum

The shareability.

XN enum

eXecute Never.

contiguous bool

Contiguous hint.

desckind enum

The kind of descriptor.

input_address unsigned int

Input address this lookup is for (after TBI normalisation).

mecid unsigned int

The MECID of the TTD fetch.

nT bool

SMMUv3.2: If true then do not cache this entry in such a way that it can cause a TLB conflict abort (F_TLB_CONFLICT). The entry must produce a consistent result.

ns enum

Non-Secure on the bus.

output_address unsigned int

Output address.

pa_address unsigned int

Physical address.

pas enum

The PAS of the TTD fetch.

s2hwu_pbha unsigned int

Top four bits are STE.S2HWU, bottom bits[62:59] of descriptor. Page Based Hardware
Attributes: only valid on a bit-per-bit basis enabled by STE.S2HWU.

ssd enum

SSD of the transaction.

ssd_ns enum

Security state of stream.

streamid unsigned int

The StreamID this is for.

substreamid unsigned int

The SubstreamID this is for.

trans_id unsigned int

Transaction id.

tth_grain_stage_and_level unsigned int

Which TTB used in bits[31:24]. Which Grain size used in bit[23:16]:- 0 – 4 KiB 1 – 16 KiB 2 – 64 KiB 3 – 64 KiB ARMv8.2 where descriptor bits [15:12] for bits [51:48] of output address 4 – 4 KiB FEAT_LPA2 where address is {bits[9:8], bits[49:12]} 5 – 16 KiB FEAT_LPA2 where address is {bits[9:8], bits[49:14], 2'b00} 6 – 64 KiB ARMv8.2 where descriptor bits [15:12] are ignored Stage in bits[15:8]. Level in bits[7:0], level -1 is 0xF.

ptw_read_st2_table_descriptor

Page Table Walk (read). The descriptor that the PTW fetched is a Table, this decodes the fields. Fields:

AF enum

.

APTable enum

Remove permissions independently of subsequent descriptors.

PXNTable bool

Force PXN independently of subsequent descriptors.

TableAddress unsigned int

Address of the next table.

XNTable bool

Force XN independently of subsequent descriptors.

desckind enum

The kind of descriptor.

input_address unsigned int

Input address this lookup is for (after TBI normalisation).

mecid unsigned int

The MECID of the TTD fetch.

ns enum

Non-Secure on the bus.

pa_address unsigned int

Physical address.

pas enum

The PAS of the TTD fetch.

ssd enum

SSD of the transaction.

ssd_ns enum

Security state of stream.

streamid unsigned int

The StreamID this is for.

substreamid unsigned int

The SubstreamID this is for.

trans_id unsigned int

Transaction id.

ttb_grain_stage_and_level unsigned int

Which TTB used in bits[31:24]. Which Grain size used in bit[23:16]:- 0 – 4 KiB 1 – 16 KiB 2 – 64 KiB 3 – 64 KiB ARMv8.2 where descriptor bits [15:12] for bits [51:48] of output address 4 – 4 KiB FEAT_LPA2 where address is {bits[9:8], bits[49:12]} 5 – 16 KiB FEAT_LPA2 where address is {bits[9:8], bits[49:14], 2'b00} 6 – 64 KiB ARMv8.2 where descriptor bits [15:12] are ignored Stage in bits[15:8]. Level in bits[7:0], level -1 is 0xF.

raw_register_end_read

The raw register read transaction. This is the transaction as directed to the register port.
Fields:

address unsigned int

The address of the transaction.

id unsigned int

Id of this transaction.

ok bool

Was the access OK? The bus response will always be OK, but is the register access reasonable.

pas enum

The PAS of the transaction.

read_data unsigned int

The data read.

raw_register_end_write

The raw register write transaction. This is the transaction as directed to the register port.
Fields:

address unsigned int

The address of the transaction.

id unsigned int

Id of this transaction.

ok bool

Was the access OK? The bus response will always be OK, but is the register access reasonable.

pas enum

The PAS of the transaction.

raw_register_start_read

The raw register read transaction. This is the transaction as directed to the register port.
Fields:

address unsigned int

Address of transaction.

id unsigned int

Id of this transaction. Top bit is set if is a debug transaction.

ns enum

Is the transaction non-secure?.

number_of_beats unsigned int

Number of beats.

pas enum

The PAS of the transaction.

size_of_beat_in_bytes unsigned int

Size of each beat in bytes.

raw_register_start_write

The raw register write transaction. This is the transaction as directed to the register port.
Fields:

address unsigned int

Address of transaction.

id unsigned int

Id of this transaction. Top bit is set if is a debug transaction.

ns enum

Is the transaction non-secure?.

number_of_beats unsigned int

Number of beats.

pas enum

The PAS of the transaction.

size_of_beat_in_bytes unsigned int

Size of each beat in bytes.

write_data unsigned int

The data to write.

register_disallowed_read_string

A text representation of the read of a register that was disallowed. Fields:

out string

The text description of the register value read.

register_disallowed_write_string

A text representation of the write of a register write that was disallowed. Fields:

in string

The text description of the register value written.

register_read_reserved

A text representation of an access to a register address that is reserved. Fields:

in string

The text description of the register value.

register_read_string

A text representation of the read of a register. Fields:

out string

The text description of the register value read.

register_write_reserved

A text representation of an access to a register address that is reserved or a write to a **RES0** field in a register. Fields:

in string

The text description of the register value.

register_write_string

A text representation of the write of a register. Fields:

in string

The text description of the register value written.

rl_priq_auto_response

A PRIQ auto response is generated. Fields:

prgindex unsigned int

The PRG Index of the response.

resp enum

The response to send.

streamid unsigned int

The StreamID of the response.

substreamid unsigned int

The PASID/SubstreamID attached to the response, or ~0u if there is none.

trans_id unsigned int

The ID of the PPR that is being lost.

why enum

Why the PPR is being lost.

rl_priq_lost_ppr

The PRIQ is not reporting a PPR into the queue. It may or may not generate an auto-response. Fields:

is_stop_marker bool

The PPR being lost is a Stop Marker.

resp enum

What, if any, auto-response will be.

trans_id unsigned int

The ID of the PPR that is being lost.

why enum

Why the PPR is being lost.

rl_priq_received

A PRIQ Request has been received and is queued waiting processing. This does not mean that it has been written to the PRIQ but has been placed in a pending queue awaiting a decision about what to do about it. Fields:

L bool

The Last in the Page Request Group (PRG).

R bool

Read permission is requested.

W bool

Read permission is requested.

prgindex unsigned int

The Page Request Group (PRG) index that is being used to identify this request to the EndPoint.

stop_marker bool

If LRW == 0b100 then this indicates this is a stop marker.

streamid unsigned int

The StreamID this request is associated with.

substreamid unsigned int

The SubstreamID (PASID) this request is associated with, of ~0u if no SubstreamID.

substreamid_P bool

If has a SubstreamID (!= ~0u) then Privileged Mode Requested.

substreamid_X bool

If has a SubstreamID (!= ~0u) then Execution Requested (R must also be 1).

trans_id unsigned int

The transaction ID that identifies this request.

untranslated_address unsigned int

The Untranslated Address that this is requesting.

s_cmd_sync_completed_irq

“Secure” CMD_SYNC completed its IRQ action. Fields:

cmd_id unsigned int

The ID of the command being issued.

cons_incl_wrap unsigned int

CONS pointer of CMD_SYNC.

is_synthetic bool

This is a synthetic sync. The cons_incl_wrap is 0xFFFF'ffff.

s_cmd_sync_completed_no_action

Secure CMD_SYNC completed, there was no CMD_SYNC action (SIG_NONE). Fields:

cmd_id unsigned int

The ID of the command being issued.

cons_incl_wrap unsigned int

CONS pointer of CMD_SYNC.

is_synthetic bool

This is a synthetic sync. The cons_incl_wrap is 0xFFFF'ffff.

s_cmd_sync_completed_sev

Secure CMD_SYNC completed its action of SIG_SEV. Fields:

cmd_id unsigned int

The ID of the command being issued.

cons_incl_wrap unsigned int

CONS pointer of CMD_SYNC.

is_synthetic bool

This is a synthetic sync. The cons_incl_wrap is 0xFFFF'ffff.

s_cmd_sync_completed_with_error

Secure CMD_SYNC completed, however, there was an error associated with the completion of the CMD_SYNC. Fields:

error enum

The error associated with the CMD_SYNC.

cmd_id unsigned int

The ID of the command being issued.

cons_incl_wrap unsigned int

CONS pointer of CMD_SYNC.

is_synthetic bool

This is a synthetic sync. The cons_incl_wrap is 0xFFFF'ffff.

s_cmd_sync_issued

Secure CMD_SYNC has now completed issuing and we now know if something is hazarding the CMD_SYNC from completing. Fields:

cmd_id unsigned int

The ID of the command being issued.

cons_incl_wrap unsigned int

CONS pointer of CMD_SYNC.

hazarded bool

The CMD_SYNC is currently hazarded from completing.

is_synthetic bool

This is a synthetic sync. The cons_incl_wrap is 0xFFFF'ffff.

s_cmd_sync_issuing

Secure CMD_SYNC is starting to issue, we are going to start accounting for which things have to be part of the prior set of things that have to complete before the CMD_SYNC can complete. Fields:

cmd_id unsigned int

The ID of the command being issued.

cons_incl_wrap unsigned int

CONS pointer of CMD_SYNC.

is_synthetic bool

This is a synthetic sync. The cons_incl_wrap is 0xFFFF'ffff.

s_cmd_sync_starting_completion_action_irq

Secure CMD_SYNC starting IRQ completion action. Fields:

cmd_id unsigned int

The ID of the command being issued.

cons_incl_wrap unsigned int

CONS pointer of CMD_SYNC.

is_synthetic bool

This is a synthetic sync. The cons_incl_wrap is 0xFFFF'ffff.

msiaddress unsigned int

The MSI address to use (or 0 if not sending an MSI).

msidata unsigned int

The MSI data to send.

s_cmd_sync_starting_completion_action_sev

Secure CMD_SYNC starting its completion action of SIG_SEV. Fields:

cmd_id unsigned int

The ID of the command being issued.

cons_incl_wrap unsigned int

CONS pointer of CMD_SYNC.

is_synthetic bool

This is a synthetic sync. The cons_incl_wrap is 0xFFFF'ffff.

s_cmdq_cmd_sync_error

The CMD_SYNC can fault if an ATC Invalidate timeout occurs. The timeout will be attached to the next CMD_SYNC on the queue on which the invalidate was launched. Fields:

cerror enum

The CERROR to attach.

cmd_queue unsigned int

The top bits indicate the CMDQ id. These bits are attached to all cmd_ids generated for this queue. The bits [63:44] contain the queue ID. [63:62] 0: s-ssd, 1: reserved, 2: ns-ssd. 3: rl-ssd [61] is ECMDQ [59:52] QCP index [51:44] ECMDQ index in QCP.

herror enum

The HERROR to attach, or HERROR_NONE if we are attaching a CERROR.

s_cmdq_cmd_sync_error_delivered

The CMD_SYNC can fault if an ATC Invalidate timeout occurs. This is the CMD_SYNC delivering that fault. Fields:

cerror enum

The CERROR to attach.

cmd_id unsigned int

Command ID of the CMD_SYNC.

s_cmdq_ecmdq_enack

Trace the changes to ENACK. Fields:

error enum

The error being exposed, if any.

cmd_queue unsigned int

The top bits indicate the CMDQ id. These bits are attached to all cmd_ids generated for this queue. The bits [63:44] contain the queue ID. [63:62] 0: s-ssd, 1: reserved, 2: ns-ssd. 3: rl-ssd [61] is ECMDQ [59:52] QCP index [51:44] ECMDQ index in QCP.

enack bool

The value of ENACK.

forced_error bool

The queue was enabled with ERR != ERRACK and so the implementation chooses to trigger the error reporting mechanism.

s_cmdq_issue

A command is actually being executed. Fields:

cmd_id unsigned int

Command ID of the command being executed.

cons unsigned int

CONS of the command.

what string

What is being executed.

s_cmdq_pointers_state

The command queue pointers. Fields:

cmd_queue unsigned int

The top bits indicate the CMDQ id. These bits are attached to all cmd_ids generated for this queue. The bits [63:44] contain the queue ID. [63:62] 0: s-ssd, 1: reserved, 2: ns-ssd. 3: rl-ssd [61] is ECMDQ [59:52] QCP index [51:44] ECMDQ index in QCP.

disable_fetch bool

The model has decided that as you programmed PROD/CONS inconsistently that it would not fetch as it has no confidence as to which are commands and which are not.

q_issuable_and_issued_cons unsigned int

The CONS pointer of the region that is issuable and/or issued (including wrap).

q_issuable_and_issued_prod unsigned int

The PROD pointer of the region that is issuable and/or issued (including wrap).

q_programmer_cons unsigned int

The CONS pointer as seen by the programmer (including wrap).

q_programmer_prod unsigned int

The PROD pointer as seen by the programmer (including wrap).

q_unfetched_cons unsigned int

The CONS pointer of the region unfetched (including wrap).

q_unfetched_prod unsigned int

The PROD pointer of the region unfetched (including wrap).

s_cmdq_state

A description of what the CMDQ can do now. Fields:

can_cerror bool

There is a CERROR that can be recognised.

can_cmd_sync bool

There is a CMD_SYNC that can be recognised.

can_fetch bool

The SMMU can fetch.

can_herror bool

There is an HERROR that can be recognised.

can_issue bool

There are fetched commands that are eligible to be issued.

can_update_cons bool

The programmer view of CMDQ_CONS can be moved to say some have been consumed.

cmd_queue unsigned int

The top bits indicate the CMDQ id. These bits are attached to all cmd_ids generated for this queue. The bits [63:44] contain the queue ID. [63:62] 0: s-ssd, 1: reserved, 2: ns-ssd, 3: rl-ssd [61] is ECMDQ [59:52] QCP index [51:44] ECMDQ index in QCP.

cmd_sync enum

Why a CMD_SYNC cannot be issued.

cmdqen bool

CMDQEN, when enabled then the CMDQ can fetch commands.

cmdqen_1_to_0_unacknowledged bool

CMDQEN has been set to 0, but we have yet to acknowledge it.

current-real-cerror enum

The current CERROR that would be visible to the programmer if there was an error.

fetch-disabled_due_to_programmer_error bool

The PROD/CONS pointer was inconsistent and the model decided to stop processing commands.

fetch-number_of_unfetched_commands unsigned int

The number of commands that have yet to be fetched from memory, or ~0u if CMDQEN == 0.

fetch-state enum

The state of the fetch state machine.

fetch-too_many_commands_outstanding_so_not_fetching_any_more bool

If there are too many issued commands, or pending to be issued then we won't fetch any more.

issue-number_of_issuable_commands unsigned int

The number of commands that have been fetched but not yet issued.

issue-number_of_unissuable_commands unsigned int

This might include commands behind an illegal command, or behind a CMD_SYNC. Under some circumstances this can include a command representing an external abort.

issue-state enum

The internal issued state machine state.

outstanding_fetches_since_reset unsigned int

The number of outstanding fetches.

synthetic_sync_can_be_completed bool

The synthetic sync that is outstanding can be completed.

waiting_for_synthetic_sync bool

There is a synthetic sync that is outstanding.

s_eventq_adding_to_cmd_sync_prior_set

The event created by the specified transaction id must become visible to the programmer before the CMD_SYNC can complete. Fields:

trans_id unsigned int

The transaction id that caused the event.

s_eventq_auto_retry_stalled_transaction

A transaction that was stalled but not yet reported on the Event Queue was retried. The retry can happen if the transaction didn't write because it was de-duplicated behind a reported stalling transaction and that stalling transaction was CMD_RESUMEd, or if the queue was unwritable and the queue is now writable. Fields:

trans_id unsigned int

Transaction id of the auto-retried transaction.

why enum

Why the transaction retried.

s_eventq_cmd_sync_unhazarded

The CMD_SYNC has been unhazarded as the prior set is empty and as much of it as required is visible in the programmers' view. Fields:

prod_incl_wrap unsigned int

The prod index including wrap bit that was required to become visible.

prog_prod_incl_wrap unsigned int

The current programmer-visible prod including wrap bit.

s_eventq_eventqen_and_eventqenack

Trace the values of EVENTQEN and EVENTQENACK as they change. Fields:

about enum

Some more information about the queue state.

cons_incl_wrap unsigned int

The CONS pointer (including wrap bits).

eventqen bool

EVENTQEN.

eventqenack bool

EVENTQENACK.

prod_incl_wrap unsigned int

The PROD pointer (including wrap bits).

s_eventq_external_abort

A particular eventq record got an external abort. The event will subsequently appear in the ns/s_eventq_lost_event_records trace. Fields:

prod unsigned int

The prod pointer (excluding wrap bit) of the event record that aborted.

trans_id unsigned int

The transaction id that produced the event record that aborted.

s_eventq_lost_event_records

Event records that are lost and never record appear on this trace. Fields:

dword0 unsigned int

Double word 0 of event.

dword1 unsigned int

Double word 1 of event.

dword2 unsigned int

Double word 2 of event.

dword3 unsigned int

Double word 3 of event.

event enum

Event code that was thrown away.

prod unsigned int

The PROD index of the event if appropriate or ~0u otherwise.

reason enum

The reason the record was lost.

trans_id unsigned int

The transaction id that this corresponds to.

s_eventq_overflow_acknowledged

When events are lost then they appear on the eventq_lost_event_records_* trace. This will activate the overflow flag in the corresponding SMMU_s_EVENTQ_PROD register. Software can then acknowledge that flag using the overflow acknowledge flag in SMMU_s_EVENTQ_CONS. Fields:

bad bool

If true then the programmer has attempted to acknowledge an overflow condition that didn't exist.

count_of_events_lost_due_to_overflow unsigned int

This is the number of events that were lost because the event queue was full. This does not include any lost because of an EVENTQ_ABT_ERR.

s_eventq_pending_event_records

Event records that are pending to be dealt with by the eventq state machine. A specific event can be traced multiple times if it becomes part of the prior set of transactions of a CMD_SYNC that has to complete before the CMD_SYNC is allowed to be completed. Fields:

dword0 unsigned int

Double word 0 of event.

dword1 unsigned int

Double word 1 of event.

dword2 unsigned int

Double word 2 of event.

dword3 unsigned int

Double word 3 of event.

event enum

Event code.

is_prior_set bool

The transaction forms part of the prior set of a CMD_SYNC that have to become visible (or thrown away if necessary) before the CMD_SYNC can complete.

trans_id unsigned int

The transaction id that this corresponds to.

s_eventq_prod_cons_updated

This records when the programmer visible PROD/CONS pointer is updated. More can be visible in memory than are advertised in the programmer-visible PROD pointer. This also traces the latest PROD that could have been advertised. Fields:

mem_prod unsigned int

The PROD pointer value for all records that are visible in the memory.

num_events_available unsigned int

The number of events available to the programmer.

prog_cons unsigned int

The programmer-visible CONS pointer.

prog_prod unsigned int

The programmer-visible PROD pointer.

what enum

What changed?.

s_eventq_raising_eventq_abt_err

EVENTQ_ABT_ERR can now be raised.

s_eventq_removing_from_cmd_sync_prior_set

The event captured by a CMD_SYNC and created by the specified transaction id is now in memory. Fields:

prod unsigned int

The prod index (no wrap bit) that must become visible to the programmer to see this record. If not head_of_line then the record will not become visible and this field is meaningless.

reason enum

The reason the event was removed.

trans_id unsigned int

The transaction id that caused the event.

s_eventq_stashing_unreported_stalled_transaction_for_auto_retry

A stall event record would have been lost, instead it is stashed away to auto-retry when it can. Fields:

dword0 unsigned int

Double word 0 of event.

dword1 unsigned int

Double word 1 of event.

dword2 unsigned int

Double word 2 of event.

dword3 unsigned int

Double word 3 of event.

event enum

Event code that was thrown away.

reason enum

The reason the record was lost.

trans_id unsigned int

The transaction id that this corresponds to.

s_eventq_write_event

An event is committed to being written. Fields:

dword0 unsigned int

Double word 0 of event.

dword1 unsigned int

Double word 1 of event.

dword2 unsigned int

Double word 2 of event.

dword3 unsigned int

Double word 3 of event.

event enum

Event code that was thrown away.

prod unsigned int

The PROD pointer (without wrap) that this event will be written to.

stall_event bool

Describes an event for a stalled transaction.

trans_id unsigned int

Transaction ID that generated this event.

s_eventq_write_event_visible_in_memory

An event is now visible in memory. Fields:

abort bool

The write aborted.

dword0 unsigned int

Double word 0 of event.

dword1 unsigned int

Double word 1 of event.

dword2 unsigned int

Double word 2 of event.

dword3 unsigned int

Double word 3 of event.

event enum

Event code that was thrown away.

prod unsigned int

The PROD pointer (without wrap) that this event will be written to.

stall_event bool

Describes an event for a stalled transaction.

trans_id unsigned int

Transaction ID that generated this event.

s_gerror_inverted

A GERROR bit was attempted to be inverted to record a fault. Fields:

already_different_to_gerrorn bool

True if the bit was already different to the corresponding bit in GERRORN and so it was not inverted.

gerrorn unsigned int

GERRORN register value.

interrupt_potentially_sent bool

An interrupt is potentially sent, depending on IRQEN.
GERROR(N).MSI_GERROR_ABT_ERR does not send an interrupt.

new_gerror unsigned int

GERROR register value after any inversion.

which enum

Which bit was attempted to be inverted.

s_gerrorn_acknowledge

A GERROR was acknowledged by SW. Fields:

which enum

Which GERROR was acknowledged.

s_priq_auto_response

A PRIQ auto response is generated. Fields:

prgindex unsigned int

The PRG Index of the response.

resp enum

The response to send.

streamid unsigned int

The StreamID of the response.

substreamid unsigned int

The PASID/SubstreamID attached to the response, or ~0u if there is none.

trans_id unsigned int

The ID of the PPR that is being lost.

why enum

Why the PPR is being lost.

s_priq_lost_ppr

The PRIQ is not reporting a PPR into the queue. It may or may not generate an auto-response. Fields:

is_stop_marker bool

The PPR being lost is a Stop Marker.

resp enum

What, if any, auto-response will be.

trans_id unsigned int

The ID of the PPR that is being lost.

why enum

Why the PPR is being lost.

s_priq_received

A PRIQ Request has been received and is queued waiting processing. This does not mean that it has been written to the PRIQ but has been placed in a pending queue awaiting a decision about what to do about it. Fields:

L bool

The Last in the Page Request Group (PRG).

R bool

Read permission is requested.

W bool

Read permission is requested.

prgindex unsigned int

The Page Request Group (PRG) index that is being used to identify this request to the EndPoint.

stop_marker bool

If LRW == 0b100 then this indicates this is a stop marker.

streamid unsigned int

The StreamID this request is associated with.

substreamid unsigned int

The SubstreamID (PASID) this request is associated with, of ~0u if no SubstreamID.

substreamid_P bool

If has a SubstreamID (!= ~0u) then Privileged Mode Requested.

substreamid_X bool

If has a SubstreamID (!= ~0u) then Execution Requested (R must also be 1).

trans_id unsigned int

The transaction ID that identifies this request.

untranslated_address unsigned int

The Untranslated Address that this is requesting.

sec_override

Controls non-secure accesses to some registers. Fields:

value bool

The value of the signal.

sev

Send a SEV. Fields:

why enum

Why the SEV was generated.

smmu_atc_inv

The CMD_ATC_INV command is sent. Fields:

address unsigned int

Untranslated Address to invalidate.

cmd_id unsigned int

Command id.

cons unsigned int

CONS of the command.

global bool

Global flag.

log2_size_in_bytes unsigned int

Log 2 of the size in bytes of the region to invalidate.

ns_number_in_flight unsigned int

Number of ATC invalidates in flight issued from the Non-secure CMDQ.

ns_number_waiting_to_go_out unsigned int

The total number of ATC invalidates waiting to be pushed out from the Non-secure CMDQ.

rl_number_in_flight unsigned int

Number of ATC invalidates in flight issued from the Realm CMDQ.

rl_number_waiting_to_go_out unsigned int

The total number of ATC invalidates waiting to be pushed out from the Realm CMDQ.

s_number_in_flight unsigned int

Number of ATC invalidates in flight issued from the Secure CMDQ.

s_number_waiting_to_go_out unsigned int

The total number of ATC invalidates waiting to be pushed out from the Secure CMDQ.

ssd_of_queue enum

The SSD of the CMDQ.

ssd_of_streamid enum

The SSD of the StreamID.

streamid unsigned int

StreamID in the invalidate command.

substreamid unsigned int

SubstreamID or ~0u if SSValid == false.

smmu_atc_inv_completed

The CMD_ATC_INV command completed. Fields:

cmd_id unsigned int

Command id, if top-bit is set then was issued from the Non-secure CMDQ.

ns_number_in_flight unsigned int

Number of ATC invalidates in flight now this one has completed, that were issued by the Non-secure CMDQ.

ns_number_waiting_to_go_out unsigned int

The total number of ATC invalidates waiting to be pushed out from the Non-secure CMDQ.

response enum

The response.

rl_number_in_flight unsigned int

Number of ATC invalidates in flight now this one has completed, that were issued by the Realm CMDQ.

rl_number_waiting_to_go_out unsigned int

The total number of ATC invalidates waiting to be pushed out from the Realm CMDQ.

s_number_in_flight unsigned int

Number of ATC invalidates in flight now this one has completed, that were issued by the Secure CMDQ.

s_number_waiting_to_go_out unsigned int

The total number of ATC invalidates waiting to be pushed out from the Secure CMDQ.

ssd_of_cmdq enum

The SSD of the CMDQ.

ssd_of_streamid enum

The SSD of the StreamID.

smmu_atc_inv_end

An ATC invalidate request has been started. Fields:

PASID_global bool

If has a SubstreamID (PASID) then if it is 'global' or not.

address unsigned int

The Untranslated Address to use.

log2_size_in_bytes unsigned int

Log2 of the size of the region in bytes to invalidate.

response enum

The response to the ATC invalidate.

ssd enum

The SSD of the stream.

streamid unsigned int

The StreamID to use.

substreamid unsigned int

The SubstreamID to use (or ~0u if not used). This corresponds to the PASID.

smmu_atc_inv_start

An ATC invalidate request has been started. Fields:

PASID_global bool

If has a SubstreamID (PASID) then if it is 'global' or not.

address unsigned int

The Untranslated Address to use.

log2_size_in_bytes unsigned int

Log2 of the size of the region in bytes to invalidate.

ssd enum

The SSD of the stream.

streamid unsigned int

The StreamID to use.

substreamid unsigned int

The SubstreamID to use (or ~0u if not used). This corresponds to the PASID.

smmu_ats_initial

This is the initial ATS request. Fields:

XT bool

The XT bit.

ia unsigned int

Input address.

max_number_of_replies unsigned int

The maximum number of replies allowed to return.

no_write bool

The NW (no write flag) of the ATS request. If clear then the requester is going to do a write.

pasid_execute_requested bool

If there is a PASID (substreamid != ~0u) then this represents the requester asking for execution permissions.

pasid_privileged_mode_requested bool

If there is a PASID (substreamid != ~0u) then this represents the requester asking for the privileged mode's permissions.

protected_mode bool

Is the ATS Request protected-mode?.

source_cxl bool

Does the ATS request have Source.CXL set?.

ssd enum

The SSD of the ATS request.

ssd_ns bool

Incoming SSD is non-secure.

streamid unsigned int

StreamID of the ATS request.

substreamid unsigned int

SubstreamID (which is identical to the PASID) of the ATS request. If no PASID-prefix is sent then this is ~0u.

tbu unsigned int

Translation Buffer Unit number.

smmu_ats_reply_failure

This is an ATS reply indicating failure. Fields:

event enum

Equivalent event number that would have been generated for an equivalent ordinary transaction.

failure enum

What is the failure response code?.

state enum

The transaction state of the successfully ATS request.

smmu_ats_reply_success

This is an ATS reply, typically the SMMU will only return a single response, even if the requester indicated it could accept more replies. NOTE that the SMMU responds with 'success' in some cases when a fault is encountered and RW==0. Fields:

N bool

Non-snooped access. If one then the requester must clear the NoSnoop bit on transactions, unless otherwise enabled in a Function-specific manner.

P bool

Privileged mode. These permissions related to privileged mode.

RWX enum

Read/Write/Execute.

U bool

Untranslated access. If one, and RW !=0 then use UntranslatedAccesses for the allowed accesses by RW(X).

cxl_io bool

The CXL.io response.

inner enum

The inner cacheability attributes to use for TranslatedAccesses.

input_address unsigned int

Input address of the ATS request.

instcfg enum

The STE.INSTCFG field.

outer enum

The outer cacheability attributes to use for TranslatedAccesses.

pas enum

The PAS this mapping corresponds to. This holds the same information as the TE bit for realm streams.

privcfg enum

The STE.PRIVCFG field.

shareability unsigned int

The shareability to use for TranslatedAccesses.

size unsigned int

The size of the region covered by this translation.

state enum

The transaction state of the successfully ATS request.

translated_address unsigned int

If $RW \neq 0$ & $U \neq 0$, then the Translated Address that a TranslatedAccess can be made with.

smmu_axi_stream_msi

An SMMU generated MSI is directly sent through the axi_stream_msi_m port, typically connected to the GIC port axi_stream_msi_s. Fields:

TDEST unsigned int

Routing information for the data stream, typically identifying the GIC.

TID unsigned int

Data stream identifier for the SMMU.

axi_stream_msi_addr_to_match unsigned int

Current address to match for SMMU-originated MSIs to send out of the axi_stream_msi_m port.

data unsigned int

The MSI sent.

smmu_final_transaction

This is the transaction group request to remap has completed one set of remapping. For ordinary transactions, this represents a bundle of transactions with the same attributes but different addresses within a certain range around the address. If it stalls then it will report through this trace source, stall (stag_if_stalling != ~0u) and when resume will issue another smmu_initial_transaction as it undergoes remapping again. This trace source can also represent part of the process for ATOS/ATS or finding STE.PPAR for PRI requests that need to be auto-responded to. Fields:

axmmuflow enum

The AxMMUFLOW for this transaction group. storable – allow the transaction to be stalled if configured. translated-access – is a PCIe Translated Access non-storable – do not stall transaction even if configured. transfault-flow – do not report for Translation Related Faults, but tell client device to use a page request mechanism instead (for example PRI).

begin_input_address_range unsigned int

The start of the input address range that is size_of_region_in_bytes.

begin_ipa_range unsigned int

The start of the IPA range that is of size_of_region_in_bytes.

begin_output_address_range unsigned int

The start of the output address range that is of size_of_region_in_bytes.

cmo_point enum

The point associated with the CMO, if applicable.

input_address unsigned int

The input address of the transaction group.

input_inner enum

Inner cacheability for the input attributes.

input_outer enum

Outer cacheability for the input attributes.

input_prot enum

The instruction/data, bus-ns/s and privileged/user marking.

input_sh enum

Shareability for the input attributes.

ipa_address unsigned int

The IPA of the transaction.

kind enum

The kind of access this transaction group that this represents.

mecid unsigned int

The MECID of the transaction.

mpam_pmg_and_partid unsigned int

Bits[23:16] are the MPAM PMG. Bits[15:0] are the physical PARTID.

mpam_sp enum

The MPAM_SP of the transaction.

operation enum

The kind of operation that this represents.

output_address unsigned int

The input address of the transaction group.

output_inner enum

Inner cacheability for the output attributes.

output_outer enum

Outer cacheability for the output attributes.

output_prot enum

The instruction/data, bus-ns/s and privileged/user marking.

output_sh enum

Shareability for the output attributes.

output_vmid unsigned int

The output VMID/GBPA.IMPDEF or ~0u if not valid.

size_of_region_in_bytes unsigned int

An imp def size of region for which this translation is valid for.

ssd enum

The SSD of the transaction.

ssd_ns enum

The security state of the transaction.

stag_if_stalling unsigned int

This is the STAG used by the transaction if it is going to stall. It is ~0u if it is not going to stall.

state enum

The final transaction state.

streamid unsigned int

The StreamID of the transaction. ~0u if NoStreamID.

substreamid unsigned int

The SubstreamID of the transaction or 0xFFFF if no SubstreamID.

tbu unsigned int

Translation Buffer Unit number.

trans_id unsigned int

Transaction id. Top bit set if debug transaction.

smmu_gpt_tlb_invalidate

A GPT TLB invalidate has been initiated. Fields:

address unsigned int

The address as it appears in the operation.

encoded_size unsigned int

The size as it is encoded in the operation.

kind enum

The kind of operation this is.

pgs_in_bytes unsigned int

The PGS size in bytes.

size_in_bytes unsigned int

For range operations, the size as it appears in the operation.

source enum

Where the TLBI came from.

state enum

Is the operation well formed.

trans_id unsigned int

The transaction id of this invalidate.

smmu_gpt_tlb_invalidate_complete

The GPT TLB invalidate completed. Fields:

source enum

Where the TLBI came from.

trans_id unsigned int

The transaction id of this invalidate.

smmu_initial_transaction

This is the transaction group request to remap is going to start one set of remapping. For ordinary transactions, this represents a bundle of transactions with the same attributes but different addresses within a certain range around the address. A stalling transaction will report through this trace source when it unstalls. This trace source can also represent part of the process for ATOS/ATS or finding STE.PPAR for PRI requests that need to be auto-responded to. Fields:

XT_and_output_pas_checking enum

The XT bit for PCIe Transactions. This specifies the requested check on the output PAS that the device asked for.

axmmuflow enum

The AxMMUFLOW for this transaction group. stallable – allow the transaction to be stalled if configured. translated-access – is a PCIe Translated Access non-stallable – do not stall transaction even if configured. transfault-flow – do not report for Translation Related Faults, but tell client device to use a page request mechanism instead (for example PRI).

cmo_point enum

The point associated with the CMO, if applicable.

dcmdq_qcp_and_index unsigned int

If this is a DCMDQ fetch being translated then this field indicates the DCMDQ QCP index in bits [23:8] and the index in the page in bits [7:0]. Otherwise this is 0xFFFFFFFF.

input_address unsigned int

The input address of the transaction group.

input_inner enum

Inner cacheability for the input attributes.

input_outer enum

Outer cacheability for the input attributes.

input_prot enum

The instruction/data, bus-ns/s and privileged/user marking.

input_sh enum

Shareability for the input attributes.

kind enum

The kind of access this transaction group that this represents.

mecid unsigned int

The incoming MECID for NoStreamID transactions. ~0u for all other transactions.

mpam_partid unsigned int

The MPAM_PARTID for NoStreamID transactions. ~0u for all other transactions.

mpam_pmg unsigned int

The MPAM_PMG for NoStreamID transactions. ~0u for all other transactions.

mpam_sp enum

The MPAM_SP for NoStreamID transactions. ~0u for all other transactions.

operation enum

The kind of operation that this represents.

protected_mode enum

The PM bit.

ssd enum

The SSD of the transaction.

ssd_ns enum

The security state of the transaction.

streamid unsigned int

The StreamID of the transaction. ~0ull means NoStreamID.

substreamid unsigned int

The SubstreamID of the transaction or 0xFFFFFFFF if no SubstreamID.

tbu unsigned int

Translation Buffer Unit number.

trans_id unsigned int

Transaction id. Top bit set if debug transaction.

smmu_normalize_input_transaction

If the input transaction is normalized before being processed then this traceevent will fire. This is usually dependent on parameters of the implementation. Fields:

incoming_inner enum

The inner cacheability attributes.

incoming_is_instruction bool

The incoming transaction is marked as 'instruction'.

incoming_is_privileged bool

The incoming transaction is marked as 'privileged'.

incoming_outer enum

The outer cacheability attributes.

incoming_pas enum

The PAS of the incoming transaction.

incoming_shareability unsigned int

The incoming shareability.

normalized_inner enum

The normalized inner cacheability attributes.

normalized_is_instruction bool

The incoming transaction is marked as 'instruction'.

normalized_is_privileged bool

The incoming transaction is marked as 'privileged'.

normalized_outer enum

The normalized outer cacheability attributes.

normalized_pas enum

The PAS this mapping corresponds to.

normalized_shareability unsigned int

The normalized shareability.

ssd enum

The SSD of the transaction.

streamid unsigned int

The StreamID of the transaction. ~0ull means NoStreamID.

substreamid unsigned int

The SubstreamID of the transaction or 0xFFFFffff if no SubstreamID.

trans_id unsigned int

ID of the original transaction.

smmu_pmusnapshot_ack

Acknowledge the pmusnapshot_req to indicate the snapshot has occurred. This is part of a four-phase handshake. Fields:

pin_index unsigned int

The pin index of the signal, or ~0u if not relevant.

pmcg_index enum

The PMCG index.

value bool

The value of the signal.

smmu_pmusnapshot_ack_overridden

The value of pmusnapshot_ack was overridden (likely due to being in integration mode).

Fields:

pin_index unsigned int

The pin index of the signal, or ~0u if not relevant.

pmcg_index enum

The PMCG index.

value bool

The value of the signal.

smmu_pmusnapshot_ack_override_end

The overriding of the pmusnapshot_ack signal has ended. Fields:

pin_index unsigned int

The pin index of the signal, or ~0u if not relevant.

pmcg_index enum

The PMCG index.

smmu_pmusnapshot_req

Take a snap shot of the PMU values as though SMMU_PMCG_CAPR.CAPTURE had been written. This is part of a four-phase handshake. Fields:

pin_index unsigned int

The pin index of the signal, or ~0u if not relevant.

pmcg_index enum

The PMCG index.

value bool

The value of the signal.

smmu_poison_tw_data

Poison data has been returned to a table walk transaction. Fields:

bitmap_of_poison unsigned int

The bitmap of which beats of the transaction where poisoned.

is_cas bool

True if this is a compare-and-swap operation.

number_of_64bit_beats unsigned int

Number of beats this transaction fetched.

paddress unsigned int

Physical address of the table walk transaction.

pas enum

The PAS of the bus transaction.

ras_group_id unsigned int

If non-~0u then is the RAS group that we wish to attach the error to. Otherwise will use the IMP DEF policy.

ras_record_index_in_group unsigned int

If non-~0u then is the RAS record index in the group that we wish to attach the error to. Otherwise will use the IMP DEF policy.

what enum

What table walk was being performed.

smmu_pri_resp

The CMD_PRI_RESP command is queued to be sent to the PCIe system. Fields:

auto_response_trans_id unsigned int

trans_id of PRI request we are auto-responding to, or ~0u if not valid.

cmd_id unsigned int

Command id, or ~0u if not valid.

cons unsigned int

CONS of the command. ~0u if an auto-response.

prgindex unsigned int

PRG Index.

response enum

If has a SubstreamID (PASID) then if it is 'global' or not.

ssd enum

The SSD of the StreamID this corresponds to.

streamid unsigned int

StreamID in the invalidate command.

substreamid unsigned int

SubstreamID or ~0u if SSValid == false.

smmu_pri_resp_nop

The CMD_PRI_RESP command was NOPped. Fields:

cmd_id unsigned int

Command id.

cons unsigned int

CONS of the command.

prgindex unsigned int

PRG Index.

response enum

If has a SubstreamID (PASID) then if it is 'global' or not.

ssd enum

The SSD of the StreamID this corresponds to.

streamid unsigned int

StreamID in the invalidate command.

substreamid unsigned int

SubstreamID or ~0u if SSValid == false.

why enum

Why the CMD_PRI_RESP was NOPped.

smmu_priq_resp_fake_return

A PRIQ Response is posted to the PCIe subsystem and so has no acknowledgement that it is received. However, in the model then we artificially know when the the PRIQ Response has been delivered to the PCIe subsystem, even if the ATC has not yet acted on it. Fields:

prgindex unsigned int

The PRG Index this request corresponds to.

ssd enum

SSD of response.

streamid unsigned int

The StreamID to use.

substreamid unsigned int

The SubstreamID to use (or ~0u if not used). This corresponds to the PASID.

what enum

What happened.

smmu_priq_resp_start

A PRIQ Response has been posted to the PCIe subsystem. As the response is posted then there is no way of knowing when it is received by the EndPoint. Fields:

prgindex unsigned int

The PRG Index this request corresponds to.

response enum

If has a SubstreamID (PASID) then if it is 'global' or not.

ssd enum

SSD of response.

streamid unsigned int

The StreamID to use.

substreamid unsigned int

The SubstreamID to use (or ~0u if not used). This corresponds to the PASID.

smmu_thread_wait_wake

Traces a thread's wait/wake status. Fields:

current_ticks unsigned int

The current tick count of simulated time.

event enum

What is happening to this thread.

thread_index unsigned int

The ID of this thread.

ticks unsigned int

If the event relates to a time then this is held in this field. Otherwise, 0.

stall_transaction

A transaction is about to stall. Fields:

stag unsigned int

STAG.

streamid unsigned int

StreamID.

streamid_ns bool

The StreamID was for the non-secure world.

trans_id unsigned int

Transaction id.

stall_transaction_inhibited_by_STALL_MAX

A transaction is about to stall but the maximum number of transactions have stalled and we can't report this one to the event queue (even if non-full). Fields:

streamid unsigned int

StreamID.

streamid_ns bool

The StreamID was for the non-secure world.

trans_id unsigned int

Transaction id.

stall_transaction_resuming

A stalled transaction is resuming. Fields:

stag unsigned int

STAG if appropriate, or if was inhibited by STALL_MAX then 0xFAFA.

stallresult enum

What the transaction resumed to do.

streamid unsigned int

StreamID.

streamid_ns bool

The StreamID was for the non-secure world.

trans_id unsigned int

Transaction id.

start_ptw_read

Page Table Walk (read). This is the start of the physical access that the SMMU is making. Fields:

adomain enum

The actual attributes of the access that was used, after IMP DEF mangling.

aprot enum

The actual attributes of the access that was used, after IMP DEF mangling. PAS[1] (bit[3]), Instruction/Data (bit[2]), PAS[0] (bit[1]), Privileged/User (bit[0]).

desckind enum

The kind of descriptor.

inner_cache enum

The actual attributes of the access that was used, after IMP DEF mangling.

input_address unsigned int

Input address this lookup is for (after TBI normalisation).

mecid unsigned int

The MECID of the TTD fetch.

mpam_pmg_and_partid unsigned int

Bits[23:16] are the MPAM PMG. Bits[15:0] are the physical PARTID.

ns enum

Non-Secure on the bus.

outer_cache enum

The actual attributes of the access that was used, after IMP DEF mangling.

pa_address unsigned int

Physical address.

pas enum

The PAS of the TTD fetch.

ssd enum

SSD of the transaction.

ssd_ns enum

Security state of stream.

streamid unsigned int

The StreamID this is for.

substreamid unsigned int

The SubstreamID this is for.

trans_id unsigned int

Transaction id.

ttb_grain_stage_and_level unsigned int

Which TTB used in bits[31:24]. Which Grain size used in bit[23:16]:- 0 – 4 KiB 1 – 16 KiB 2 – 64 KiB 3 – 64 KiB ARMv8.2 where descriptor bits [15:12] for bits [51:48] of output address 4 – 4 KiB FEAT_LPA2 where address is {bits[9:8], bits[49:12]} 5 – 16 KiB FEAT_LPA2 where address is {bits[9:8], bits[49:14], 2'b00} 6 – 64 KiB ARMv8.2 where descriptor bits [15:12] are ignored Stage in bits[15:8]. Level in bits[7:0], level -1 is 0xF.

ste_cc.STE_commentary

A commentary from the cache about what is going on. Fields:

output string

The stream output.

ste_cc.STE_entry_invalidated

An entry in the cache has been invalidated. Fields:

entry_id unsigned int

The entry id that is being invalidated.

reason enum

Why the entry is being invalidated.

ste_entry_allocated

An STE entry has been allocated. Fields:

entry_id unsigned int

The id of the cache entry. Top bit is set if is a debug entry.

ns enum

For the non-secure world.

ssd enum

The SSD of the entry.

ste string

A textual description of the STE.

streamid unsigned int

The StreamID of the entry this will match.

sup_btm

The pin is driven. This indicates the system supports BTM. This pin is sampled on the negedge of the reset pin and has to be driven beforehand for the value to take effect. Fields:

value bool

The value of the signal.

sup_cohacc

The pin is driven. This indicates the system supports COHACC. This pin is sampled on the negedge of the reset pin and has to be driven beforehand for the value to take effect. Fields:

value bool

The value of the signal.

sup_sev

The pin is driven. This indicates the system supports SEV. This pin is sampled on the negedge of the reset pin and has to be driven beforehand for the value to take effect. Fields:

value bool

The value of the signal.

tbu0_reset_in

The reset signal of the TBU0. Fields:

value bool

The value of the signal.

tcu_reset_in

The reset signal. Fields:

value bool

The value of the signal.

tlb_entry_allocated

A TLB entry has been allocated. Fields:

aset unsigned int

The ASET of this entry.

asid unsigned int

ASID if appropriate.

index unsigned int

Index of TLB entry.

input_end_incl_address unsigned int

The end inclusive address of the input range that this matches.

input_start_address unsigned int

The start address of the input range that this matches.

output_end_incl_address unsigned int

The end inclusive address of the output range.

output_start_address unsigned int

The start address of the output range.

scheme enum

The tagging scheme used.

ssd enum

The SSD of the tagging scheme.

ssd_ns enum

Security State.

tbi bool

Was the entry formed using Top Byte Ignore (TBI).

vmid unsigned int

VMID if appropriate.

tlb_info_tlb_entries_overlap

A TLB entry was inserted into the TLB and it overlaps an existing entry. This isn't a problem as it was inserted in such a way that it architecturally works. Fields:

end_address_of_new_entry unsigned int

End address of new entry.

end_address_of_old_entry unsigned int

End address of old entry.

how_inserted enum

How the entry was inserted.

index_of_new_entry unsigned int

Index of new entry.

index_of_old_entry unsigned int

Index of old entry.

start_address_of_new_entry unsigned int

Start address of new entry.

start_address_of_old_entry unsigned int

Start address of old entry.

verbose_commentary

This is a verbose commentary on the translation process the SMMU is performing. Fields:

output string

The stream output.

warning_MSI_EVENTQ_address_out_of_range_of_oas

The MSI address is configured outside of the range of the downstream address bus size.
Fields:

address unsigned int

The untruncated address of the MSI.

warning_MSI_GERROR_address_out_of_range_of_oas

The MSI address is configured outside of the range of the downstream address bus size.
Fields:

address unsigned int

The untruncated address of the MSI.

warning_MSI_HACDBS_PROCESSING_COMPLETE_address_out_of_range_of_oas

The MSI address is configured outside of the range of the downstream address bus size.
Fields:

address unsigned int

The untruncated address of the MSI.

warning_MSI_HDBSS_TABLE_FULL_address_out_of_range_of_oas

The MSI address is configured outside of the range of the downstream address bus size.
Fields:

address unsigned int

The untruncated address of the MSI.

warning_MSI_PRIQ_address_out_of_range_of_oas

The MSI address is configured outside of the range of the downstream address bus size.
Fields:

address unsigned int

The untruncated address of the MSI.

warning_MSI_S_EVENTQ_address_out_of_range_of_oas

The MSI address is configured outside of the range of the downstream address bus size.
Fields:

address unsigned int

The untruncated address of the MSI.

warning_MSI_S_GERROR_address_out_of_range_of_oas

The MSI address is configured outside of the range of the downstream address bus size.
Fields:

address unsigned int

The untruncated address of the MSI.

warning_MSI_S_HACDBS_PROCESSING_COMPLETE_address_out_of_range_of_oas

The MSI address is configured outside of the range of the downstream address bus size.

Fields:

address unsigned int

The untruncated address of the MSI.

warning_MSI_S_HDBSS_TABLE_FULL_address_out_of_range_of_oas

The MSI address is configured outside of the range of the downstream address bus size.

Fields:

address unsigned int

The untruncated address of the MSI.

warning_MSI_pmcg_address_out_of_range_of_oas

The MSI Address of the Performance Monitor Counter Group (PMCG) is out of range of the OAS and so will be silently truncated. Fields:

address unsigned int

The untruncated address of the MSI.

pmcg_index unsigned int

Index of the PMCG.

ssd_ns bool

True if this PMCG controlled by the non-secure world.

warning_discarding_interrupt_EVENTQ_as_irqen_low

Interrupt generation is turned off by SMMU_IRQ_CTRL.EVENTQ_IRQEN.

warning_discarding_interrupt_GERROR_as_irqen_low

Interrupt generation is turned off by SMMU_IRQ_CTRL.GERROR_IRQEN.

warning_discarding_interrupt_HACDBS_PROCESSING_COMPLETE_as_irqen_low

Interrupt generation is turned off by SMMU_IRQ_CTRL.HACDBS_IRQEN.

warning_discarding_interrupt_HDBSS_TABLE_FULL_as_irqen_low

Interrupt generation is turned off by SMMU_IRQ_CTRL.HDBSS_IRQEN.

warning_discarding_interrupt_PRIQ_as_irqen_low

Interrupt generation is turned off by SMMU_IRQ_CTRL.PRIQ_IRQEN.

warning_discarding_interrupt_S_EVENTQ_as_irqen_low

Interrupt generation is turned off by SMMU_S_IRQ_CTRL.EVENTQ_IRQEN.

warning_discarding_interrupt_S_GERROR_as_irqen_low

Interrupt generation is turned off by SMMU_S_IRQ_CTRL.GERROR_IRQEN.

warning_discarding_interrupt_S_HACDBS_PROCESSING_COMPLETE_as_irqen_low

Interrupt generation is turned off by SMMU_IRQ_S_CTRL.HACDBS_IRQEN.

warning_discarding_interrupt_S_HDBSS_TABLE_FULL_as_irqen_low

Interrupt generation is turned off by SMMU_IRQ_S_CTRL.HDBSS_IRQEN.

warning_ns_gerrorn_bad_acknowledge

A GERROR was acknowledged by SW in the GERRORN register that did not have an active error. The result is **UNPREDICTABLE** if this will generate an interrupt or not. Fields:

which enum

Which GERROR was acknowledge when there was no active error.

warning_reg_after_doesnt_match_written_value

A write occurred that tried to set bits in a register, that for one reason or another, failed to get written. Fields:

desc string

The textual description of what happened.

warning_s_gerrorn_bad_acknowledge

A GERROR was acknowledged by SW in the GERRORN register that did not have an active error. The result is **UNPREDICTABLE** if this will generate an interrupt or not. Fields:

which enum

Which GERROR was acknowledge when there was no active error.

what_going_to_do_with_terminated_event

A terminating transaction has produced an event, this tells you what the model is going to do with the event. Fields:

CD.S bool

The CD.S field if available.

S2 bool

The event is related to Stage 2.

STE.S1STALLD bool

The STE.S1STALLD field if available.

STE.S2S bool

The STE.S2S field if available.

aborts bool

The transaction will abort.

axmmuflow enum

The AxMMUFLOW for this transaction group. storable – allow the transaction to be stalled if configured. translated-access – is a PCIe Translated Access non-storable – do not stall transaction even if configured. transfault-flow – do not report for Translation Related Faults, but tell client device to use a page request mechanism instead (for example PRI).

is_tr_fault bool

Is a Translation Related fault.

protected_mode bool

The transaction is protected-mode. As such, it cannot stall and will obey the report configuration bits.

reports bool

The transaction will attempt to report.

ssd enum

The SSD of the transaction.

ssd_ns bool

The transaction is classified as SSD non-secure.

supports_stall_model bool

The implementation supports the stall model.

trans_id unsigned int

The transaction id.

why_abort_decision enum

The reason why the transaction aborted/did not abort.

why_report_decision enum

The reason why the transaction reported/did not report.

2.125 MMU_700

This section describes the trace sources.

ArchMsg.Error.error

These messages are about activity occurring on the SMMU that is considered an error. Messages will only come out here if parameter `all_error_messages_through_trace` is true. `DISPLAY %{\output}`. Fields:

output string

The stream output.

ArchMsg.Error.fetch_from_memory_type_not_supporting_httu

A descriptor fetch from an HTTU-enabled translation regime to an unsupported memory type was made. Whilst the fetch itself may succeed, if an update to the descriptor was attempted then it would fail. Fields:

address unsigned int

The address of the descriptor fetch.

desc_inner enum

Inner cacheability of descriptor.

desc_outer enum

Outer cacheability of descriptor.

desc_prot enum

The instruction/data, bus-ns/s and privileged/user marking.

desc_sh enum

Shareability of descriptor.

ssd enum

The SSD of the transaction.

ssd_ns bool

Is the Security State Determination of the transaction non-secure.

stage unsigned int

The stage at which we had a problem.

streamid unsigned int

The StreamID of the transaction.

substreamid unsigned int

The SubstreamID of the transaction (or ~0u if no substreamid).

trans_id unsigned int

Transaction id.

ArchMsg.Error.ns_cmdq

There is an error associated with the command queue. Fields:

cmd_id unsigned int

Command ID of the command in error.

cons unsigned int

CONS of the command.

what string

What is wrong.

ArchMsg.Error.priq_streamid_truncated

The SMMU received a PCIe PRI request with a StreamID that was larger than that which the SMMU has been configured for. The StreamID that appears in the PRIQ entry will be truncated. Fields:

actual_streamid unsigned int

The actual StreamID that this request has.

sidsize unsigned int

The bit width of the SMMU for StreamIDs, as indicated by SMMU_IDR1.SIDSIZE.

trans_id unsigned int

The transaction ID that identifies this request.

trunc_streamid unsigned int

The truncated StreamID that will appear in the PRIQ entry.

ArchMsg.Error.priq_substreamid_truncated

The SMMU received a PCIe PRI request with a PASID prefix (SubstreamID) that was larger than that which the SMMU has been configured for. The SubstreamID that appears in the PRIQ entry will be truncated. Fields:

actual_substreamid unsigned int

The actual SubstreamID that this request has.

ssidsize unsigned int

The bit width of the SMMU for SubstreamIDs, as indicated by SMMU_IDR1.SSIDSIZE.

trans_id unsigned int

The transaction ID that identifies this request.

trunc_substreamid unsigned int

The truncated SubstreamID that will appear in the PRIQ entry.

ArchMsg.Error.s_cmdq

There is an error associated with the command queue. Fields:

cmd_id unsigned int

Command ID of the command in error.

cons unsigned int

CONS of the command.

what string

What is wrong.

ArchMsg.Error.tlb_entries_overlap

A TLB entry was attempted to be inserted into the TLB and was determined that it overlaps an existing entry. This check is not perfect but will catch simple errors. Fields:

do_f_tlb_conflict bool

Chosen to perform an F_TLB_CONFLICT.

end_address_of_new_entry unsigned int

End address of new entry.

end_address_of_old_entry unsigned int

End address of old entry.

index_of_new_entry unsigned int

Index of new entry.

index_of_old_entry unsigned int

Index of old entry.

start_address_of_new_entry unsigned int

Start address of new entry.

start_address_of_old_entry unsigned int

Start address of old entry.

why enum

Why it is an error for these two entries to overlap.

ArchMsg.Error.tlb_entry_not_invalidated_due_to_ril

An entry in the cache was not invalidated even though in the right address range because of the RIL fields of the command the entry do not match. Fields:

cmd_num unsigned int

The NUM field of the RIL part of the command.

cmd_ril_tg enum

The RIL_TG field of the RIL part of the command.

cmd_ril_ttl unsigned int

The TTL field of the RIL part of the command, zero means any level. 0x80 means match level 0 (and is from a DVM message).

cmd_scale unsigned int

The SCALE field of the RIL part of the command.

entry_id unsigned int

The entry id that is being invalidated.

tlb_entry string

The TLB entry.

ArchMsg.Error.vatos_sel_vmid_out_of_range

The SMMU_(S_)VATOS_SEL.VMID field was programmed with a VMID that was too wide for this implementation (SMMU_IDR0.VMID16 == 0). DISPLAY %{ssd_ns:(s-|ns-)}VMID:%{vmid} is out of range. Fields:

ssd_ns bool

The security state of the VATOS interface.

vmid unsigned int

The VMID programmed.

ArchMsg.Info.info

These are information messages about what is happening in the SMMU. DISPLAY %{output}. Fields:

output string

The stream output.

ArchMsg.Warning.CMD_RESUME_no_transaction_resumed

A CMD_RESUME was issued that matched no transaction. Fields:

stag unsigned int

STAG in the CMD_RESUME.

streamid unsigned int

StreamID in the CMD_RESUME.

streamid_ns bool

The StreamID was for the non-secure world.

ArchMsg.Warning.atc_inv_strange

Something was odd about the CMD_ATC_INV. DISPLAY CMD_ATC_INV strange as: %{why}.
Fields:

cmd_id unsigned int

Command id.

ssd_of_cmdq enum

The SSD of the CMDQ.

why string

Why the CMD_ATC_INV was strange.

ArchMsg.Warning.bad_axi_stream_msi_addr_to_match_s

The signal was not driven at reset by the external system, but was driven after we first started using the default value we had assumed. The external system, if it is going to drive the signal, must do so at first reset so that both sides of the connection can agree on the value of this signal. Fields:

default_value unsigned int

The default value of the signal we have been assuming.

value unsigned int

The value of the signal.

ArchMsg.Warning.bad_conf_reset_of_SMMU_S_GBPA_ABORT

The signal was not driven at reset by the external system, but was driven after we first started using the default value we had assumed. The external system, if it is going to drive the signal, must do so at first reset so that both sides of the connection can agree on the value of this signal. Fields:

default_value bool

The default value of the signal we have been assuming.

value bool

The value of the signal.

ArchMsg.Warning.bad_sec_override

The signal was not driven at reset by the external system, but was driven after we first started using the default value we had assumed. The external system, if it is going to drive the signal, must do so at first reset so that both sides of the connection can agree on the value of this signal. Fields:

default_value bool

The default value of the signal we have been assuming.

value bool

The value of the signal.

ArchMsg.Warning.bad_sup_btm

The signal was not driven at reset by the external system, but was driven after we first started using the default value we had assumed. The external system, if it is going to drive the signal, must do so at first reset so that both sides of the connection can agree on the value of this signal. Fields:

default_value bool

The default value of the signal we have been assuming.

value bool

The value of the signal.

ArchMsg.Warning.bad_sup_cohacc

The signal was not driven at reset by the external system, but was driven after we first started using the default value we had assumed. The external system, if it is going to drive the signal, must do so at first reset so that both sides of the connection can agree on the value of this signal. Fields:

default_value bool

The default value of the signal we have been assuming.

value bool

The value of the signal.

ArchMsg.Warning.bad_sup_httu

The signal was not driven at reset by the external system, but was driven after we first started using the default value we had assumed. The external system, if it is going to drive the signal, must do so at first reset so that both sides of the connection can agree on the value of this signal. Fields:

default_value bool

The default value of the signal we have been assuming.

value bool

The value of the signal.

ArchMsg.Warning.bad_sup_sev

The signal was not driven at reset by the external system, but was driven after we first started using the default value we had assumed. The external system, if it is going to drive the signal, must do so at first reset so that both sides of the connection can agree on the value of this signal. Fields:

default_value bool

The default value of the signal we have been assuming.

value bool

The value of the signal.

ArchMsg.Warning.bad_tbu0_reset_in

The signal was not driven at reset by the external system, but was driven after we first started using the default value we had assumed. The external system, if it is going to drive

the signal, must do so at first reset so that both sides of the connection can agree on the value of this signal. Fields:

default_value bool

The default value of the signal we have been assuming.

value bool

The value of the signal.

ArchMsg.Warning.bad_tcu_reset_in

The signal was not driven at reset by the external system, but was driven after we first started using the default value we had assumed. The external system, if it is going to drive the signal, must do so at first reset so that both sides of the connection can agree on the value of this signal. Fields:

default_value bool

The default value of the signal we have been assuming.

value bool

The value of the signal.

ArchMsg.Warning.contig_bit_gives_too_large_region_for_TxSZ

If the contig bit was used then the size of the contig region would be larger than that indicated by TxSZ. Fields:

desc_kind enum

Descriptor kind.

el enum

Exception level.

input_address unsigned int

Input address.

log2_contig_region_size unsigned int

Log2(contig region size in bytes).

output_address unsigned int

Output address.

stage_and_level unsigned int

The stage in bits [7:4] and the level in [3:0]. Level -1 is 0xF.

vmid unsigned int

VMID if appropriate.

ArchMsg.Warning.contig_bit_has_inconsistent_input_and_output_address

If the contig bit was used then the some bits of the output address held in the descriptor and the input address must match. Fields:

desc_kind enum

Descriptor kind.

el enum

Exception level.

input_address unsigned int

Input address.

log2_contig_region_size unsigned int

Log2(contig region size in bytes).

match_mask unsigned int

Bits that must match.

output_address unsigned int

Output address.

stage_and_level unsigned int

The stage in bits [7:4] and the level in [3:0]. Level -1 is 0xF.

vmid unsigned int

VMID if appropriate.

ArchMsg.Warning.fetch_from_memory_type_not_supporting_httu

A descriptor fetch from an HTTU-enabled translation regime to an unsupported memory type was made. Whilst the fetch itself may succeed, if an update to the descriptor was attempted then it would fail. Fields:

address unsigned int

The address of the descriptor fetch.

desc_inner enum

Inner cacheability of descriptor.

desc_outer enum

Outer cacheability of descriptor.

desc_prot enum

The instruction/data, bus-ns/s and privileged/user marking.

desc_sh enum

Shareability of descriptor.

ssd enum

The SSD of the transaction.

ssd_ns bool

Is the Security State Determination of the transaction non-secure.

stage unsigned int

The stage at which we had a problem.

streamid unsigned int

The StreamID of the transaction.

substreamid unsigned int

The SubstreamID of the transaction (or ~0u if no substreamid).

trans_id unsigned int

Transaction id.

ArchMsg.Warning.msi_address_truncated

An MSI was generated, but the address was silently truncated due to the limited downstream address bus width. Fields:

pmcg_index_and_counter unsigned int

If this is a PMCG interrupt, then the top 16 bits are the PMCG index, the lower 16 bits are counter index.

ras_error_group_id_node_id_and_record_index_in_node unsigned int

If this is a RAS interrupt then this is the error group record id in bits[23:16], the node_id in [15:8] and the record index in the node in [7:0].

truncated_address unsigned int

The truncated address.

untruncated_address unsigned int

The untruncated address.

which enum

Which MSI this is.

ArchMsg.Warning.msi_lost

An MSI was attempted to be sent, but couldn't be sent. Fields:

id unsigned int

ID of this interrupt transaction.

kind enum

What kind of interrupt.

why enum

Why this interrupt was denied.

ArchMsg.Warning.pmcg_non_secure_world

A commentary on potential mis-programming of the PMCG when transitioning the PMCG between security states. Fields:

commentary string

The commentary.

ArchMsg.Warning.pmcg_programming_violates_security

A commentary on potential mis-programming of the PMCG when transitioning the PMCG between different security states. This is under the assumption that an agent from another security state could be writing to the PMCG. Fields:

commentary string

The commentary.

ArchMsg.Warning.priq_auto_response_failed_to_find_STE

The PRIQ was going to generate an auto-response, but failed to find an STE and so is returning a Failure message to the EndPoint which should disable the PRI interface of the EndPoint. Fields:

trans_id unsigned int

The ID of the PPR that is being lost.

why enum

Why the PPR is being lost.

ArchMsg.Warning.priq_overflow_bad_acking

Indicates that an overflow condition was acknowledged by writing to:- SMMU_PRIQ_CONS.OVACKFLG but an overflow condition did not exist. Fields:

new_ovackflg bool

The new value of the SMMU_PRIQ_CONS.OVACKFLG.

ArchMsg.Warning.priq_smmuen_forces_effective_priqen_low

If SMMUEN == 0, then the effective value of PRIQEN is 0. This warning is triggered when PRIQEN == 1 && SMMUEN == 0; which may not be what was intended. The PRIQ cannot be active if SMMUEN == 0.

ArchMsg.Warning.sev_lost

A SEV was lost because it isn't supported according to SMMU_IDRO.SEV. DISPLAY SEV was lost because: %{why}. Fields:

why enum

Why the SEV was generated.

ArchMsg.Warning.smmu_pcie_rc_is_in_reset_ignoring_atc_invalidate

A CMD_PRI_RESP or CMD_ATC_INV was sent with a streamid that mapped to a PCIe Root Complex that is in reset. The message will be discarded and act as though completed successfully. Fields:

port_index unsigned int

The port index (node index) that is in reset but we would have sent it to.

streamid unsigned int

The StreamID that failed to map.

ArchMsg.Warning.smmu_pcie_rc_not_found_for_streamid

A CMD_PRI_RESP or CMD_ATC_INV was sent with a streamid that did not map to a PCIe Root Complex. This might be that the SW used an incorrect StreamID or it might be that the model has not been connected correctly. ATC Invalidate messages complete as though successful and PRI Requests are ignored. Fields:

streamid unsigned int

The StreamID that failed to map.

ArchMsg.Warning.smmu_pmusnapshot

Something strange happened on the pmusnapshot_req/pmusnapshot_ack interface. Fields:

pin_index unsigned int

If the PMCG index corresponds to an array of signals, this is the index in the array, or 0 otherwise.

pmcg_index enum

The PMCG index.

warning string

The warning message.

ArchMsg.Warning.suspicious_overlapping_entries

Two DPT TLB entries are overlapping but they differ in ways that are potentially a SW error. Fields:

new_AC enum

The value of the 'AC' field that controls access to this region.

new_FWB bool

The region is FWB.

new_VMID unsigned int

The VMID, if any, associated with this region.

new_entry_index unsigned int

The entry index of the new DPT TLB entry.

new_entry_is_vmsa_formed bool

The new entry was inserted as a result of VMSA information.

new_input_end_incl_address unsigned int

The end inclusive address of the input range that this matches.

new_input_start_address unsigned int

The start address of the input range that this matches.

new_output_pas enum

The output PAS of this DPT region.

new_ssd enum

The SSD of the streams this region captures.

new_writeable bool

True if this region is writeable.

old_AC enum

The value of the 'AC' field that controls access to this region.

old_FWB bool

The region is FWB.

old_VMID unsigned int

The VMID, if any, associated with this region.

old_entry_index unsigned int

The entry index of the old DPT TLB entry.

old_entry_is_vmsa_formed bool

The old entry was inserted as a result of VMSA information.

old_input_end_incl_address unsigned int

The end inclusive address of the input range that this matches.

old_input_start_address unsigned int

The start address of the input range that this matches.

old_output_pas enum

The output PAS of this DPT region.

old_ssd enum

The SSD of the streams this region captures.

old_writeable bool

True if this region is writeable.

pas_differ bool

The output PAS of the two entries differ.

vmids_differ bool

The VMIDs are used by at least one of the AC schemes and are different.

vmsa_formed_writeable_while_DPT_entry_is_not bool

The VMSA-formed entry is writeable but the DPT Entry says it is not writeable.

ArchMsg.Warning.warning

These messages are about unusual (but not necessarily incorrect) activity occurring on the SMMU. DISPLAY %{output}. Fields:

output string

The stream output.

ArchMsg.Warning.warning_effective_EOPD_differs_from_what_would_be_cached_in_TLB

Effective value of EOPD differs from what would be cached in the TLB DISPLAY transaction (%{transaction_id}), sid (%{sid}), ssid (%{ssid}), ssd (%{ssd}), effective EOPD (%{effective_EOPD}), cached EOPD (%{cached_EOPD}). Fields:

cached_EOPD bool

The EOPD value that would be cached in the TLB.

effective_EOPD bool

The effective value of EOPD.

ssd enum

SSD.

streamid unsigned int

StreamID or ~0ull if NoStreamID.

substreamid unsigned int

SubstreamID or ~0u if SSValid == false.

transaction_id unsigned int

The transaction ID.

DPTTLB.tlb_commentary

A commentary from the cache about what is going on. Fields:

output string

The stream output.

DPTTLB.tlb_entry_invalidated

An entry in the cache has been invalidated. Fields:

entry_id unsigned int

The entry id that is being invalidated.

reason enum

Why the entry is being invalidated.

EVENTQ_config

Configuration of interrupt updated. Fields:

address unsigned int

Address of the MSI.

data unsigned int

Data payload of the MSI.

irgen bool

Appropriate bit of SMMU_s_IRQ_CTRL controlling this interrupt.

memattr enum

Memory type.

ns bool

PAS of bus attribute is non-secure.

pas enum

PAS of the MSI.

ra bool

read-allocate attribute.

sh enum

Shareability.

smmu_idr0_msi bool

Are MSIs supported for this security world.

tr bool

transient attribute.

wa bool

write-allocate attributes.

EVENTQ_irqen

The IRQEN has been changed. Fields:

new_value bool

The new value of IRQEN.

EVENTQ_irqen_ack

The IRQEN change has been acked. Fields:

new_value bool

The new value of IRQEN.

GERROR_config

Configuration of interrupt updated. Fields:

address unsigned int

Address of the MSI.

data unsigned int

Data payload of the MSI.

irqen bool

Appropriate bit of SMMU_s_IRQ_CTRL controlling this interrupt.

memattr enum

Memory type.

ns bool

PAS of bus attribute is non-secure.

pas enum

PAS of the MSI.

ra bool

read-allocate attribute.

sh enum

Shareability.

smmu_idr0_msi bool

Are MSIs supported for this security world.

tr bool

transient attribute.

wa bool

write-allocate attributes.

GERROR_irqen

The IRQEN has been changed. Fields:

new_value bool

The new value of IRQEN.

GERROR_irqen_ack

The IRQEN change has been acked. Fields:

new_value bool

The new value of IRQEN.

HACDBS_PROCESSING_COMPLETE_config

Configuration of interrupt updated. Fields:

address unsigned int

Address of the MSI.

data unsigned int

Data payload of the MSI.

irqen bool

Appropriate bit of SMMU_s_IRQ_CTRL controlling this interrupt.

memattr enum

Memory type.

ns bool

PAS of bus attribute is non-secure.

pas enum

PAS of the MSI.

ra bool

read-allocate attribute.

sh enum

Shareability.

smmu_idr0_msi bool

Are MSIs supported for this security world.

tr bool

transient attribute.

wa bool

write-allocate attributes.

HACDBS_PROCESSING_COMPLETE_irqen

The IRQEN has been changed. Fields:

new_value bool

The new value of IRQEN.

HACDBS_PROCESSING_COMPLETE_irqen_ack

The IRQEN change has been acked. Fields:

new_value bool

The new value of IRQEN.

HDBSS_TABLE_FULL_config

Configuration of interrupt updated. Fields:

address unsigned int

Address of the MSI.

data unsigned int

Data payload of the MSI.

irqen bool

Appropriate bit of SMMU_s_IRQ_CTRL controlling this interrupt.

memattr enum

Memory type.

ns bool

PAS of bus attribute is non-secure.

pas enum

PAS of the MSI.

ra bool

read-allocate attribute.

sh enum

Shareability.

smmu_idr0_msi bool

Are MSIs supported for this security world.

tr bool

transient attribute.

wa bool

write-allocate attributes.

HDBSS_TABLE_FULL_irqen

The IRQEN has been changed. Fields:

new_value bool

The new value of IRQEN.

HDBSS_TABLE_FULL_irqen_ack

The IRQEN change has been acked. Fields:

new_value bool

The new value of IRQEN.

PRIQ_config

Configuration of interrupt updated. Fields:

address unsigned int

Address of the MSI.

data unsigned int

Data payload of the MSI.

irqen bool

Appropriate bit of SMMU_s_IRQ_CTRL controlling this interrupt.

memattr enum

Memory type.

ns bool

PAS of bus attribute is non-secure.

pas enum

PAS of the MSI.

ra bool

read-allocate attribute.

sh enum

Shareability.

smmu_idr0_msi bool

Are MSIs supported for this security world.

tr bool

transient attribute.

wa bool

write-allocate attributes.

PRIQ_irqen

The IRQEN has been changed. Fields:

new_value bool

The new value of IRQEN.

PRIQ_irqen_ack

The IRQEN change has been acked. Fields:

new_value bool

The new value of IRQEN.

SMMU_CR0ACK_SMMUEN_hazarded_by_priq

The SMMU_r_CR0ACK.SMMUEN cannot acknowledge the change to SMMUEN because there are outstanding PRIQ writes.

SMMU_CR0ACK_SMMUEN_update

The acknowledge to SMMU_CR0.SMMUEN was performed. Fields:

new_value bool

The new value of the register.

SMMU_CR0_SMMUEN_old_set_complete

A set of transactions associated with the old value of SMMU_CR0.SMMUEN completed.

Fields:

last bool

This is the last set completing.

old_value bool

The old value of the set completing.

SMMU_CR0_SMMUEN_write

A write to SMMU_CR0.SMMUEN occurred. Fields:

new_value bool

The new value of the register.

old_value bool

The old value of the register.

SMMU_GBPA_old_set_complete

A set of transactions associated with the old value of SMMU_GBPA completed. Fields:

last bool

This is the last set completing.

old_value unsigned int

The old value of the set completing.

SMMU_GBPA_update

The Update flag to SMMU_GBPA was lowered. Fields:

new_value unsigned int

The new value of the register.

SMMU_GBPA_write

A write to SMMU_GBPA occurred. Fields:

new_value unsigned int

The new value of the register.

old_value unsigned int

The old value of the register.

SMMU_S_CR0ACK_SMMUEN_update

The acknowledge to SMMU_S_CR0.SMMUEN was performed. Fields:

new_value bool

The new value of the register.

SMMU_S_CR0_SMMUEN_old_set_complete

A set of transactions associated with the old value of SMMU_S_CR0.SMMUEN completed.

Fields:

last bool

This is the last set completing.

old_value bool

The old value of the set completing.

SMMU_S_CR0_SMMUEN_write

A write to SMMU_S_CR0.SMMUEN occurred. Fields:

new_value bool

The new value of the register.

old_value bool

The old value of the register.

SMMU_S_GBPA_old_set_complete

A set of transactions associated with the old value of SMMU_S_GBPA completed. Fields:

last bool

This is the last set completing.

old_value unsigned int

The old value of the set completing.

SMMU_S_GBPA_update

The Update flag to SMMU_S_GBPA was lowered. Fields:

new_value unsigned int

The new value of the register.

SMMU_S_GBPA_write

A write to SMMU_S_GBPA occurred. Fields:

new_value unsigned int

The new value of the register.

old_value unsigned int

The old value of the register.

S_EVENTQ_config

Configuration of interrupt updated. Fields:

address unsigned int

Address of the MSI.

data unsigned int

Data payload of the MSI.

irgen bool

Appropriate bit of SMMU_s_IRQ_CTRL controlling this interrupt.

memattr enum

Memory type.

ns bool

PAS of bus attribute is non-secure.

pas enum

PAS of the MSI.

ra bool

read-allocate attribute.

sh enum

Shareability.

smmu_idr0_msi bool

Are MSIs supported for this security world.

tr bool

transient attribute.

wa bool

write-allocate attributes.

S_EVENTQ_irqen

The IRQEN has been changed. Fields:

new_value bool

The new value of IRQEN.

S_EVENTQ_irqen_ack

The IRQEN change has been acked. Fields:

new_value bool

The new value of IRQEN.

S_GERROR_config

Configuration of interrupt updated. Fields:

address unsigned int

Address of the MSI.

data unsigned int

Data payload of the MSI.

irqen bool

Appropriate bit of SMMU_s_IRQ_CTRL controlling this interrupt.

memattr enum

Memory type.

ns bool

PAS of bus attribute is non-secure.

pas enum

PAS of the MSI.

ra bool
read-allocate attribute.

sh enum
Shareability.

smmu_idr0_msi bool
Are MSIs supported for this security world.

tr bool
transient attribute.

wa bool
write-allocate attributes.

S_ERROR_irqen
The IRQEN has been changed. Fields:

new_value bool
The new value of IRQEN.

S_ERROR_irqen_ack
The IRQEN change has been acked. Fields:

new_value bool
The new value of IRQEN.

S_HACDBS_PROCESSING_COMPLETE_config
Configuration of interrupt updated. Fields:

address unsigned int
Address of the MSI.

data unsigned int
Data payload of the MSI.

irqen bool
Appropriate bit of SMMU_s_IRQ_CTRL controlling this interrupt.

memattr enum
Memory type.

ns bool
PAS of bus attribute is non-secure.

pas enum
PAS of the MSI.

ra bool
read-allocate attribute.

sh enum
Shareability.

smmu_idr0_msi bool

Are MSIs supported for this security world.

tr bool

transient attribute.

wa bool

write-allocate attributes.

S_HACDBS_PROCESSING_COMPLETE_irqen

The IRQEN has been changed. Fields:

new_value bool

The new value of IRQEN.

S_HACDBS_PROCESSING_COMPLETE_irqen_ack

The IRQEN change has been acked. Fields:

new_value bool

The new value of IRQEN.

S_HDBSS_TABLE_FULL_config

Configuration of interrupt updated. Fields:

address unsigned int

Address of the MSI.

data unsigned int

Data payload of the MSI.

irqen bool

Appropriate bit of SMMU_s_IRQ_CTRL controlling this interrupt.

memattr enum

Memory type.

ns bool

PAS of bus attribute is non-secure.

pas enum

PAS of the MSI.

ra bool

read-allocate attribute.

sh enum

Shareability.

smmu_idr0_msi bool

Are MSIs supported for this security world.

tr bool

transient attribute.

wa bool

write-allocate attributes.

S_HDBSS_TABLE_FULL_irqen

The IRQEN has been changed. Fields:

new_value bool

The new value of IRQEN.

S_HDBSS_TABLE_FULL_irqen_ack

The IRQEN change has been acked. Fields:

new_value bool

The new value of IRQEN.

TLB.tlb_commentary

A commentary from the cache about what is going on. Fields:

output string

The stream output.

TLB.tlb_entry_invalidated

An entry in the cache has been invalidated. Fields:

entry_id unsigned int

The entry id that is being invalidated.

reason enum

Why the entry is being invalidated.

atc_inv_nop

The CMD_ATC_INV command is ignored as a **NOP**. This may be emitted multiple times if the CMD_ATC_INV is being ignored for multiple reasons. Fields:

address unsigned int

Untranslated Address to invalidate.

cmd_id unsigned int

Command id.

cons unsigned int

CONS of the command.

global bool

Global flag.

log2_size_in_bytes unsigned int

Log 2 of the size in bytes of the region to invalidate.

ssd_of_queue enum

The SSD of the CMDQ.

ssd_of_streamid enum

The SSD of the StreamID.

streamid unsigned int

StreamID in the invalidate command.

substreamid unsigned int

SubstreamID or ~0u if SSValid == false.

why enum

Why was NOPped.

atos_complete_fault

The ATOS operation completed with a fault. Fields:

effective_st1translate bool

Because of the settings and/or the ATOS type then the effective st1translate can be different.

fault_faddr unsigned int

The fault FADDR.

fault_faultcode enum

The fault code.

fault_reason enum

The fault reason.

ssd_ns bool

This is a non-secure ATOS operation.

st1translate bool

The StreamID/SubstreamID combination should have been translated by stage 1.

st2translate bool

The translation suffered a stage 2 translation.

state enum

The final transaction state of the ATOS operation.

vatos_vmid unsigned int

The VMID of the VATOS operation, or ~0u if not a VATOS operation.

atos_complete_fault_inv_req

The ATOS operation completed, faulted and generated an INV_REQ response. Fields:

ssd_ns bool

This is a non-secure ATOS operation.

vatos_vmid unsigned int

The VMID of the VATOS operation, or ~0u if not a VATOS operation.

why enum

Why the ATOS operation generated an INV_REQ.

atos_complete_success

The ATOS operation completed successfully. Fields:

base_addr unsigned int

The actual base address of region.

effective_st1translate bool

Because of the settings and/or the ATOS type then the effective st1translate can be different.

par_addr unsigned int

The PAR.ADDR field.

par_mair unsigned int

The memory attributes encoded as a MAIR.

par_ns bool

The PAR.NS field, for an SSD-ns request then this will always be 0.

par_sh enum

Shareability.

par_size bool

The PAR.Size field.

size_in_bytes unsigned int

The actual size in bytes of the region.

ssd_ns bool

This is a non-secure ATOS operation.

st1translate bool

The StreamID/SubstreamID combination should have been translated by stage 1.

st2translate bool

The translation suffered a stage 2 translation.

state enum

The final transaction state of the ATOS operation.

vatos_vmid unsigned int

The VMID of the VATOS operation, or ~0u if not a VATOS operation.

atos_run_set

The SMMU_s_GATOS_CTRL.RUN field was set to start the ATOS operation. Fields:

ssd_ns bool

This is a non-secure ATOS operation.

vatos bool

This is a VATOS operation.

atos_starting

The ATOS operation is starting. Fields:

addr unsigned int

The input address to the ATOS operation.

httui bool

Inhibit HTTU update.

ind bool

Instruction Data.

pnu bool

Privileged not User.

rnw bool

Read not Write.

ssd_ns bool

This is a non-secure ATOS operation.

ssec bool

If this is a secure ATOS operation then this is if it is secure or not.

streamid unsigned int

The StreamID requested.

substreamid unsigned int

The SubstreamID requested, or ~0u if no SubstreamID.

type enum

The requested ATOS type.

vatos_vmid unsigned int

The VMID of the VATOS operation, or ~0u if not a VATOS operation.

axi_stream_msi_addr_to_match_s

Address to use to send SMMU originated MSIs directly to the GIC. This pin is sampled on the negedge of the reset pin and has to be driven beforehand for the value to take effect. Fields:

value unsigned int

The value of the signal.

cd_cc.CD_commentary

A commentary from the cache about what is going on. Fields:

output string

The stream output.

cd_cc.CD_entry_invalidated

An entry in the cache has been invalidated. Fields:

entry_id unsigned int

The entry id that is being invalidated.

reason enum

Why the entry is being invalidated.

cd_entry_allocated

An CD entry has been allocated. Fields:

AssuredTranslation bool

The CD (and any L1CD) was fetched from an AssuredOnly stage 2 page and so might be capable of creating AssuredTranslations.

cd string

A textual description of the CD.

entry_id unsigned int

The id of the cache entry. Top bit is set if is a debug entry.

ns enum

For the non-secure world.

ssd enum

The SSD of the entry.

streamid unsigned int

The StreamID of the entry this will match.

substreamid unsigned int

The SubstreamID of the entry this will match. This may be zero for transactions without a SubstreamID.

conf_reset_of_SMMU_S_GBPA_ABORT

The pin is driven. This is the reset value of SMMU_S_GBPA.ABORT. This pin is sampled on the negedge of the reset pin and has to be driven beforehand for the value to take effect. Fields:

value bool

The value of the signal.

dpttlb_entry_allocated

A DPT TLB entry has been allocated. Fields:

AC enum

The value of the 'AC' field that controls access to this region.

FWB bool

The region is FWB.

VMID unsigned int

The VMID, if any, associated with this region.

index unsigned int

Index of the TLB entry.

input_end_incl_address unsigned int

The end inclusive address of the input range that this matches.

input_start_address unsigned int

The start address of the input range that this matches.

output_pas enum

The output PAS of this DPT region.

ssd enum

The SSD of the streams this region captures.

trans_id unsigned int

The trans_id of the transaction that caused this allocation.

vmsa_formed bool

The entry was formed from VMSA information rather than from a DPT walk.

writeable bool

True if this region is writeable.

dpttlb_invalidate_intersects_but_does_not_cover_entry_range

ENCODED_SIZE < size of the region covered by the DPT entry so invalidation is not architecturally guaranteed. No invalidation is performed. Fields:

dpttlb_entry_id unsigned int

ID of the TLB entry.

entry_end_incl_address unsigned int

Last address covered by the TLB entry.

entry_start_address unsigned int

First address covered by the TLB entry.

invalidate_end_incl_address unsigned int

Last address covered by the invalidation range.

invalidate_start_address unsigned int

First address covered by the invalidation range.

ssd enum

The security state of the TLB entry.

dpttlb_overlapping_entries

Two DPT TLB entries are overlapping. Fields:

new_AC enum

The value of the 'AC' field that controls access to this region.

new_FWB bool

The region is FWB.

new_VMID unsigned int

The VMID, if any, associated with this region.

new_entry_index unsigned int

The entry index of the new DPT TLB entry.

new_entry_is_vmsa_formed bool

The new entry was inserted as a result of VMSA information.

new_input_end_incl_address unsigned int

The end inclusive address of the input range that this matches.

new_input_start_address unsigned int

The start address of the input range that this matches.

new_output_pas enum

The output PAS of this DPT region.

new_ssd enum

The SSD of the streams this region captures.

new_writeable bool

True if this region is writeable.

old_AC enum

The value of the 'AC' field that controls access to this region.

old_FWB bool

The region is FWB.

old_VMID unsigned int

The VMID, if any, associated with this region.

old_entry_index unsigned int

The entry index of the old DPT TLB entry.

old_entry_is_vmsa_formed bool

The old entry was inserted as a result of VMSA information.

old_input_end_incl_address unsigned int

The end inclusive address of the input range that this matches.

old_input_start_address unsigned int

The start address of the input range that this matches.

old_output_pas enum

The output PAS of this DPT region.

old_ssd enum

The SSD of the streams this region captures.

old_writeable bool

True if this region is writeable.

dvm_tlbinvalidate_complete

The DVM TLB Invalidate message completed. Fields:

id unsigned int

The unique id of this DVM message.

ok bool

The DVM message was OK.

dvm_tlbinvalidate_received

A DVM message for a TLB Invalidate has been received. Fields:

address unsigned int

The VA or IPA to use if match_address.

asid unsigned int

The ASID to match if match_asid.

by_ipa bool

The operation is for an IPA operation if match_address.

id unsigned int

The unique id of this DVM message.

ignored enum

The DVM message was ignored.

last_level bool

The operation is for last level if supported.

match_address bool

Match the address field.

match_asid bool

Match the asid field.

match_vmid bool

Match the vmid field.

num unsigned int

If a range operation, the NUM field. If a single-address operation this is 0.

prot enum

The protection level for which this TLB Invalidate will operate on.

security_world enum

The security world that this will apply to.

smmu_scale unsigned int

If a range operation, then the SCALE field with the meaning in the SMMU architecture which is different to the PE architecture. If a single-address operation this is 0.

stage1_only bool

The operation is for stage 1 only if supported.

tg enum

If a single-address or address-range operation, then the Translation Granule hint. Address-range operations always supply a Translation Granule.

translation_table_level enum

The leaf level of the translation table.

vmid unsigned int

The VMID to match if match_vmid.

found_tlb_entry_has_different_aset

Architecturally, a particular ASID either should be ASET0 or ASET1. However, we have managed to find a TLB entry that has a different ASET than that which we were searching for. This indicates a programming error. You should examine all contexts with this particular ASID/VMID and ensure they are consistent. Fields:

aset unsigned int

The ASET of this entry.

asid unsigned int

ASID if appropriate.

index unsigned int

Index of TLB entry.

input_start_address unsigned int

The start address of the input range that this matches.

scheme enum

The tagging scheme used.

ssd enum

The SSD of the tagging scheme.

ssd_ns enum

Security State.

vmid unsigned int

VMID if appropriate.

httu_update_abandoned_update

The HTTU update of a descriptor in memory was potentially possible, but it was behind an update that failed to apply cleanly. Fields:

AF enum

Whether an AF update should/could occur.

DBM enum

Whether a DBM update should/could occur.

address unsigned int

Address of HTTU update.

httu_inner enum

Inner cacheability of descriptor to update.

httu_outer enum

Outer cacheability of descriptor to update.

httu_prot enum

The instruction/data, bus-ns/s and privileged/user marking.

httu_sh enum

Shareability of descriptor to update.

mpam_pmg_and_partid unsigned int

Bits[23:16] are the MPAM PMG. Bits[15:0] are the physical PARTID.

mpam_sp enum

The MPAM_SP of the transaction.

original_descriptor unsigned int

The original descriptor that the model observed.

stage_and_level unsigned int

The stage (bits[7:4]) and level (bits[3:0]). A level of 0xF means -1.

trans_id unsigned int

The transaction id that this HTTU update is associated with.

what enum

What this descriptor represents.

httu_update_end_update

The attempted update of a descriptor in memory has occurred. Fields:

is_big_endian bool

The descriptor is big-endian in memory.

original_descriptor unsigned int

The original descriptor value.

result enum

The result of the attempt to update.

trans_id unsigned int

The transaction id that this HTTU update is associated with.

try_to_change_to_descriptor unsigned int

The new descriptor value that tried to replace the original.

value_that_was_in_memory unsigned int

The value that the compare-and-swap operation returned as the value that was in memory.

httu_update_not_done

A discretionary HTTU update could occur and the implementation choose not to do it. A mandatory_do_if_linked_performed means that this is a stage 2 leaf descriptor that maps a subsequent stage 1 leaf descriptor whose update is discretionary and if that subsequent discretionary update is going to occur then this update becomes mandatory. Fields:

AF enum

Whether an AF update should/could occur.

DBM enum

Whether a DBM update should/could occur.

address unsigned int

Address of HTTU update.

httu_inner enum

Inner cacheability of descriptor to update.

httu_outer enum

Outer cacheability of descriptor to update.

httu_prot enum

The instruction/data, bus-ns/s and privileged/user marking.

httu_sh enum

Shareability of descriptor to update.

mpam_pmg_and_partid unsigned int

Bits[23:16] are the MPAM PMG. Bits[15:0] are the physical PARTID.

mpam_sp enum

The MPAM_SP of the transaction.

original_descriptor unsigned int

The original descriptor that the model observed.

stage_and_level unsigned int

The stage (bits[7:4]) and level (bits[3:0]). A level of 0xF means -1.

trans_id unsigned int

The transaction id that this HTTU update is associated with.

what enum

What this descriptor represents.

httu_update_start_update

An HTTU update could occur and the implementation chose to try it. A mandatory_do_if_linked_performed means that this is a stage 2 leaf descriptor that maps a subsequent stage 1 leaf descriptor whose update is discretionary and if that subsequent discretionary update is going to occur then this update becomes mandatory. Fields:

AF enum

Whether an AF update should/could occur.

DBM enum

Whether a DBM update should/could occur.

address unsigned int

Address of HTTU update.

httu_inner enum

Inner cacheability of descriptor to update.

httu_outer enum

Outer cacheability of descriptor to update.

httu_prot enum

The instruction/data, bus-ns/s and privileged/user marking.

httu_sh enum

Shareability of descriptor to update.

is_big_endian bool

The descriptor will be written to memory as big-endian.

mecid unsigned int

The masked MECID used for the update transaction, or ~0u if not appropriate.

mpam_pmg_and_partid unsigned int

Bits[23:16] are the MPAM PMG. Bits[15:0] are the physical PARTID.

mpam_sp enum

The MPAM_SP of the transaction.

original_descriptor unsigned int

The original descriptor that the model observed.

stage_and_level unsigned int

The stage (bits[7:4]) and level (bits[3:0]). A level of 0xF means -1.

trans_id unsigned int

The transaction id that this HTTU update is associated with.

try_to_change_to_descriptor unsigned int

The new descriptor value that is going to try replace the original.

what enum

What this descriptor represents.

will_do_AF bool

What the implementation chose to do for the AF update.

will_do_DBM bool

What the implementation chose to do for the DBM update.

interrupt_returned

An interrupt/MSI returned from downstream. Fields:

InD_NS_PnU enum

Instruction/Data, Non-secure/Secure/Root/Realm, Privileged/User. Bit[1] = PAS[0], and bit[3] = PAS[1].

address unsigned int

Address of the MSI.

data unsigned int

Data payload of the MSI.

dcmdqcp_and_index unsigned int

If this not a DCMDQ MSI CMD_SYNC, then this is 0xFFFFFFFF. If it is then the DCMDQ QCP index is in bits[31:16] and the index in the QCP is in [7:0].

gpc_denied_msi bool

True if the MSI was denied as it failed its GPC checks. Thus the field 'ok' will be false.

id unsigned int

ID of this interrupt transaction.

inner_cache enum

The actual attributes of the access.

kind enum

What kind of interrupt.

mecid unsigned int

The MECID of the MSI, if applicable.

mpam_ns bool

For a non-RME system, this is true if the MPAM space is Non-secure. For an RME system, this is MPAM_SP[0].

mpam_pmg_and_partid unsigned int

Bits[23:16] are the MPAM PMG. Bits[15:0] are the physical PARTID.

mpam_sp enum

The MPAM_SP security state of the MPAM partition. 0 – secure, 1 – non-secure, 2 – root, 3 – realm.

msi_valid bool

MSI will attempt to be sent.

ok bool

Did the access return OK or an abort?.

outer_cache enum

The actual attributes of the access.

pas enum

The PAS of the MSI, if applicable.

pmcg_index_and_counter unsigned int

If this is a PMCG interrupt, then the top 16 bits are the PMCG index, the lower 16 bits are counter index.

ras_error_group_id_node_id_and_record_index_in_node unsigned int

If this is a RAS interrupt then this is the error group record id in bits[23:16], the node_id in [15:8] and the record index in the node in [7:0].

sh enum

Shareability.

interrupt_sent

An interrupt is raised. If it sends an MSI then this is *after* any device-dependent transform on the architectural attributes and so may differ from what is programmed. Fields:

InD_NS_PnU enum

Instruction/Data, Non-secure/Secure/Root/Realm, Privileged/User. Bit[1] = PAS[0], and bit[3] = PAS[1].

address unsigned int

Address of the MSI.

data unsigned int

Data payload of the MSI.

dcmdq_device_id unsigned int

If this is a DCMDQ then this is the DeviceID of the MSI write. Otherwise, 0xFFFF'ffff.

dcmdq_qcp_and_index unsigned int

If this not a DCMDQ MSI CMD_SYNC, then this is 0xFFFF'ffff. If it is then the DCMDQ QCP index is in bits[31:16] and the index in the QCP is in [7:0].

id unsigned int

ID of this interrupt transaction.

inner_cache enum

The actual attributes of the access.

kind enum

What kind of interrupt.

mecid unsigned int

The MECID of the MSI, if applicable.

mpam_ns bool

For a non-RME system, this is true if the MPAM space is Non-secure. For an RME system, this is MPAM_SP[0].

mpam_pmg_and_partid unsigned int

Bits[23:16] are the MPAM PMG. Bits[15:0] are the physical PARTID.

mpam_sp enum

The MPAM_SP security state of the MPAM partition. 0 – secure, 1 – non-secure, 2 – root, 3 – realm.

msi_valid bool

MSI will attempt to be sent.

outer_cache enum

The actual attributes of the access.

pas enum

The PAS of the MSI, if applicable.

pmcg_index_and_counter unsigned int

If this is a PMCG interrupt, then the top 16 bits are the PMCG index, the lower 16 bits are counter index.

qSID unsigned int

If this is a DCMDQ then this is the qSID of the MSI write. Otherwise, 0xFFFF'ffff.

ras_error_group_id_node_id_and_record_index_in_node unsigned int

If this is a RAS interrupt then this is the error group record id in bits[23:16], the node_id in [15:8] and the record index in the node in [7:0].

sh enum

Shareability.

l1cd_cc.L1CD_commentary

A commentary from the cache about what is going on. Fields:

output string

The stream output.

l1cd_cc.L1CD_entry_invalidated

An entry in the cache has been invalidated. Fields:

entry_id unsigned int

The entry id that is being invalidated.

reason enum

Why the entry is being invalidated.

l1cd_entry_allocated

An L1 CD entry has been allocated. Fields:

AssuredTranslation bool

The L1CD was fetched from an AssuredOnly stage 2 page and so might be capable of creating AssuredTranslations.

entry_id unsigned int

The id of the cache entry. Top bit is set if is a debug entry.

normalised_substreamid unsigned int

The first SubstreamID of the range of SubstreamIDs that this L1CD entry will match.

ns enum

For the non-secure world.

pa_l2 unsigned int

The PA of the L2 CD table. This is L2Ptr << 12.

ssd enum

The SSD of the entry.

streamid unsigned int

The StreamID this CD is for.

valid bool

Is the entry valid.

l1ste_cc.L1STE_commentary

A commentary from the cache about what is going on. Fields:

output string

The stream output.

l1ste_cc.l1ste_entry_invalidated

An entry in the cache has been invalidated. Fields:

entry_id unsigned int

The entry id that is being invalidated.

reason enum

Why the entry is being invalidated.

l1ste_entry_allocated

An L1 STE entry has been allocated. Fields:

entry_id unsigned int

The id of the cache entry. Top bit is set if is a debug entry.

first_streamid_of_range unsigned int

The first StreamID of the range of StreamIDs that this L1STE entry will match.

ns enum

For the non-secure world.

num_entries_in_l2 unsigned int

The number of entries in the L2 table. This is $2^{(\text{Span}-1)}$ or 0 if invalid.

pa_l2 unsigned int

The PA of the L2 ST table. This is $\text{L2Ptr} \ll 6$ and aligned to the size of the table.

ssd enum

The SSD of the entry.

level_interrupt_sent

A level interrupt changed state. Fields:

kind enum

What kind of interrupt.

ras_error_group_id_node_id_and_record_index_in_node unsigned int

If this is a level RAS interrupt then this is the error group record id in bits[23:16], the node_id in [15:8] and the record index in the node in [7:0].

set_high bool

Level interrupt state.

mmu700_integration_mode_end_ras_level_interrupt_restored

RAS level sensitive interrupt restored due to integration mode ending. Fields:

is_tcu bool

Is TCU integration mode.

ras_interrupt enum

RAS interrupt being restored.

tbu_index unsigned int

TBU index. ~0u if it's TCU.

value bool

Level of the interrupt signal.

mmu700_integration_mode_pmcg_interrupt_lost

PMCG interrupt lost due to being in integration mode. Fields:

is_tcu enum

Is TCU integration mode.

pmcg_interrupt enum

PMCG interrupt being dropped.

tbu_index unsigned int

TBU index. ~0u if it's TCU.

mmu700_integration_mode_pmusnapshot_ack_lost

pmusnapshot_ack lost due to being in integration mode. Fields:

is_tcu enum

Is TCU integration mode.

tbu_index unsigned int

TBU index. ~0u if it's TCU.

mmu700_integration_mode_ras_interrupt_lost

RAS interrupt lost due to being in integration mode. Fields:

is_tcu enum

Is TCU integration mode.

ras_interrupt enum

RAS interrupt being dropped.

tbu_index unsigned int

TBU index. ~0u if it's TCU.

mmu700_integration_mode_ras_level_interrupt_lost

RAS level sensitive interrupt lost due to being in integration mode. Fields:

is_tcu bool

Is TCU integration mode.

ras_interrupt enum

RAS interrupt being dropped.

tbu_index unsigned int

TBU index. ~0u if it's TCU.

value bool

Level of the interrupt signal.

mmu700_integration_mode_start_ras_level_interrupt_cleared

RAS level sensitive interrupt cleared due to integration mode starting. Fields:

is_tcu bool

Is TCU integration mode.

ras_interrupt enum

RAS interrupt being cleared.

tbu_index unsigned int

TBU index. ~0u if it's TCU.

mmu700_integration_mode_tcu_evento_lost

Evento lost due to being in integration mode.

mmu700_integration_mode_tcu_interrupt_lost

Interrupt lost due to being in integration mode. Fields:

interrupt enum

Interrupt that is being dropped.

mmu700_integration_register_change

An integration register write occurred and it's modified, which drives a signal. Fields:

is_tcu enum

Is TCU integration mode.

new_value bool

New value of the signal.

old_value bool

Old value of the signal.

register enum

The register being modified.

register_value unsigned int

Value written to the register.

signal string

Signal affected by the register write.

tbu_index unsigned int

TBU index. ~0u if it's TCU.

ns_cmd_sync_completed_irq

"Non-secure" CMD_SYNC completed its IRQ action. Fields:

cmd_id unsigned int

The ID of the command being issued.

cons_incl_wrap unsigned int

CONS pointer of CMD_SYNC.

is_synthetic bool

This is a synthetic sync. The cons_incl_wrap is 0xFFFF'ffff.

ns_cmd_sync_completed_no_action

Non-secure CMD_SYNC completed, there was no CMD_SYNC action (SIG_NONE). Fields:

cmd_id unsigned int

The ID of the command being issued.

cons_incl_wrap unsigned int

CONS pointer of CMD_SYNC.

is_synthetic bool

This is a synthetic sync. The cons_incl_wrap is 0xFFFF'ffff.

ns_cmd_sync_completed_sev

Non-secure CMD_SYNC completed its action of SIG_SEV. Fields:

cmd_id unsigned int

The ID of the command being issued.

cons_incl_wrap unsigned int

CONS pointer of CMD_SYNC.

is_synthetic bool

This is a synthetic sync. The cons_incl_wrap is 0xFFFF'ffff.

ns_cmd_sync_completed_with_error

Non-secure CMD_SYNC completed, however, there was an error associated with the completion of the CMD_SYNC. Fields:

error enum

The error associated with the CMD_SYNC.

cmd_id unsigned int

The ID of the command being issued.

cons_incl_wrap unsigned int

CONS pointer of CMD_SYNC.

is_synthetic bool

This is a synthetic sync. The cons_incl_wrap is 0xFFFF'ffff.

ns_cmd_sync_issued

Non-secure CMD_SYNC has now completed issuing and we now know if something is hazarding the CMD_SYNC from completing. Fields:

cmd_id unsigned int

The ID of the command being issued.

cons_incl_wrap unsigned int

CONS pointer of CMD_SYNC.

hazarded bool

The CMD_SYNC is currently hazarded from completing.

is_synthetic bool

This is a synthetic sync. The cons_incl_wrap is 0xFFFF'ffff.

ns_cmd_sync_issuing

Non-secure CMD_SYNC is starting to issue, we are going to start accounting for which things have to be part of the prior set of things that have to complete before the CMD_SYNC can complete. Fields:

cmd_id unsigned int

The ID of the command being issued.

cons_incl_wrap unsigned int

CONS pointer of CMD_SYNC.

is_synthetic bool

This is a synthetic sync. The cons_incl_wrap is 0xFFFF'ffff.

ns_cmd_sync_starting_completion_action_irq

Non-secure CMD_SYNC starting IRQ completion action. Fields:

cmd_id unsigned int

The ID of the command being issued.

cons_incl_wrap unsigned int

CONS pointer of CMD_SYNC.

is_synthetic bool

This is a synthetic sync. The cons_incl_wrap is 0xFFFF'ffff.

msiaddress unsigned int

The MSI address to use (or 0 if not sending an MSI).

msidata unsigned int

The MSI data to send.

ns_cmd_sync_starting_completion_action_sev

Non-secure CMD_SYNC starting its completion action of SIG_SEV. Fields:

cmd_id unsigned int

The ID of the command being issued.

cons_incl_wrap unsigned int

CONS pointer of CMD_SYNC.

is_synthetic bool

This is a synthetic sync. The cons_incl_wrap is 0xFFFF'ffff.

ns_cmdq_cmd_sync_error

The CMD_SYNC can fault if an ATC Invalidate timeout occurs. The timeout will be attached to the next CMD_SYNC on the queue on which the invalidate was launched. Fields:

cerror enum

The CERROR to attach.

cmd_queue unsigned int

The top bits indicate the CMDQ id. These bits are attached to all cmd_ids generated for this queue. The bits [63:44] contain the queue ID. [63:62] 0: s-ssd, 1: reserved, 2: ns-ssd. 3: rl-ssd [61] is ECMDQ [59:52] QCP index [51:44] ECMDQ index in QCP.

herror enum

The HERROR to attach, or HERROR_NONE if we are attaching a CERROR.

ns_cmdq_cmd_sync_error_delivered

The CMD_SYNC can fault if an ATC Invalidate timeout occurs. This is the CMD_SYNC delivering that fault. Fields:

cerror enum

The CERROR to attach.

cmd_id unsigned int

Command ID of the CMD_SYNC.

ns_cmdq_ecmdq_enack

Trace the changes to ENACK. Fields:

cerror enum

The error being exposed, if any.

cmd_queue unsigned int

The top bits indicate the CMDQ id. These bits are attached to all cmd_ids generated for this queue. The bits [63:44] contain the queue ID. [63:62] 0: s-ssd, 1: reserved, 2: ns-ssd. 3: rl-ssd [61] is ECMDQ [59:52] QCP index [51:44] ECMDQ index in QCP.

enack bool

The value of ENACK.

forced_error bool

The queue was enabled with ERR != ERRACK and so the implementation chooses to trigger the error reporting mechanism.

ns_cmdq_issue

A command is actually being executed. Fields:

cmd_id unsigned int

Command ID of the command being executed.

cons unsigned int

CONS of the command.

what string

What is being executed.

ns_cmdq_pointers_state

The command queue pointers. Fields:

cmd_queue unsigned int

The top bits indicate the CMDQ id. These bits are attached to all cmd_ids generated for this queue. The bits [63:44] contain the queue ID. [63:62] 0: s-ssd, 1: reserved, 2: ns-ssd. 3: rl-ssd [61] is ECMDQ [59:52] QCP index [51:44] ECMDQ index in QCP.

disable_fetch bool

The model has decided that as you programmed PROD/CONS inconsistently that it would not fetch as it has no confidence as to which are commands and which are not.

q_issuable_and_issued_cons unsigned int

The CONS pointer of the region that is issuable and/or issued (including wrap).

q_issuable_and_issued_prod unsigned int

The PROD pointer of the region that is issuable and/or issued (including wrap).

q_programmer_cons unsigned int

The CONS pointer as seen by the programmer (including wrap).

q_programmer_prod unsigned int

The PROD pointer as seen by the programmer (including wrap).

q_unfetched_cons unsigned int

The CONS pointer of the region unfetched (including wrap).

q_unfetched_prod unsigned int

The PROD pointer of the region unfetched (including wrap).

ns_cmdq_state

A description of what the CMDQ can do now. Fields:

can_cerror bool

There is a CERROR that can be recognised.

can_cmd_sync bool

There is a CMD_SYNC that can be recognised.

can_fetch bool

The SMMU can fetch.

can_herror bool

There is an HERROR that can be recognised.

can_issue bool

There are fetched commands that are eligible to be issued.

can_update_cons bool

The programmer view of CMDQ_CONS can be moved to say some have been consumed.

cmd_queue unsigned int

The top bits indicate the CMDQ id. These bits are attached to all cmd_ids generated for this queue. The bits [63:44] contain the queue ID. [63:62] 0: s-ssd, 1: reserved, 2: ns-ssd. 3: rl-ssd [61] is ECMDQ [59:52] QCP index [51:44] ECMDQ index in QCP.

cmd_sync enum

Why a CMD_SYNC cannot be issued.

cmdqen bool

CMDQEN, when enabled then the CMDQ can fetch commands.

cmdqen_1_to_0_unacknowledged bool

CMDQEN has been set to 0, but we have yet to acknowledge it.

current-real-cerror enum

The current CERROR that would be visible to the programmer if there was an error.

fetch-disabled_due_to_programmer_error bool

The PROD/CONS pointer was inconsistent and the model decided to stop processing commands.

fetch-number_of_unfetched_commands unsigned int

The number of commands that have yet to be fetched from memory, or ~0u if CMDQEN == 0.

fetch-state enum

The state of the fetch state machine.

fetch-too_many_commands_outstanding_so_not_fetching_any_more bool

If there are too many issued commands, or pending to be issued then we won't fetch any more.

issue-number_of_issuable_commands unsigned int

The number of commands that have been fetched but not yet issued.

issue-number_of_unissuable_commands unsigned int

This might include commands behind an illegal command, or behind a CMD_SYNC. Under some circumstances this can include a command representing an external abort.

issue-state enum

The internal issued state machine state.

outstanding_fetches_since_reset unsigned int

The number of outstanding fetches.

synthetic_sync_can_be_completed bool

The synthetic sync that is outstanding can be completed.

waiting_for_synthetic_sync bool

There is a synthetic sync that is outstanding.

ns_eventq_adding_to_cmd_sync_prior_set

The event created by the specified transaction id must become visible to the programmer before the CMD_SYNC can complete. Fields:

trans_id unsigned int

The transaction id that caused the event.

ns_eventq_auto_retry_stalled_transaction

A transaction that was stalled but not yet reported on the Event Queue was retried. The retry can happen if the transaction didn't write because it was de-duplicated behind a reported stalling transaction and that stalling transaction was CMD_RESUMEd, or if the queue was unwriteable and the queue is now writeable. Fields:

trans_id unsigned int

Transaction id of the auto-retried transaction.

why enum

Why the transaction retried.

ns_eventq_cmd_sync_unhazardred

The CMD_SYNC has been unhazardred as the prior set is empty and as much of it as required is visible in the programmers' view. Fields:

prod_incl_wrap unsigned int

The prod index including wrap bit that was required to become visible.

prog_prod_incl_wrap unsigned int

The current programmer-visible prod including wrap bit.

ns_eventq_eventqen_and_eventqenack

Trace the values of EVENTQEN and EVENTQENACK as they change. Fields:

about enum

Some more information about the queue state.

cons_incl_wrap unsigned int

The CONS pointer (including wrap bits).

eventqen bool

EVENTQEN.

eventqenack bool

EVENTQENACK.

prod_incl_wrap unsigned int

The PROD pointer (including wrap bits).

ns_eventq_external_abort

A particular eventq record got an external abort. The event will subsequently appear in the ns/s_eventq_lost_event_records trace. Fields:

prod unsigned int

The prod pointer (excluding wrap bit) of the event record that aborted.

trans_id unsigned int

The transaction id that produced the event record that aborted.

ns_eventq_lost_event_records

Event records that are lost and never record appear on this trace. Fields:

dword0 unsigned int

Double word 0 of event.

dword1 unsigned int

Double word 1 of event.

dword2 unsigned int

Double word 2 of event.

dword3 unsigned int

Double word 3 of event.

event enum

Event code that was thrown away.

prod unsigned int

The PROD index of the event if appropriate or ~0u otherwise.

reason enum

The reason the record was lost.

trans_id unsigned int

The transaction id that this corresponds to.

ns_eventq_overflow_acknowledged

When events are lost then they appear on the eventq_lost_event_records_* trace. This will activate the overflow flag in the corresponding SMMU_s_EVENTQ_PROD register. Software can then acknowledge that flag using the overflow acknowledge flag in SMMU_s_EVENTQ_CONS. Fields:

bad bool

If true then the programmer has attempted to acknowledge an overflow condition that didn't exist.

count_of_events_lost_due_to_overflow unsigned int

This is the number of events that were lost because the event queue was full. This does not include any lost because of an EVENTQ_ABT_ERR.

ns_eventq_pending_event_records

Event records that are pending to be dealt with by the eventq state machine. A specific event can be traced multiple times if it becomes part of the prior set of transactions of a CMD_SYNC that has to complete before the CMD_SYNC is allowed to be completed. Fields:

dword0 unsigned int

Double word 0 of event.

dword1 unsigned int

Double word 1 of event.

dword2 unsigned int

Double word 2 of event.

dword3 unsigned int

Double word 3 of event.

event enum

Event code.

is_prior_set bool

The transaction forms part of the prior set of a CMD_SYNC that have to become visible (or thrown away if necessary) before the CMD_SYNC can complete.

trans_id unsigned int

The transaction id that this corresponds to.

ns_eventq_prod_cons_updated

This records when the programmer visible PROD/CONS pointer is updated. More can be visible in memory than are advertised in the programmer-visible PROD pointer. This also traces the latest PROD that could have been advertised. Fields:

mem_prod unsigned int

The PROD pointer value for all records that are visible in the memory.

num_events_available unsigned int

The number of events available to the programmer.

prog_cons unsigned int

The programmer-visible CONS pointer.

prog_prod unsigned int

The programmer-visible PROD pointer.

what enum

What changed?.

ns_eventq_raising_eventq_abt_err

EVENTQ_ABT_ERR can now be raised.

ns_eventq_removing_from_cmd_sync_prior_set

The event captured by a CMD_SYNC and created by the specified transaction id is now in memory. Fields:

prod unsigned int

The prod index (no wrap bit) that must become visible to the programmer to see this record. If not head_of_line then the record will not become visible and this field is meaningless.

reason enum

The reason the event was removed.

trans_id unsigned int

The transaction id that caused the event.

ns_eventq_stashing_unreported_stalled_transaction_for_auto_retry

A stall event record would have been lost, instead it is stashed away to auto-retry when it can. Fields:

dword0 unsigned int

Double word 0 of event.

dword1 unsigned int

Double word 1 of event.

dword2 unsigned int

Double word 2 of event.

dword3 unsigned int

Double word 3 of event.

event enum

Event code that was thrown away.

reason enum

The reason the record was lost.

trans_id unsigned int

The transaction id that this corresponds to.

ns_eventq_write_event

An event is committed to being written. Fields:

dword0 unsigned int

Double word 0 of event.

dword1 unsigned int

Double word 1 of event.

dword2 unsigned int

Double word 2 of event.

dword3 unsigned int

Double word 3 of event.

event enum

Event code that was thrown away.

prod unsigned int

The PROD pointer (without wrap) that this event will be written to.

stall_event bool

Describes an event for a stalled transaction.

trans_id unsigned int

Transaction ID that generated this event.

ns_eventq_write_event_visible_in_memory

An event is now visible in memory. Fields:

abort bool

The write aborted.

dword0 unsigned int

Double word 0 of event.

dword1 unsigned int

Double word 1 of event.

dword2 unsigned int

Double word 2 of event.

dword3 unsigned int

Double word 3 of event.

event enum

Event code that was thrown away.

prod unsigned int

The PROD pointer (without wrap) that this event will be written to.

stall_event bool

Describes an event for a stalled transaction.

trans_id unsigned int

Transaction ID that generated this event.

ns_gerror_inverted

A GERROR bit was attempted to be inverted to record a fault. Fields:

already_different_to_gerrorn bool

True if the bit was already different to the corresponding bit in GERRORN and so it was not inverted.

gerrorn unsigned int

GERRORN register value.

interrupt_potentially_sent bool

An interrupt is potentially sent, depending on IRQEN.
GERROR(N).MSI_GERROR_ABT_ERR does not send an interrupt.

new_gerror unsigned int

GERROR register value after any inversion.

which enum

Which bit was attempted to be inverted.

ns_gerrorn_acknowledge

A GERROR was acknowledged by SW. Fields:

which enum

Which GERROR was acknowledged.

pmcg_irq_config

The interrupt configuration of the Performance Monitor Counter Group (PMCG) changed.
Fields:

address unsigned int

Address of the MSI.

data unsigned int

Data payload of the MSI.

irqen bool

SMMU_PMCG_CTRL.IRQEN.

memattr enum

Memory type.

mpam_ns bool

The NS state of the MPAM PARTID and MPAM PMG.

mpam_pmg_and_partid unsigned int

Bits[23:16] are the MPAM PMG. Bits[15:0] are the physical PARTID.

msi_supported bool

Are MSIs supported for this security world.

ns bool

Non-secure bus attribute.

number_of_interrupts_in_flight unsigned int

The number of interrupts that have been committed to be produced or in flight.

pmcg_index unsigned int

Index of the PMCG.

sh enum

Shareability.

smmu_pmcg_gmpam_Update bool

The SMMU_PMCG_GMPAM.Update flag. Only when this is zero are writes predictable.

ssd_ns bool

True if this PMCG controlled by the non-secure world.

pmcg_irqen

A trace of SMMU_PMCG_IRQ_CTRL.IRQEN. Fields:

new_value bool

The new value of IRQEN.

pmcg_index unsigned int

Index of the PMCG.

ssd_ns bool

True if this PMCG controlled by the non-secure world.

pmcg_irqen_ack

A trace of SMMU_PMCg_IRQ_CTRLACK.IRQEN. Fields:

new_value bool

The new value of IRQEN.

pmcg_index unsigned int

Index of the PMCG.

ssd_ns bool

True if this PMCG controlled by the non-secure world.

pmcg_merging_interrupts

An interrupt was wanted to be generated, but one was already pending so the two were merged together. Fields:

pmcg_index unsigned int

Index of the PMCG.

ssd_ns bool

True if this PMCG controlled by the non-secure world.

pmcg_trigger

A PMCG counter has been triggered. Fields:

NoStreamID bool

True if the transaction is a NoStreamID transaction.

counter_index unsigned int

The index of the counter within the PMCG.

event_id unsigned int

The event id that has been triggered.

ns_event bool

Is the event associated with non-secure state.

pmcg_index unsigned int

Index of the PMCG.

prior_counter_value unsigned int

The Counter value *before* the event has incremented it.

ssd enum

The security state associated with the event.

streamid unsigned int

The StreamID associated with the event, if there is one.

tbu_index unsigned int

The TBU index of the transaction, or ~0u if not applicable.

pmu_active_counter

Traces what active counters are in a PMCG and what StreamIDs it might filter on. Those counters that trace StreamIDs for multiple security states, or those that are not filtered by StreamID, will appear multiple times, once for each security state. All active counters for a PMCG are traced one after another. Fields:

NoStreamID bool

True if NoStreamID transactions will be traced.

begin_streamid unsigned int

The start StreamID to filter on.

counter_index unsigned int

The counter index within the PMCG.

end_incl_streamid unsigned int

The end inclusive StreamID to filter on.

evcnt unsigned int

The current count.

event_id unsigned int

The event ID to filter.

ns bool

Are the StreamIDs non-secure?.

pmcg_index unsigned int

The index of the PMCG that this counter belongs to.

ssd enum

SSD of the StreamID.

tbu_index_to_match unsigned int

The TBU index that must match, or ~0u if no matching applicable.

pmu_all_counters_in_pmcg_became_inactive

The PMCG was tracing some events and now is not tracing any. Fields:

pmcg_index unsigned int

The index of the PMCG.

pmu_capture

For some reason, a capture event occurred. Fields:

pmcg_index unsigned int

The index of the PMCG that the capture occurred on.

why enum

Why did the capture occur?.

pmu_counter_configured_to_use_unsupported_event

An enabled counter was configured to use a unsupported event. Fields:

counter_index unsigned int

The counter index within the PMCG.

event_id unsigned int

The unsupported event id.

pmcg_index unsigned int

The index of the PMCG that this counter belongs to.

pmu_counter_overflowed

A counter in a particular PMCG overflowed. Fields:

already_overflowed bool

True if the overflow flag was already set.

capture bool

True if it captured the other counter values.

counter_index unsigned int

The counter index within the PMCG.

interrupt bool

True if going to attempt to generate an interrupt.

interrupt_action enum

The interrupt action that is going to occur.

pmcg_index unsigned int

The index of the PMCG that this counter belongs to.

ssd_ns bool

The PMCG is controlled by the Non-secure security state.

priq_auto_response

A PRIQ auto response is generated. Fields:

prgindex unsigned int

The PRG Index of the response.

resp enum

The response to send.

streamid unsigned int

The StreamID of the response.

substreamid unsigned int

The PASID/SubstreamID attached to the response, or ~0u if there is none.

trans_id unsigned int

The ID of the PPR that is being lost.

why enum

Why the PPR is being lost.

priq_lost_ppr

The PRIQ is not reporting a PPR into the queue. It may or may not generate an auto-response. Fields:

is_stop_marker bool

The PPR being lost is a Stop Marker.

resp enum

What, if any, auto-response will be.

trans_id unsigned int

The ID of the PPR that is being lost.

why enum

Why the PPR is being lost.

priq_overflow_acking

Indicates that an overflow condition was acknowledged by writing to:- SMMU_PRIQ_CONS.OVACKFLG. Fields:

new_ovackflg bool

The new value of the SMMU_PRIQ_CONS.OVACKFLG.

priq_overflow_asserting

Indicates that we are toggling the SMMU_PRIQ_PROD.OVFLG because we lost a PRI request due to the PRIQ being full and an existing overflow condition does not already exist. Fields:

new_ovflg bool

The new value of the SMMU_PRIQ_PROD.OVFLG.

trans_id unsigned int

The transaction ID of the PPR that caused the overflow.

priq_received

A PRIQ Request has been received and is queued waiting processing. This does not mean that it has been written to the PRIQ but has been placed in a pending queue awaiting a decision about what to do about it. Fields:

L bool

The Last in the Page Request Group (PRG).

R bool

Read permission is requested.

W bool

Read permission is requested.

prgindex unsigned int

The Page Request Group (PRG) index that is being used to identify this request to the EndPoint.

stop_marker bool

If LRW == 0b100 then this indicates this is a stop marker.

streamid unsigned int

The StreamID this request is associated with.

substreamid unsigned int

The SubstreamID (PASID) this request is associated with, of ~0u if no SubstreamID.

substreamid_P bool

If has a SubstreamID (!= ~0u) then Privileged Mode Requested.

substreamid_X bool

If has a SubstreamID (!= ~0u) then Execution Requested (R must also be 1).

trans_id unsigned int

The transaction ID that identifies this request.

untranslated_address unsigned int

The Untranslated Address that this is requesting.

priq_state

The trace of various fields that indicate the state of the PRIQ. Fields:

cons_incl_wrap unsigned int

The value of SMMU_PRIQ.CONS.RD_and_RD_wrap.

number_of_pprs unsigned int

The number of PPRs as indicated by the CONS/PROD.

number_of_pprs_still_to_deal_with unsigned int

This is the number of PPRs that are currently waiting to either be written to the PRIQ, or auto-responded to.

number_of_priq_writes_in_flight unsigned int

The number of writes to the PRIQ that are currently in flight.

ovackflg bool

The OVACKFLG which if different to OVFLG is used to indicate that the PRIQ overflowed.

ovflg bool

The OVFLG which if different to OVACKFLG is used to indicate that the PRIQ overflowed.

priq_abt_err bool

There is an active SMMU_GERROR{N}.PRIQ_ABT_ERR.

priqen bool

The value of SMMU_CRO.PRIQEN. The *effective* value is 0 if SMMUEN == 0.

priqenack bool

The value of SMMU_CROACK.PRIQEN.

prod_incl_wrap unsigned int

The value of SMMU_PRIQ.CONS.WR_and_WR_wrap.

queue_disabled_due_to_prior_programming_error bool

The queue was disabled as the programmer got CONS/PROD into an inconsistent state. The model will disable the PRIQ until SW disables and re-enables the queue via SMMU_CR0.PRIQEN.

smmuen bool

The value of SMMU_CR0.SMMUEN. If this is 0 then the effective PRIQEN is 0.

smmuenack bool

The value of SMMU_CR0.SMMUEN. If this is 0 then the effective PRIQEN is 0.

table_size_in_elements unsigned int

The size of the table in the number of items it can hold.

priq_write_aborted

A PRIQ write aborted. The PRIQ now goes into an error state and will start auto-responding to PRI requests. Fields:

prod_incl_wrap unsigned int

The PROD including the Wrap bit where we were trying to write.

trans_id unsigned int

The transaction ID of the PPR that aborted.

priq_write_ok

A PRIQ write completed OK. Fields:

prod_incl_wrap unsigned int

The PROD including the Wrap bit where we wrote..

trans_id unsigned int

The transaction ID of the PPR.

priq_write_start

A PRIQ request has been received and is going to be attempt to be written to the queue. Fields:

L bool

The Last in the Page Request Group (PRG).

R bool

Read permission is requested.

W bool

Read permission is requested.

prgindex unsigned int

The Page Request Group (PRG) index that is being used to identify this request to the EndPoint.

prod_incl_wrap unsigned int

PROD position including the wrap bit.

stop_marker bool

If LRW == 0b100 then this indicates this is a stop marker.

streamid unsigned int

The StreamID this request is associated with.

substreamid unsigned int

The SubstreamID (PASID) this request is associated with, of ~0u if no SubstreamID.

substreamid_P bool

If has a SubstreamID (!= ~0u) then Privileged Mode Requested.

substreamid_X bool

If has a SubstreamID (!= ~0u) then Execution Requested (R must also be 1).

trans_id unsigned int

The transaction ID that identifies this request.

untranslated_address unsigned int

The Untranslated Address that this is requesting.

ptw_read

Page Table Walk (read). This is the result of the physical access that the SMMU is making.
Fields:

abort enum

Non-zero if the access aborted/failed.

adomain enum

The actual attributes of the access that was used, after IMP DEF mangling.

aprot enum

The actual attributes of the access that was used, after IMP DEF mangling. PAS[1] (bit[3]), Instruction/Data (bit[2]), PAS[0] (bit[1]), Privileged/User (bit[0]).

data unsigned int

The data fetch if it didn't abort.

desckind enum

The kind of descriptor.

inner_cache enum

The actual attributes of the access that was used, after IMP DEF mangling.

input_address unsigned int

Input address this lookup is for (after TBI normalisation).

mecid unsigned int

The MECID of the TTD fetch.

ns enum

Non-Secure on the bus.

outer_cache enum

The actual attributes of the access that was used, after IMP DEF mangling.

pa_address unsigned int

Physical address.

pas enum

The PAS of the TTD fetch.

ssd enum

SSD of the transaction.

ssd_ns enum

Security state of stream.

streamid unsigned int

The StreamID this is for.

substreamid unsigned int

The SubstreamID this is for.

trans_id unsigned int

Transaction id.

ttb_grain_stage_and_level unsigned int

Which TTB used in bits[31:24]. Which Grain size used in bit[23:16]:- 0 – 4 KiB 1 – 16 KiB 2 – 64 KiB 3 – 64 KiB ARMv8.2 where descriptor bits [15:12] for bits [51:48] of output address 4 – 4 KiB FEAT_LPA2 where address is {bits[9:8], bits[49:12]} 5 – 16 KiB FEAT_LPA2 where address is {bits[9:8], bits[49:14], 2'b00} 6 – 64 KiB ARMv8.2 where descriptor bits [15:12] are ignored Stage in bits[15:8]. Level in bits[7:0], level -1 is 0xF.

ptw_read_st1_invalid_long_descriptor

Page Table Walk (read). The descriptor that the PTW fetched is invalid. Fields:

desckind enum

The kind of descriptor.

input_address unsigned int

Input address this lookup is for (after TBI normalisation).

mecid unsigned int

The MECID of the TTD fetch.

ns enum

Non-Secure on the bus.

pa_address unsigned int

Physical address.

pas enum

The PAS of the TTD fetch.

ssd enum

SSD of the transaction.

ssd_ns enum

Security state of stream.

streamid unsigned int

The StreamID this is for.

substreamid unsigned int

The SubstreamID this is for.

trans_id unsigned int

Transaction id.

ttb_grain_stage_and_level unsigned int

Which TTB used in bits[31:24]. Which Grain size used in bit[23:16]:- 0 – 4 KiB 1 – 16 KiB 2 – 64 KiB 3 – 64 KiB ARMv8.2 where descriptor bits [15:12] for bits [51:48] of output address 4 – 4 KiB FEAT_LPA2 where address is {bits[9:8], bits[49:12]} 5 – 16 KiB FEAT_LPA2 where address is {bits[9:8], bits[49:14], 2'b00} 6 – 64 KiB ARMv8.2 where descriptor bits [15:12] are ignored Stage in bits[15:8]. Level in bits[7:0], level -1 is 0xF.

ptw_read_st1_leaf_long_descriptor

Page Table Walk (read). The descriptor that the PTW fetched is a block or page and this is the decode. Fields:

AF bool

Access Flag.

AMEC enum

Use Primary or Alternative MECID.

AP21 enum

The access permissions.

AttrIndx210 unsigned int

The attribute index into the MAIR0/1. If AIE is implemented then this is the full index AttrIndx[3:0].

DBM bool

Dirty Bit Modifier. This only has meaning if running in AArch64 mode.

NS bool

The encoding is for non-secure if this is a secure fetch.

PIE_Dirty enum

The Dirty bit if S2PIE is in use.

PIE_PIIIndex unsigned int

The S2PIIndex if S2PIE is in use, or 0xFFFF if not.

PXN bool

Privileged eXecute Never.

Protected enum

Is the descriptor producing an AssuredTranslation.

SH10 enum

The shareability.

XN bool

eXecute Never.

contiguous bool

Contiguous hint.

desckind enum

The kind of descriptor.

hwu_pbha unsigned int

Top four bits are appropriate CD.HWU, *bottom bits[62:59] of descriptor. Page Based Hardware Attributes: only valid on a bit-per-bit basis enabled by CD.HWU0/CD.HWU1**.

input_address unsigned int

Input address this lookup is for (after TBI normalisation).

mecid unsigned int

The MECID of the TTD fetch.

nG bool

not Global.

nT bool

SMMUv3.2: If true then do not cache this entry in such a way that it can cause a TLB conflict abort (F_TLB_CONFLICT). The entry must produce a consistent result.

ns enum

Non-Secure on the bus.

output_address unsigned int

Output address.

pa_address unsigned int

Physical address.

pas enum

The PAS of the TTD fetch.

ssd enum

SSD of the transaction.

ssd_ns enum

Security state of stream.

streamid unsigned int

The StreamID this is for.

substreamid unsigned int

The SubstreamID this is for.

trans_id unsigned int

Transaction id.

ttb_grain_stage_and_level unsigned int

Which TTB used in bits[31:24]. Which Grain size used in bit[23:16]:- 0 – 4 KiB 1 – 16 KiB 2 – 64 KiB 3 – 64 KiB ARMv8.2 where descriptor bits [15:12] for bits [51:48] of output address 4 – 4 KiB FEAT_LPA2 where address is {bits[9:8], bits[49:12]} 5 – 16 KiB FEAT_LPA2 where address is {bits[9:8], bits[49:14], 2'b00} 6 – 64 KiB ARMv8.2 where descriptor bits [15:12] are ignored Stage in bits[15:8]. Level in bits[7:0], level -1 is 0xF.

ptw_read_st1_table_long_descriptor

Page Table Walk (read). The descriptor that the PTW fetched is a Table, this decodes the fields. Fields:

AF enum

.

APTable enum

Remove permissions independently of subsequent descriptors.

NSTable bool

The next level table descriptor is forced to non-secure.

PXNTable bool

Force PXN independently of subsequent descriptors.

Protected enum

Is the descriptor capable of producing an AssuredTranslation.

TableAddress unsigned int

Address of the next table.

XNTable bool

Force XN independently of subsequent descriptors.

desckind enum

The kind of descriptor.

input_address unsigned int

Input address this lookup is for (after TBI normalisation).

mecid unsigned int

The MECID of the TTD fetch.

ns enum

Non-Secure on the bus.

pa_address unsigned int

Physical address.

pas enum

The PAS of the TTD fetch.

ssd enum

SSD of the transaction.

ssd_ns enum

Security state of stream.

streamid unsigned int

The StreamID this is for.

substreamid unsigned int

The SubstreamID this is for.

trans_id unsigned int

Transaction id.

tth_grain_stage_and_level unsigned int

Which TTB used in bits[31:24]. Which Grain size used in bit[23:16]:- 0 – 4 KiB 1 – 16 KiB 2 – 64 KiB 3 – 64 KiB ARMv8.2 where descriptor bits [15:12] for bits [51:48] of output address 4 – 4 KiB FEAT_LPA2 where address is {bits[9:8], bits[49:12]} 5 – 16 KiB FEAT_LPA2 where address is {bits[9:8], bits[49:14], 2'b00} 6 – 64 KiB ARMv8.2 where descriptor bits [15:12] are ignored Stage in bits[15:8]. Level in bits[7:0], level -1 is 0xF.

ptw_read_st2_invalid_descriptor

Page Table Walk (read). The descriptor that the PTW fetched is invalid. Fields:

desckind enum

The kind of descriptor.

input_address unsigned int

Input address this lookup is for (after TBI normalisation).

mecid unsigned int

The MECID of the TTD fetch.

ns enum

Non-Secure on the bus.

pa_address unsigned int

Physical address.

pas enum

The PAS of the TTD fetch.

ssd enum

SSD of the transaction.

ssd_ns enum

Security state of stream.

streamid unsigned int

The StreamID this is for.

substreamid unsigned int

The SubstreamID this is for.

trans_id unsigned int

Transaction id.

ttb_grain_stage_and_level unsigned int

Which TTB used in bits[31:24]. Which Grain size used in bit[23:16]:- 0 – 4 KiB 1 – 16 KiB 2 – 64 KiB 3 – 64 KiB ARMv8.2 where descriptor bits [15:12] for bits [51:48] of output address 4 – 4 KiB FEAT_LPA2 where address is {bits[9:8], bits[49:12]} 5 – 16 KiB FEAT_LPA2 where address is {bits[9:8], bits[49:14], 2'b00} 6 – 64 KiB ARMv8.2 where descriptor bits [15:12] are ignored Stage in bits[15:8]. Level in bits[7:0], level -1 is 0xF.

ptw_read_st2_leaf_descriptor

Page Table Walk (read). The descriptor that the PTW fetched is a block or page and this is the decode. Fields:

AF bool

Access Flag.

AMEC enum

Use Primary or Alternative MECID.

AssuredOnly enum

The descriptor is marked as AssuredOnly.

DBM bool

Dirty Bit Modifier. This only has meaning if running in AArch64 mode.

HAP21 enum

The access permissions.

MemAttr3_0 enum

The memory attributes.

NS enum

Whether this descriptor forces NS.

PIE_Dirty enum

The Dirty bit if S2PIE is in use.

PIE_PIndex unsigned int

The S2PIndex if S2PIE is in use, or 0xFFFF if not.

POE_POIndex unsigned int

The S2POIndex if S2POE is in use, or 0xFFFF if not.

SH10 enum

The shareability.

XN enum

eXecute Never.

contiguous bool

Contiguous hint.

desckind enum

The kind of descriptor.

input_address unsigned int

Input address this lookup is for (after TBI normalisation).

mecid unsigned int

The MECID of the TTD fetch.

nT bool

SMMUv3.2: If true then do not cache this entry in such a way that it can cause a TLB conflict abort (F_TLB_CONFLICT). The entry must produce a consistent result.

ns enum

Non-Secure on the bus.

output_address unsigned int

Output address.

pa_address unsigned int

Physical address.

pas enum

The PAS of the TTD fetch.

s2hwi_pbha unsigned int

Top four bits are STE.S2HWU, bottom bits[62:59] of descriptor. *Page Based Hardware Attributes: only valid on a bit-per-bit basis enabled by STE.S2HWU.*

ssd enum

SSD of the transaction.

ssd_ns enum

Security state of stream.

streamid unsigned int

The StreamID this is for.

substreamid unsigned int

The SubstreamID this is for.

trans_id unsigned int

Transaction id.

ttb_grain_stage_and_level unsigned int

Which TTB used in bits[31:24]. Which Grain size used in bit[23:16]:- 0 – 4 KiB 1 – 16 KiB 2 – 64 KiB 3 – 64 KiB ARMv8.2 where descriptor bits [15:12] for bits [51:48] of output address 4 – 4 KiB FEAT_LPA2 where address is {bits[9:8], bits[49:12]} 5 – 16 KiB FEAT_LPA2 where address is {bits[9:8], bits[49:14], 2'b00} 6 – 64 KiB ARMv8.2 where descriptor bits [15:12] are ignored Stage in bits[15:8]. Level in bits[7:0], level -1 is 0xF.

ptw_read_st2_table_descriptor

Page Table Walk (read). The descriptor that the PTW fetched is a Table, this decodes the fields. Fields:

AF enum

.

APTable enum

Remove permissions independently of subsequent descriptors.

PXNTable bool

Force PXN independently of subsequent descriptors.

TableAddress unsigned int

Address of the next table.

XNTable bool

Force XN independently of subsequent descriptors.

desckind enum

The kind of descriptor.

input_address unsigned int

Input address this lookup is for (after TBI normalisation).

mecid unsigned int

The MECID of the TTD fetch.

ns enum

Non-Secure on the bus.

pa_address unsigned int

Physical address.

pas enum

The PAS of the TTD fetch.

ssd enum

SSD of the transaction.

ssd_ns enum

Security state of stream.

streamid unsigned int

The StreamID this is for.

substreamid unsigned int

The SubstreamID this is for.

trans_id unsigned int

Transaction id.

tbtb_grain_stage_and_level unsigned int

Which TTB used in bits[31:24]. Which Grain size used in bit[23:16]:- 0 – 4 KiB 1 – 16 KiB 2 – 64 KiB 3 – 64 KiB ARMv8.2 where descriptor bits [15:12] for bits [51:48] of

output address 4 – 4 KiB FEAT_LPA2 where address is {bits[9:8], bits[49:12]} 5 – 16 KiB FEAT_LPA2 where address is {bits[9:8], bits[49:14], 2'b00} 6 – 64 KiB ARMv8.2 where descriptor bits [15:12] are ignored Stage in bits[15:8]. Level in bits[7:0], level -1 is 0xF.

raw_register_end_read

The raw register read transaction. This is the transaction as directed to the register port.

Fields:

address unsigned int

The address of the transaction.

id unsigned int

Id of this transaction.

ok bool

Was the access OK? The bus response will always be OK, but is the register access reasonable.

pas enum

The PAS of the transaction.

read_data unsigned int

The data read.

raw_register_end_write

The raw register write transaction. This is the transaction as directed to the register port.

Fields:

address unsigned int

The address of the transaction.

id unsigned int

Id of this transaction.

ok bool

Was the access OK? The bus response will always be OK, but is the register access reasonable.

pas enum

The PAS of the transaction.

raw_register_start_read

The raw register read transaction. This is the transaction as directed to the register port.

Fields:

address unsigned int

Address of transaction.

id unsigned int

Id of this transaction. Top bit is set if is a debug transaction.

ns enum

Is the transaction non-secure?.

number_of_beats unsigned int

Number of beats.

pas enum

The PAS of the transaction.

size_of_beat_in_bytes unsigned int

Size of each beat in bytes.

raw_register_start_write

The raw register write transaction. This is the transaction as directed to the register port.

Fields:

address unsigned int

Address of transaction.

id unsigned int

Id of this transaction. Top bit is set if is a debug transaction.

ns enum

Is the transaction non-secure?.

number_of_beats unsigned int

Number of beats.

pas enum

The PAS of the transaction.

size_of_beat_in_bytes unsigned int

Size of each beat in bytes.

write_data unsigned int

The data to write.

register_disallowed_read_string

A text representation of the read of a register that was disallowed. Fields:

out string

The text description of the register value read.

register_disallowed_write_string

A text representation of the write of a register write that was disallowed. Fields:

in string

The text description of the register value written.

register_read_reserved

A text representation of an access to a register address that is reserved. Fields:

in string

The text description of the register value.

register_read_string

A text representation of the read of a register. Fields:

out string

The text description of the register value read.

register_write_reserved

A text representation of an access to a register address that is reserved or a write to a **RES0** field in a register. Fields:

in string

The text description of the register value.

register_write_string

A text representation of the write of a register. Fields:

in string

The text description of the register value written.

rl_priq_auto_response

A PRIQ auto response is generated. Fields:

prgindex unsigned int

The PRG Index of the response.

resp enum

The response to send.

streamid unsigned int

The StreamID of the response.

substreamid unsigned int

The PASID/SubstreamID attached to the response, or ~0u if there is none.

trans_id unsigned int

The ID of the PPR that is being lost.

why enum

Why the PPR is being lost.

rl_priq_lost_ppr

The PRIQ is not reporting a PPR into the queue. It may or may not generate an auto-response. Fields:

is_stop_marker bool

The PPR being lost is a Stop Marker.

resp enum

What, if any, auto-response will be.

trans_id unsigned int

The ID of the PPR that is being lost.

why enum

Why the PPR is being lost.

rl_priq_received

A PRIQ Request has been received and is queued waiting processing. This does not mean that it has been written to the PRIQ but has been placed in a pending queue awaiting a decision about what to do about it. Fields:

L bool

The Last in the Page Request Group (PRG).

R bool

Read permission is requested.

W bool

Read permission is requested.

prgindex unsigned int

The Page Request Group (PRG) index that is being used to identify this request to the EndPoint.

stop_marker bool

If LRW == 0b100 then this indicates this is a stop marker.

streamid unsigned int

The StreamID this request is associated with.

substreamid unsigned int

The SubstreamID (PASID) this request is associated with, of ~0u if no SubstreamID.

substreamid_P bool

If has a SubstreamID (!= ~0u) then Privileged Mode Requested.

substreamid_X bool

If has a SubstreamID (!= ~0u) then Execution Requested (R must also be 1).

trans_id unsigned int

The transaction ID that identifies this request.

untranslated_address unsigned int

The Untranslated Address that this is requesting.

s_cmd_sync_completed_irq

“Secure” CMD_SYNC completed its IRQ action. Fields:

cmd_id unsigned int

The ID of the command being issued.

cons_incl_wrap unsigned int

CONS pointer of CMD_SYNC.

is_synthetic bool

This is a synthetic sync. The cons_incl_wrap is 0xFFFF'ffff.

s_cmd_sync_completed_no_action

Secure CMD_SYNC completed, there was no CMD_SYNC action (SIG_NONE). Fields:

cmd_id unsigned int

The ID of the command being issued.

cons_incl_wrap unsigned int

CONS pointer of CMD_SYNC.

is_synthetic bool

This is a synthetic sync. The cons_incl_wrap is 0xFFFF'ffff.

s_cmd_sync_completed_sev

Secure CMD_SYNC completed its action of SIG_SEV. Fields:

cmd_id unsigned int

The ID of the command being issued.

cons_incl_wrap unsigned int

CONS pointer of CMD_SYNC.

is_synthetic bool

This is a synthetic sync. The cons_incl_wrap is 0xFFFF'ffff.

s_cmd_sync_completed_with_error

Secure CMD_SYNC completed, however, there was an error associated with the completion of the CMD_SYNC. Fields:

error enum

The error associated with the CMD_SYNC.

cmd_id unsigned int

The ID of the command being issued.

cons_incl_wrap unsigned int

CONS pointer of CMD_SYNC.

is_synthetic bool

This is a synthetic sync. The cons_incl_wrap is 0xFFFF'ffff.

s_cmd_sync_issued

Secure CMD_SYNC has now completed issuing and we now know if something is hazarding the CMD_SYNC from completing. Fields:

cmd_id unsigned int

The ID of the command being issued.

cons_incl_wrap unsigned int

CONS pointer of CMD_SYNC.

hazarded bool

The CMD_SYNC is currently hazarded from completing.

is_synthetic bool

This is a synthetic sync. The cons_incl_wrap is 0xFFFF'ffff.

s_cmd_sync_issuing

Secure CMD_SYNC is starting to issue, we are going to start accounting for which things have to be part of the prior set of things that have to complete before the CMD_SYNC can complete. Fields:

cmd_id unsigned int

The ID of the command being issued.

cons_incl_wrap unsigned int

CONS pointer of CMD_SYNC.

is_synthetic bool

This is a synthetic sync. The cons_incl_wrap is 0xFFFF'ffff.

s_cmd_sync_starting_completion_action_irq

Secure CMD_SYNC starting IRQ completion action. Fields:

cmd_id unsigned int

The ID of the command being issued.

cons_incl_wrap unsigned int

CONS pointer of CMD_SYNC.

is_synthetic bool

This is a synthetic sync. The cons_incl_wrap is 0xFFFF'ffff.

msiaddress unsigned int

The MSI address to use (or 0 if not sending an MSI).

msidata unsigned int

The MSI data to send.

s_cmd_sync_starting_completion_action_sev

Secure CMD_SYNC starting its completion action of SIG_SEV. Fields:

cmd_id unsigned int

The ID of the command being issued.

cons_incl_wrap unsigned int

CONS pointer of CMD_SYNC.

is_synthetic bool

This is a synthetic sync. The cons_incl_wrap is 0xFFFF'ffff.

s_cmdq_cmd_sync_error

The CMD_SYNC can fault if an ATC Invalidate timeout occurs. The timeout will be attached to the next CMD_SYNC on the queue on which the invalidate was launched. Fields:

cerror enum

The CERROR to attach.

cmd_queue unsigned int

The top bits indicate the CMDQ id. These bits are attached to all cmd_ids generated for this queue. The bits [63:44] contain the queue ID. [63:62] 0: s-ssd, 1: reserved, 2: ns-ssd. 3: rl-ssd [61] is ECMDQ [59:52] QCP index [51:44] ECMDQ index in QCP.

herror enum

The HERROR to attach, or HERROR_NONE if we are attaching a CERROR.

s_cmdq_cmd_sync_error_delivered

The CMD_SYNC can fault if an ATC Invalidate timeout occurs. This is the CMD_SYNC delivering that fault. Fields:

cerror enum

The CERROR to attach.

cmd_id unsigned int

Command ID of the CMD_SYNC.

s_cmdq_ecmdq_enack

Trace the changes to ENACK. Fields:

cerror enum

The error being exposed, if any.

cmd_queue unsigned int

The top bits indicate the CMDQ id. These bits are attached to all cmd_ids generated for this queue. The bits [63:44] contain the queue ID. [63:62] 0: s-ssd, 1: reserved, 2: ns-ssd. 3: rl-ssd [61] is ECMDQ [59:52] QCP index [51:44] ECMDQ index in QCP.

enack bool

The value of ENACK.

forced_error bool

The queue was enabled with ERR != ERRACK and so the implementation chooses to trigger the error reporting mechanism.

s_cmdq_issue

A command is actually being executed. Fields:

cmd_id unsigned int

Command ID of the command being executed.

cons unsigned int

CONS of the command.

what string

What is being executed.

s_cmdq_pointers_state

The command queue pointers. Fields:

cmd_queue unsigned int

The top bits indicate the CMDQ id. These bits are attached to all cmd_ids generated for this queue. The bits [63:44] contain the queue ID. [63:62] 0: s-ssd, 1: reserved, 2: ns-ssd. 3: rl-ssd [61] is ECMDQ [59:52] QCP index [51:44] ECMDQ index in QCP.

disable_fetch bool

The model has decided that as you programmed PROD/CONS inconsistently that it would not fetch as it has no confidence as to which are commands and which are not.

q_issuable_and_issued_cons unsigned int

The CONS pointer of the region that is issuable and/or issued (including wrap).

q_issuable_and_issued_prod unsigned int

The PROD pointer of the region that is issuable and/or issued (including wrap).

q_programmer_cons unsigned int

The CONS pointer as seen by the programmer (including wrap).

q_programmer_prod unsigned int

The PROD pointer as seen by the programmer (including wrap).

q_unfetched_cons unsigned int

The CONS pointer of the region unfetched (including wrap).

q_unfetched_prod unsigned int

The PROD pointer of the region unfetched (including wrap).

s_cmdq_state

A description of what the CMDQ can do now. Fields:

can_cerror bool

There is a CERROR that can be recognised.

can_cmd_sync bool

There is a CMD_SYNC that can be recognised.

can_fetch bool

The SMMU can fetch.

can_herror bool

There is an HERROR that can be recognised.

can_issue bool

There are fetched commands that are eligible to be issued.

can_update_cons bool

The programmer view of CMDQ_CONS can be moved to say some have been consumed.

cmd_queue unsigned int

The top bits indicate the CMDQ id. These bits are attached to all cmd_ids generated for this queue. The bits [63:44] contain the queue ID. [63:62] 0: s-ssd, 1: reserved, 2: ns-ssd. 3: rl-ssd [61] is ECMDQ [59:52] QCP index [51:44] ECMDQ index in QCP.

cmd_sync enum

Why a CMD_SYNC cannot be issued.

cmdqen bool

CMDQEN, when enabled then the CMDQ can fetch commands.

cmdqen_1_to_0_unacknowledged bool

CMDQEN has been set to 0, but we have yet to acknowledge it.

current-real-cerror enum

The current CERROR that would be visible to the programmer if there was an error.

fetch-disabled_due_to_programmer_error bool

The PROD/CONS pointer was inconsistent and the model decided to stop processing commands.

fetch-number_of_unfetched_commands unsigned int

The number of commands that have yet to be fetched from memory, or ~0u if CMDQEN == 0.

fetch-state enum

The state of the fetch state machine.

fetch-too_many_commands_outstanding_so_not_fetching_any_more bool

If there are too many issued commands, or pending to be issued then we won't fetch any more.

issue-number_of_issuable_commands unsigned int

The number of commands that have been fetched but not yet issued.

issue-number_of_unissuable_commands unsigned int

This might include commands behind an illegal command, or behind a CMD_SYNC. Under some circumstances this can include a command representing an external abort.

issue-state enum

The internal issued state machine state.

outstanding_fetches_since_reset unsigned int

The number of outstanding fetches.

synthetic_sync_can_be_completed bool

The synthetic sync that is outstanding can be completed.

waiting_for_synthetic_sync bool

There is a synthetic sync that is outstanding.

s_eventq_adding_to_cmd_sync_prior_set

The event created by the specified transaction id must become visible to the programmer before the CMD_SYNC can complete. Fields:

trans_id unsigned int

The transaction id that caused the event.

s_eventq_auto_retry_stalled_transaction

A transaction that was stalled but not yet reported on the Event Queue was retried. The retry can happen if the transaction didn't write because it was de-duplicated behind a reported stalling transaction and that stalling transaction was CMD_RESUMEd, or if the queue was unwritable and the queue is now writable. Fields:

trans_id unsigned int

Transaction id of the auto-retried transaction.

why enum

Why the transaction retried.

s_eventq_cmd_sync_unhazardred

The CMD_SYNC has been unhazardred as the prior set is empty and as much of it as required is visible in the programmers' view. Fields:

prod_incl_wrap unsigned int

The prod index including wrap bit that was required to become visible.

prog_prod_incl_wrap unsigned int

The current programmer-visible prod including wrap bit.

s_eventq_eventqen_and_eventqenack

Trace the values of EVENTQEN and EVENTQENACK as they change. Fields:

about enum

Some more information about the queue state.

cons_incl_wrap unsigned int

The CONS pointer (including wrap bits).

eventqen bool

EVENTQEN.

eventqenack bool

EVENTQENACK.

prod_incl_wrap unsigned int

The PROD pointer (including wrap bits).

s_eventq_external_abort

A particular eventq record got an external abort. The event will subsequently appear in the ns/s_eventq_lost_event_records trace. Fields:

prod unsigned int

The prod pointer (excluding wrap bit) of the event record that aborted.

trans_id unsigned int

The transaction id that produced the event record that aborted.

s_eventq_lost_event_records

Event records that are lost and never record appear on this trace. Fields:

dword0 unsigned int

Double word 0 of event.

dword1 unsigned int

Double word 1 of event.

dword2 unsigned int

Double word 2 of event.

dword3 unsigned int

Double word 3 of event.

event enum

Event code that was thrown away.

prod unsigned int

The PROD index of the event if appropriate or ~0u otherwise.

reason enum

The reason the record was lost.

trans_id unsigned int

The transaction id that this corresponds to.

s_eventq_overflow_acknowledged

When events are lost then they appear on the eventq_lost_event_records_* trace.

This will activate the overflow flag in the corresponding SMMU_s_EVENTQ_PROD register. Software can then acknowledge that flag using the overflow acknowledge flag in SMMU_s_EVENTQ_CONS. Fields:

bad bool

If true then the programmer has attempted to acknowledge an overflow condition that didn't exist.

count_of_events_lost_due_to_overflow unsigned int

This is the number of events that were lost because the event queue was full. This does not include any lost because of an EVENTQ_ABT_ERR.

s_eventq_pending_event_records

Event records that are pending to be dealt with by the eventq state machine. A specific event can be traced multiple times if it becomes part of the prior set of transactions of a CMD_SYNC that has to complete before the CMD_SYNC is allowed to be completed. Fields:

dword0 unsigned int

Double word 0 of event.

dword1 unsigned int

Double word 1 of event.

dword2 unsigned int

Double word 2 of event.

dword3 unsigned int

Double word 3 of event.

event enum

Event code.

is_prior_set bool

The transaction forms part of the prior set of a CMD_SYNC that have to become visible (or thrown away if necessary) before the CMD_SYNC can complete.

trans_id unsigned int

The transaction id that this corresponds to.

s_eventq_prod_cons_updated

This records when the programmer visible PROD/CONS pointer is updated. More can be visible in memory than are advertised in the programmer-visible PROD pointer. This also traces the latest PROD that could have been advertised. Fields:

mem_prod unsigned int

The PROD pointer value for all records that are visible in the memory.

num_events_available unsigned int

The number of events available to the programmer.

prog_cons unsigned int

The programmer-visible CONS pointer.

prog_prod unsigned int

The programmer-visible PROD pointer.

what enum

What changed?.

s_eventq_raising_eventq_abt_err

EVENTQ_ABT_ERR can now be raised.

s_eventq_removing_from_cmd_sync_prior_set

The event captured by a CMD_SYNC and created by the specified transaction id is now in memory. Fields:

prod unsigned int

The prod index (no wrap bit) that must become visible to the programmer to see this record. If not head_of_line then the record will not become visible and this field is meaningless.

reason enum

The reason the event was removed.

trans_id unsigned int

The transaction id that caused the event.

s_eventq_stashing_unreported_stalled_transaction_for_auto_retry

A stall event record would have been lost, instead it is stashed away to auto-retry when it can. Fields:

dword0 unsigned int

Double word 0 of event.

dword1 unsigned int

Double word 1 of event.

dword2 unsigned int

Double word 2 of event.

dword3 unsigned int

Double word 3 of event.

event enum

Event code that was thrown away.

reason enum

The reason the record was lost.

trans_id unsigned int

The transaction id that this corresponds to.

s_eventq_write_event

An event is committed to being written. Fields:

dword0 unsigned int

Double word 0 of event.

dword1 unsigned int

Double word 1 of event.

dword2 unsigned int

Double word 2 of event.

dword3 unsigned int

Double word 3 of event.

event enum

Event code that was thrown away.

prod unsigned int

The PROD pointer (without wrap) that this event will be written to.

stall_event bool

Describes an event for a stalled transaction.

trans_id unsigned int

Transaction ID that generated this event.

s_eventq_write_event_visible_in_memory

An event is now visible in memory. Fields:

abort bool

The write aborted.

dword0 unsigned int

Double word 0 of event.

dword1 unsigned int

Double word 1 of event.

dword2 unsigned int

Double word 2 of event.

dword3 unsigned int

Double word 3 of event.

event enum

Event code that was thrown away.

prod unsigned int

The PROD pointer (without wrap) that this event will be written to.

stall_event bool

Describes an event for a stalled transaction.

trans_id unsigned int

Transaction ID that generated this event.

s_gerror_inverted

A GERROR bit was attempted to be inverted to record a fault. Fields:

already_different_to_gerrorn bool

True if the bit was already different to the corresponding bit in GERRORN and so it was not inverted.

gerrorn unsigned int

GERRORN register value.

interrupt_potentially_sent bool

An interrupt is potentially sent, depending on IRQEN.
GERROR(N).MSI_GERROR_ABT_ERR does not send an interrupt.

new_gerror unsigned int

GERROR register value after any inversion.

which enum

Which bit was attempted to be inverted.

s_gerrorn_acknowledge

A GERROR was acknowledged by SW. Fields:

which enum

Which GERROR was acknowledged.

s_priq_auto_response

A PRIQ auto response is generated. Fields:

prgindex unsigned int

The PRG Index of the response.

resp enum

The response to send.

streamid unsigned int

The StreamID of the response.

substreamid unsigned int

The PASID/SubstreamID attached to the response, or ~0u if there is none.

trans_id unsigned int

The ID of the PPR that is being lost.

why enum

Why the PPR is being lost.

s_priq_lost_ppr

The PRIQ is not reporting a PPR into the queue. It may or may not generate an auto-response. Fields:

is_stop_marker bool

The PPR being lost is a Stop Marker.

resp enum

What, if any, auto-response will be.

trans_id unsigned int

The ID of the PPR that is being lost.

why enum

Why the PPR is being lost.

s_priq_received

A PRIQ Request has been received and is queued waiting processing. This does not mean that it has been written to the PRIQ but has been placed in a pending queue awaiting a decision about what to do about it. Fields:

L bool

The Last in the Page Request Group (PRG).

R bool

Read permission is requested.

W bool

Read permission is requested.

prgindex unsigned int

The Page Request Group (PRG) index that is being used to identify this request to the EndPoint.

stop_marker bool

If LRW == 0b100 then this indicates this is a stop marker.

streamid unsigned int

The StreamID this request is associated with.

substreamid unsigned int

The SubstreamID (PASID) this request is associated with, of ~0u if no SubstreamID.

substreamid_P bool

If has a SubstreamID (!= ~0u) then Privileged Mode Requested.

substreamid_X bool

If has a SubstreamID (!= ~0u) then Execution Requested (R must also be 1).

trans_id unsigned int

The transaction ID that identifies this request.

untranslated_address unsigned int

The Untranslated Address that this is requesting.

sec_override

Controls non-secure accesses to some registers. Fields:

value bool

The value of the signal.

sev

Send a SEV. Fields:

why enum

Why the SEV was generated.

smmu_atc_inv

The CMD_ATC_INV command is sent. Fields:

address unsigned int

Untranslated Address to invalidate.

cmd_id unsigned int

Command id.

cons unsigned int

CONS of the command.

global bool

Global flag.

log2_size_in_bytes unsigned int

Log 2 of the size in bytes of the region to invalidate.

ns_number_in_flight unsigned int

Number of ATC invalidates in flight issued from the Non-secure CMDQ.

ns_number_waiting_to_go_out unsigned int

The total number of ATC invalidates waiting to be pushed out from the Non-secure CMDQ.

rl_number_in_flight unsigned int

Number of ATC invalidates in flight issued from the Realm CMDQ.

rl_number_waiting_to_go_out unsigned int

The total number of ATC invalidates waiting to be pushed out from the Realm CMDQ.

s_number_in_flight unsigned int

Number of ATC invalidates in flight issued from the Secure CMDQ.

s_number_waiting_to_go_out unsigned int

The total number of ATC invalidates waiting to be pushed out from the Secure CMDQ.

ssd_of_queue enum

The SSD of the CMDQ.

ssd_of_streamid enum

The SSD of the StreamID.

streamid unsigned int

StreamID in the invalidate command.

substreamid unsigned int

SubstreamID or ~0u if SSValid == false.

smmu_atc_inv_completed

The CMDQ_ATC_INV command completed. Fields:

cmd_id unsigned int

Command id, if top-bit is set then was issued from the Non-secure CMDQ.

ns_number_in_flight unsigned int

Number of ATC invalidates in flight now this one has completed, that were issued by the Non-secure CMDQ.

ns_number_waiting_to_go_out unsigned int

The total number of ATC invalidates waiting to be pushed out from the Non-secure CMDQ.

response enum

The response.

rl_number_in_flight unsigned int

Number of ATC invalidates in flight now this one has completed, that were issued by the Realm CMDQ.

rl_number_waiting_to_go_out unsigned int

The total number of ATC invalidates waiting to be pushed out from the Realm CMDQ.

s_number_in_flight unsigned int

Number of ATC invalidates in flight now this one has completed, that were issued by the Secure CMDQ.

s_number_waiting_to_go_out unsigned int

The total number of ATC invalidates waiting to be pushed out from the Secure CMDQ.

ssd_of_cmdq enum

The SSD of the CMDQ.

ssd_of_streamid enum

The SSD of the StreamID.

smmu_atc_inv_end

An ATC invalidate request has been started. Fields:

PASID_global bool

If has a SubstreamID (PASID) then if it is 'global' or not.

address unsigned int

The Untranslated Address to use.

log2_size_in_bytes unsigned int

Log2 of the size of the region in bytes to invalidate.

response enum

The response to the ATC invalidate.

ssd enum

The SSD of the stream.

streamid unsigned int

The StreamID to use.

substreamid unsigned int

The SubstreamID to use (or ~0u if not used). This corresponds to the PASID.

smmu_atc_inv_start

An ATC invalidate request has been started. Fields:

PASID_global bool

If has a SubstreamID (PASID) then if it is 'global' or not.

address unsigned int

The Untranslated Address to use.

log2_size_in_bytes unsigned int

Log2 of the size of the region in bytes to invalidate.

ssd enum

The SSD of the stream.

streamid unsigned int

The StreamID to use.

substreamid unsigned int

The SubstreamID to use (or ~0u if not used). This corresponds to the PASID.

smmu_ats_initial

This is the initial ATS request. Fields:

XT bool

The XT bit.

ia unsigned int

Input address.

max_number_of_replies unsigned int

The maximum number of replies allowed to return.

no_write bool

The NW (no write flag) of the ATS request. If clear then the requester is going to do a write.

pasid_execute_requested bool

If there is a PASID (substreamid != ~0u) then this represents the requester asking for execution permissions.

pasid_privileged_mode_requested bool

If there is a PASID (substreamid != ~0u) then this represents the requester asking for the privileged mode's permissions.

protected_mode bool

Is the ATS Request protected-mode?.

source_cxl bool

Does the ATS request have Source.CXL set?.

ssd enum

The SSD of the ATS request.

ssd_ns bool

Incoming SSD is non-secure.

streamid unsigned int

StreamID of the ATS request.

substreamid unsigned int

SubstreamID (which is identical to the PASID) of the ATS request. If no PASID-prefix is sent then this is ~0u.

tbu unsigned int

Translation Buffer Unit number.

smmu_ats_reply_failure

This is an ATS reply indicating failure. Fields:

event enum

Equivalent event number that would have been generated for an equivalent ordinary transaction.

failure enum

What is the failure response code?.

state enum

The transaction state of the successfully ATS request.

smmu_ats_reply_success

This is an ATS reply, typically the SMMU will only return a single response, even if the requester indicated it could accept more replies. NOTE that the SMMU responds with 'success' in some cases when a fault is encountered and RW==0. Fields:

N bool

Non-snooped access. If one then the requester must clear the NoSnoop bit on transactions, unless otherwise enabled in a Function-specific manner.

P bool

Privileged mode. These permissions related to privileged mode.

RWX enum

Read/Write/Execute.

U bool

Untranslated access. If one, and RW !=0 then use UntranslatedAccesses for the allowed accesses by RW(X).

cxl_io bool

The CXL.io response.

inner enum

The inner cacheability attributes to use for TranslatedAccesses.

input_address unsigned int

Input address of the ATS request.

instcfg enum

The STE.INSTCFG field.

outer enum

The outer cacheability attributes to use for TranslatedAccesses.

pas enum

The PAS this mapping corresponds to. This holds the same information as the TE bit for realm streams.

privcfg enum

The STE.PRIVCFG field.

shareability unsigned int

The shareability to use for TranslatedAccesses.

size unsigned int

The size of the region covered by this translation.

state enum

The transaction state of the successfully ATS request.

translated_address unsigned int

If $RW \neq 0$ && $U \neq 0$, then the Translated Address that a TranslatedAccess can be made with.

smmu_axi_stream_msi

An SMMU generated MSI is directly sent through the axi_stream_msi_m port, typically connected to the GIC port axi_stream_msi_s. Fields:

TDEST unsigned int

Routing information for the data stream, typically identifying the GIC.

TID unsigned int

Data stream identifier for the SMMU.

axi_stream_msi_addr_to_match unsigned int

Current address to match for SMMU-originated MSIs to send out of the axi_stream_msi_m port.

data unsigned int

The MSI sent.

smmu_final_transaction

This is the transaction group request to remap has completed one set of remapping. For ordinary transactions, this represents a bundle of transactions with the same attributes but different addresses within a certain range around the address. If it stalls then it will report through this trace source, stall (stag_if_stalling != ~0u) and when resume will issue another smmu_initial_transaction as it undergoes remapping again. This trace source can also represent part of the process for ATOS/ATS or finding STE.PPAR for PRI requests that need to be auto-responded to. Fields:

axmmuflow enum

The AxMMUFLOW for this transaction group. stallable – allow the transaction to be stalled if configured. translated-access – is a PCIe Translated Access non-stallable – do not stall transaction even if configured. transfault-flow – do not report for Translation Related Faults, but tell client device to use a page request mechanism instead (for example PRI).

begin_input_address_range unsigned int

The start of the input address range that is size_of_region_in_bytes.

begin_ipa_range unsigned int

The start of the IPA range that is of size_of_region_in_bytes.

begin_output_address_range unsigned int

The start of the output address range that is of size_of_region_in_bytes.

cmo_point enum

The point associated with the CMO, if applicable.

input_address unsigned int

The input address of the transaction group.

input_inner enum

Inner cacheability for the input attributes.

input_outer enum

Outer cacheability for the input attributes.

input_prot enum

The instruction/data, bus-ns/s and privileged/user marking.

input_sh enum

Shareability for the input attributes.

ipa_address unsigned int

The IPA of the transaction.

kind enum

The kind of access this transaction group that this represents.

mecid unsigned int

The MECID of the transaction.

mpam_pmg_and_partid unsigned int

Bits[23:16] are the MPAM PMG. Bits[15:0] are the physical PARTID.

mpam_sp enum

The MPAM_SP of the transaction.

operation enum

The kind of operation that this represents.

output_address unsigned int

The input address of the transaction group.

output_inner enum

Inner cacheability for the output attributes.

output_outer enum

Outer cacheability for the output attributes.

output_prot enum

The instruction/data, bus-ns/s and privileged/user marking.

output_sh enum

Shareability for the output attributes.

output_vmid unsigned int

The output VMID/GBPA.IMPDEF or ~0u if not valid.

size_of_region_in_bytes unsigned int

An imp def size of region for which this translation is valid for.

ssd enum

The SSD of the transaction.

ssd_ns enum

The security state of the transaction.

stag_if_stalling unsigned int

This is the STAG used by the transaction if it is going to stall. It is ~0u if it is not going to stall.

state enum

The final transaction state.

streamid unsigned int

The StreamID of the transaction. ~0ull if NoStreamID.

substreamid unsigned int

The SubstreamID of the transaction or 0xFFFFffff if no SubstreamID.

tbu unsigned int

Translation Buffer Unit number.

trans_id unsigned int

Transaction id. Top bit set if debug transaction.

smmu_gpt_tlb_invalidate

A GPT TLB invalidate has been initiated. Fields:

address unsigned int

The address as it appears in the operation.

encoded_size unsigned int

The size as it is encoded in the operation.

kind enum

The kind of operation this is.

pgs_in_bytes unsigned int

The PGS size in bytes.

size_in_bytes unsigned int

For range operations, the size as it appears in the operation.

source enum

Where the TLBI came from.

state enum

Is the operation well formed.

trans_id unsigned int

The transaction id of this invalidate.

smmu_gpt_tlb_invalidate_complete

The GPT TLB invalidate completed. Fields:

source enum

Where the TLBI came from.

trans_id unsigned int

The transaction id of this invalidate.

smmu_initial_transaction

This is the transaction group request to remap is going to start one set of remapping. For ordinary transactions, this represents a bundle of transactions with the same attributes but different addresses within a certain range around the address. A stalling transaction will report through this trace source when it unstalls. This trace source can also represent part of the process for ATOS/ATS or finding STE.PPAR for PRI requests that need to be auto-responded to. Fields:

XT_and_output_pas_checking enum

The XT bit for PCIe Transactions. This specifies the requested check on the output PAS that the device asked for.

axmmuflow enum

The AxMMUFLOW for this transaction group. stallable – allow the transaction to be stalled if configured. translated-access – is a PCIe Translated Access non-stallable – do not stall transaction even if configured. transfault-flow – do not report for Translation Related Faults, but tell client device to use a page request mechanism instead (for example PRI).

cmo_point enum

The point associated with the CMO, if applicable.

dcmdq_qcp_and_index unsigned int

If this is a DCMDQ fetch being translated then this field indicates the DCMDQ QCP index in bits [23:8] and the index in the page in bits [7:0]. Otherwise this is 0xFFFFFFFF.

input_address unsigned int

The input address of the transaction group.

input_inner enum

Inner cacheability for the input attributes.

input_outer enum

Outer cacheability for the input attributes.

input_prot enum

The instruction/data, bus-ns/s and privileged/user marking.

input_sh enum

Shareability for the input attributes.

kind enum

The kind of access this transaction group that this represents.

mecid unsigned int

The incoming MECID for NoStreamID transactions. ~0u for all other transactions.

mpam_partid unsigned int

The MPAM_PARTID for NoStreamID transactions. ~0u for all other transactions.

mpam_pmg unsigned int

The MPAM_PMG for NoStreamID transactions. ~0u for all other transactions.

mpam_sp enum

The MPAM_SP for NoStreamID transactions. ~0u for all other transactions.

operation enum

The kind of operation that this represents.

protected_mode enum

The PM bit.

ssd enum

The SSD of the transaction.

ssd_ns enum

The security state of the transaction.

streamid unsigned int

The StreamID of the transaction. ~0ull means NoStreamID.

substreamid unsigned int

The SubstreamID of the transaction or 0xFFFFffff if no SubstreamID.

tbu unsigned int

Translation Buffer Unit number.

trans_id unsigned int

Transaction id. Top bit set if debug transaction.

smmu_normalize_input_transaction

If the input transaction is normalized before being processed then this traceevent will fire. This is usually dependent on parameters of the implementation. Fields:

incoming_inner enum

The inner cacheability attributes.

incoming_is_instruction bool

The incoming transaction is marked as 'instruction'.

incoming_is_privileged bool

The incoming transaction is marked as 'privileged'.

incoming_outer enum

The outer cacheability attributes.

incoming_pas enum

The PAS of the incoming transaction.

incoming_shareability unsigned int

The incoming shareability.

normalized_inner enum

The normalized inner cacheability attributes.

normalized_is_instruction bool

The incoming transaction is marked as 'instruction'.

normalized_is_privileged bool

The incoming transaction is marked as 'privileged'.

normalized_outer enum

The normalized outer cacheability attributes.

normalized_pas enum

The PAS this mapping corresponds to.

normalized_shareability unsigned int

The normalized shareability.

ssd enum

The SSD of the transaction.

streamid unsigned int

The StreamID of the transaction. ~0ull means NoStreamID.

substreamid unsigned int

The SubstreamID of the transaction or 0xFFFFffff if no SubstreamID.

trans_id unsigned int

ID of the original transaction.

smmu_pmusnapshot_ack

Acknowledge the pmusnapshot_req to indicate the snapshot has occurred. This is part of a four-phase handshake. Fields:

pin_index unsigned int

The pin index of the signal, or ~0u if not relevant.

pmcg_index enum

The PMCG index.

value bool

The value of the signal.

smmu_pmusnapshot_ack_overridden

The value of pmusnapshot_ack was overridden (likely due to being in integration mode). Fields:

pin_index unsigned int

The pin index of the signal, or ~0u if not relevant.

pmcg_index enum

The PMCG index.

value bool

The value of the signal.

smmu_pmusnapshot_ack_override_end

The overriding of the pmusnapshot_ack signal has ended. Fields:

pin_index unsigned int

The pin index of the signal, or ~0u if not relevant.

pmcg_index enum

The PMCG index.

smmu_pmusnapshot_req

Take a snap shot of the PMU values as though SMMU_PMCGR_CAPTURE had been written. This is part of a four-phase handshake. Fields:

pin_index unsigned int

The pin index of the signal, or ~0u if not relevant.

pmcg_index enum

The PMCG index.

value bool

The value of the signal.

smmu_poison_tw_data

Poison data has been returned to a table walk transaction. Fields:

bitmap_of_poison unsigned int

The bitmap of which beats of the transaction were poisoned.

is_cas bool

True if this is a compare-and-swap operation.

number_of_64bit_beats unsigned int

Number of beats this transaction fetched.

paddress unsigned int

Physical address of the table walk transaction.

pas enum

The PAS of the bus transaction.

ras_group_id unsigned int

If non-~0u then is the RAS group that we wish to attach the error to. Otherwise will use the IMP DEF policy.

ras_record_index_in_group unsigned int

If non-~0u then is the RAS record index in the group that we wish to attach the error to. Otherwise will use the IMP DEF policy.

what enum

What table walk was being performed.

smmu_pri_resp

The CMD_PRI_RESP command is queued to be sent to the PCIe system. Fields:

auto_response_trans_id unsigned int

trans_id of PRI request we are auto-responding to, or ~0ull if not valid.

cmd_id unsigned int

Command id, or ~0ull if not valid.

cons unsigned int

CONS of the command. ~0u if an auto-response.

prgindex unsigned int

PRG Index.

response enum

If has a SubstreamID (PASID) then if it is 'global' or not.

ssd enum

The SSD of the StreamID this corresponds to.

streamid unsigned int

StreamID in the invalidate command.

substreamid unsigned int

SubstreamID or ~0u if SSValid == false.

smmu_pri_resp_nop

The CMD_PRI_RESP command was NOPped. Fields:

cmd_id unsigned int

Command id.

cons unsigned int

CONS of the command.

prgindex unsigned int

PRG Index.

response enum

If has a SubstreamID (PASID) then if it is 'global' or not.

ssd enum

The SSD of the StreamID this corresponds to.

streamid unsigned int

StreamID in the invalidate command.

substreamid unsigned int

SubstreamID or ~0u if SSValid == false.

why enum

Why the CMD_PRI_RESP was NOPped.

smmu_priq_resp_fake_return

A PRIQ Response is posted to the PCIe subsystem and so has no acknowledgement that it is received. However, in the model then we artificially know when the the PRIQ Response has been delivered to the PCIe subsystem, even if the ATC has not yet acted on it. Fields:

prgindex unsigned int

The PRG Index this request corresponds to.

ssd enum

SSD of response.

streamid unsigned int

The StreamID to use.

substreamid unsigned int

The SubstreamID to use (or ~0u if not used). This corresponds to the PASID.

what enum

What happened.

smmu_priq_resp_start

A PRIQ Response has been posted to the PCIe subsystem. As the response is posted then there is no way of knowing when it is received by the EndPoint. Fields:

prgindex unsigned int

The PRG Index this request corresponds to.

response enum

If has a SubstreamID (PASID) then if it is 'global' or not.

ssd enum

SSD of response.

streamid unsigned int

The StreamID to use.

substreamid unsigned int

The SubstreamID to use (or ~0u if not used). This corresponds to the PASID.

smmu_thread_wait_wake

Traces a thread's wait/wake status. Fields:

current_ticks unsigned int

The current tick count of simulated time.

event enum

What is happening to this thread.

thread_index unsigned int

The ID of this thread.

ticks unsigned int

If the event relates to a time then this is held in this field. Otherwise, 0.

stall_transaction

A transaction is about to stall. Fields:

stag unsigned int

STAG.

streamid unsigned int

StreamID.

streamid_ns bool

The StreamID was for the non-secure world.

trans_id unsigned int

Transaction id.

stall_transaction_inhibited_by_STALL_MAX

A transaction is about to stall but the maximum number of transactions have stalled and we can't report this one to the event queue (even if non-full). Fields:

streamid unsigned int

StreamID.

streamid_ns bool

The StreamID was for the non-secure world.

trans_id unsigned int

Transaction id.

stall_transaction_resuming

A stalled transaction is resuming. Fields:

stag unsigned int

STAG if appropriate, or if was inhibited by STALL_MAX then 0xFAFA.

stallresult enum

What the transaction resumed to do.

streamid unsigned int

StreamID.

streamid_ns bool

The StreamID was for the non-secure world.

trans_id unsigned int

Transaction id.

start_ptw_read

Page Table Walk (read). This is the start of the physical access that the SMMU is making. Fields:

adomain enum

The actual attributes of the access that was used, after IMP DEF mangling.

aprot enum

The actual attributes of the access that was used, after IMP DEF mangling. PAS[1] (bit[3]), Instruction/Data (bit[2]), PAS[0] (bit[1]), Privileged/User (bit[0]).

desckind enum

The kind of descriptor.

inner_cache enum

The actual attributes of the access that was used, after IMP DEF mangling.

input_address unsigned int

Input address this lookup is for (after TBI normalisation).

mecid unsigned int

The MECID of the TTD fetch.

mpam_pmg_and_partid unsigned int

Bits[23:16] are the MPAM PMG. Bits[15:0] are the physical PARTID.

ns enum

Non-Secure on the bus.

outer_cache enum

The actual attributes of the access that was used, after IMP DEF mangling.

pa_address unsigned int

Physical address.

pas enum

The PAS of the TTD fetch.

ssd enum

SSD of the transaction.

ssd_ns enum

Security state of stream.

streamid unsigned int

The StreamID this is for.

substreamid unsigned int

The SubstreamID this is for.

trans_id unsigned int

Transaction id.

ttb_grain_stage_and_level unsigned int

Which TTB used in bits[31:24]. Which Grain size used in bit[23:16]:- 0 – 4 KiB 1 – 16 KiB 2 – 64 KiB 3 – 64 KiB ARMv8.2 where descriptor bits [15:12] for bits [51:48] of output address 4 – 4 KiB FEAT_LPA2 where address is {bits[9:8], bits[49:12]} 5 – 16 KiB FEAT_LPA2 where address is {bits[9:8], bits[49:14], 2'b00} 6 – 64 KiB ARMv8.2 where descriptor bits [15:12] are ignored Stage in bits[15:8]. Level in bits[7:0], level -1 is 0xF.

ste_cc.STE_commentary

A commentary from the cache about what is going on. Fields:

output string

The stream output.

ste_cc.STE_entry_invalidated

An entry in the cache has been invalidated. Fields:

entry_id unsigned int

The entry id that is being invalidated.

reason enum

Why the entry is being invalidated.

ste_entry_allocated

An STE entry has been allocated. Fields:

entry_id unsigned int

The id of the cache entry. Top bit is set if is a debug entry.

ns enum

For the non-secure world.

ssd enum

The SSD of the entry.

ste string

A textual description of the STE.

streamid unsigned int

The StreamID of the entry this will match.

sup_btm

The pin is driven. This indicates the system supports BTM. This pin is sampled on the negedge of the reset pin and has to be driven beforehand for the value to take effect. Fields:

value bool

The value of the signal.

sup_cohacc

The pin is driven. This indicates the system supports COHACC. This pin is sampled on the negedge of the reset pin and has to be driven beforehand for the value to take effect. Fields:

value bool

The value of the signal.

sup_httu

The pin is driven. This indicates the system supports HTTU. This pin is sampled on the negedge of the reset pin and has to be driven beforehand for the value to take effect. Fields:

value bool

The value of the signal.

sup_sev

The pin is driven. This indicates the system supports SEV. This pin is sampled on the negedge of the reset pin and has to be driven beforehand for the value to take effect. Fields:

value bool

The value of the signal.

tbu0_reset_in

The reset signal of the TBU0. Fields:

value bool

The value of the signal.

tcu_reset_in

The reset signal. Fields:

value bool

The value of the signal.

tlb_entry_allocated

A TLB entry has been allocated. Fields:

aset unsigned int

The ASET of this entry.

asid unsigned int

ASID if appropriate.

index unsigned int

Index of TLB entry.

input_end_incl_address unsigned int

The end inclusive address of the input range that this matches.

input_start_address unsigned int

The start address of the input range that this matches.

output_end_incl_address unsigned int

The end inclusive address of the output range.

output_start_address unsigned int

The start address of the output range.

scheme enum

The tagging scheme used.

ssd enum

The SSD of the tagging scheme.

ssd_ns enum

Security State.

tbi bool

Was the entry formed using Top Byte Ignore (TBI).

vmid unsigned int

VMID if appropriate.

tlb_info_tlb_entries_overlap

A TLB entry was inserted into the TLB and it overlaps an existing entry. This isn't a problem as it was inserted in such a way that it architecturally works. Fields:

end_address_of_new_entry unsigned int

End address of new entry.

end_address_of_old_entry unsigned int

End address of old entry.

how_inserted enum

How the entry was inserted.

index_of_new_entry unsigned int

Index of new entry.

index_of_old_entry unsigned int

Index of old entry.

start_address_of_new_entry unsigned int

Start address of new entry.

start_address_of_old_entry unsigned int

Start address of old entry.

verbose_commentary

This is a verbose commentary on the translation process the SMMU is performing. Fields:

output string

The stream output.

warning_MSI_EVENTQ_address_out_of_range_of_oas

The MSI address is configured outside of the range of the downstream address bus size. Fields:

address unsigned int

The untruncated address of the MSI.

warning_MSI_GERROR_address_out_of_range_of_oas

The MSI address is configured outside of the range of the downstream address bus size. Fields:

address unsigned int

The untruncated address of the MSI.

warning_MSI_HACDBS_PROCESSING_COMPLETE_address_out_of_range_of_oas

The MSI address is configured outside of the range of the downstream address bus size.
Fields:

address unsigned int

The untruncated address of the MSI.

warning_MSI_HDBSS_TABLE_FULL_address_out_of_range_of_oas

The MSI address is configured outside of the range of the downstream address bus size.
Fields:

address unsigned int

The untruncated address of the MSI.

warning_MSI_PRIQ_address_out_of_range_of_oas

The MSI address is configured outside of the range of the downstream address bus size.
Fields:

address unsigned int

The untruncated address of the MSI.

warning_MSI_S_EVENTQ_address_out_of_range_of_oas

The MSI address is configured outside of the range of the downstream address bus size.
Fields:

address unsigned int

The untruncated address of the MSI.

warning_MSI_S_GERROR_address_out_of_range_of_oas

The MSI address is configured outside of the range of the downstream address bus size.
Fields:

address unsigned int

The untruncated address of the MSI.

warning_MSI_S_HACDBS_PROCESSING_COMPLETE_address_out_of_range_of_oas

The MSI address is configured outside of the range of the downstream address bus size.
Fields:

address unsigned int

The untruncated address of the MSI.

warning_MSI_S_HDBSS_TABLE_FULL_address_out_of_range_of_oas

The MSI address is configured outside of the range of the downstream address bus size.
Fields:

address unsigned int

The untruncated address of the MSI.

warning_MSI_pmcg_address_out_of_range_of_oas

The MSI Address of the Performance Monitor Counter Group (PMCG) is out of range of the OAS and so will be silently truncated. Fields:

address unsigned int

The untruncated address of the MSI.

pmcg_index unsigned int

Index of the PMCG.

ssd_ns bool

True if this PMCG controlled by the non-secure world.

warning_discarding_interrupt_EVENTQ_as_irqen_low

Interrupt generation is turned off by SMMU_IRQ_CTRL.EVENTQ_IRQEN.

warning_discarding_interrupt_GERROR_as_irqen_low

Interrupt generation is turned off by SMMU_IRQ_CTRL.GERROR_IRQEN.

warning_discarding_interrupt_HACDBS_PROCESSING_COMPLETE_as_irqen_low

Interrupt generation is turned off by SMMU_IRQ_CTRL.HACDBS_IRQEN.

warning_discarding_interrupt_HDBSS_TABLE_FULL_as_irqen_low

Interrupt generation is turned off by SMMU_IRQ_CTRL.HDBSS_IRQEN.

warning_discarding_interrupt_PRIQ_as_irqen_low

Interrupt generation is turned off by SMMU_IRQ_CTRL.PRIQ_IRQEN.

warning_discarding_interrupt_S_EVENTQ_as_irqen_low

Interrupt generation is turned off by SMMU_S_IRQ_CTRL.EVENTQ_IRQEN.

warning_discarding_interrupt_S_GERROR_as_irqen_low

Interrupt generation is turned off by SMMU_S_IRQ_CTRL.GERROR_IRQEN.

warning_discarding_interrupt_S_HACDBS_PROCESSING_COMPLETE_as_irqen_low

Interrupt generation is turned off by SMMU_IRQ_S_CTRL.HACDBS_IRQEN.

warning_discarding_interrupt_S_HDBSS_TABLE_FULL_as_irqen_low

Interrupt generation is turned off by SMMU_IRQ_S_CTRL.HDBSS_IRQEN.

warning_ns_gerrorn_bad_acknowledge

A GERROR was acknowledged by SW in the GERRORN register that did not have an active error. The result is **UNPREDICTABLE** if this will generate an interrupt or not. Fields:

which enum

Which GERROR was acknowledge when there was no active error.

warning_reg_after_doesnt_match_written_value

A write occurred that tried to set bits in a register, that for one reason or another, failed to get written. Fields:

desc string

The textual description of what happened.

warning_s_gerrorn_bad_acknowledge

A GERROR was acknowledged by SW in the GERRORN register that did not have an active error. The result is **UNPREDICTABLE** if this will generate an interrupt or not. Fields:

which enum

Which GERROR was acknowledge when there was no active error.

what_going_to_do_with_terminated_event

A terminating transaction has produced an event, this tells you what the model is going to do with the event. Fields:

CD.S bool

The CD.S field if available.

S2 bool

The event is related to Stage 2.

STE.S1STALLD bool

The STE.S1STALLD field if available.

STE.S2S bool

The STE.S2S field if available.

aborts bool

The transaction will abort.

axmmuflow enum

The AxMMUFLOW for this transaction group. stallable – allow the transaction to be stalled if configured. translated-access – is a PCIe Translated Access non-stallable – do not stall transaction even if configured. transfault-flow – do not report for Translation Related Faults, but tell client device to use a page request mechanism instead (for example PRI).

is_tr_fault bool

Is a Translation Related fault.

protected_mode bool

The transaction is protected-mode. As such, it cannot stall and will obey the report configuration bits.

reports bool

The transaction will attempt to report.

ssd enum

The SSD of the transaction.

ssd_ns bool

The transaction is classified as SSD non-secure.

supports_stall_model bool

The implementation supports the stall model.

trans_id unsigned int

The transaction id.

why_abort_decision enum

The reason why the transaction aborted/did not abort.

why_report_decision enum

The reason why the transaction reported/did not report.

2.126 MMU_720AE

This section describes the trace sources.

ArchMsg.Error.error

These messages are about activity occurring on the SMMU that is considered an error. Messages will only come out here if parameter `all_error_messages_through_trace` is true. `DISPLAY %{\output}`. Fields:

output string

The stream output.

ArchMsg.Error.fetch_from_memory_type_not_supporting_httu

A descriptor fetch from an HTTU-enabled translation regime to an unsupported memory type was made. Whilst the fetch itself may succeed, if an update to the descriptor was attempted then it would fail. Fields:

address unsigned int

The address of the descriptor fetch.

desc_inner enum

Inner cacheability of descriptor.

desc_outer enum

Outer cacheability of descriptor.

desc_prot enum

The instruction/data, bus-ns/s and privileged/user marking.

desc_sh enum

Shareability of descriptor.

ssd enum

The SSD of the transaction.

ssd_ns bool

Is the Security State Determination of the transaction non-secure.

stage unsigned int

The stage at which we had a problem.

streamid unsigned int

The StreamID of the transaction.

substreamid unsigned int

The SubstreamID of the transaction (or ~0u if no substreamid).

trans_id unsigned int

Transaction id.

ArchMsg.Error.ns_cmdq

There is an error associated with the command queue. Fields:

cmd_id unsigned int

Command ID of the command in error.

cons unsigned int

CONS of the command.

what string

What is wrong.

ArchMsg.Error.priq_streamid_truncated

The SMMU received a PCIe PRI request with a StreamID that was larger than that which the SMMU has been configured for. The StreamID that appears in the PRIQ entry will be truncated. Fields:

actual_streamid unsigned int

The actual StreamID that this request has.

sidsize unsigned int

The bit width of the SMMU for StreamIDs, as indicated by SMMU_IDR1.SIDSIZE.

trans_id unsigned int

The transaction ID that identifies this request.

trunc_streamid unsigned int

The truncated StreamID that will appear in the PRIQ entry.

ArchMsg.Error.priq_substreamid_truncated

The SMMU received a PCIe PRI request with a PASID prefix (SubstreamID) that was larger than that which the SMMU has been configured for. The SubstreamID that appears in the PRIQ entry will be truncated. Fields:

actual_substreamid unsigned int

The actual SubstreamID that this request has.

ssidsize unsigned int

The bit width of the SMMU for SubstreamIDs, as indicated by SMMU_IDR1.SSIDSIZE.

trans_id unsigned int

The transaction ID that identifies this request.

trunc_substreamid unsigned int

The truncated SubstreamID that will appear in the PRIQ entry.

ArchMsg.Error.rl_cmdq

There is an error associated with the command queue. Fields:

cmd_id unsigned int

Command ID of the command in error.

cons unsigned int

CONS of the command.

what string

What is wrong.

ArchMsg.Error.rl_priq_streamid_truncated

The SMMU received a PCIe PRI request with a StreamID that was larger than that which the SMMU has been configured for. The StreamID that appears in the PRIQ entry will be truncated. Fields:

actual_streamid unsigned int

The actual StreamID that this request has.

sidsize unsigned int

The bit width of the SMMU for StreamIDs, as indicated by SMMU_IDR1.SIDSIZE.

trans_id unsigned int

The transaction ID that identifies this request.

trunc_streamid unsigned int

The truncated StreamID that will appear in the PRIQ entry.

ArchMsg.Error.rl_priq_substreamid_truncated

The SMMU received a PCIe PRI request with a PASID prefix (SubstreamID) that was larger than that which the SMMU has been configured for. The SubstreamID that appears in the PRIQ entry will be truncated. Fields:

actual_substreamid unsigned int

The actual SubstreamID that this request has.

ssidsize unsigned int

The bit width of the SMMU for SubstreamIDs, as indicated by SMMU_IDR1.SSIDSIZE.

trans_id unsigned int

The transaction ID that identifies this request.

trunc_substreamid unsigned int

The truncated SubstreamID that will appear in the PRIQ entry.

ArchMsg.Error.s_cmdq

There is an error associated with the command queue. Fields:

cmd_id unsigned int

Command ID of the command in error.

cons unsigned int

CONS of the command.

what string

What is wrong.

ArchMsg.Error.tlb_entries_overlap

A TLB entry was attempted to be inserted into the TLB and was determined that it overlaps an existing entry. This check is not perfect but will catch simple errors. Fields:

do_f_tlb_conflict bool

Chosen to perform an F_TLB_CONFLICT.

end_address_of_new_entry unsigned int

End address of new entry.

end_address_of_old_entry unsigned int

End address of old entry.

index_of_new_entry unsigned int

Index of new entry.

index_of_old_entry unsigned int

Index of old entry.

start_address_of_new_entry unsigned int

Start address of new entry.

start_address_of_old_entry unsigned int

Start address of old entry.

why enum

Why it is an error for these two entries to overlap.

ArchMsg.Error.tlb_entry_not_invalidated_due_to_ril

An entry in the cache was not invalidated even though in the right address range because of the RIL fields of the command the entry do not match. Fields:

cmd_num unsigned int

The NUM field of the RIL part of the command.

cmd_ril_tg enum

The RIL_TG field of the RIL part of the command.

cmd_ril_ttl unsigned int

The TTL field of the RIL part of the command, zero means any level. 0x80 means match level 0 (and is from a DVM message).

cmd_scale unsigned int

The SCALE field of the RIL part of the command.

entry_id unsigned int

The entry id that is being invalidated.

tlb_entry string

The TLB entry.

ArchMsg.Error.vatos_sel_vmid_out_of_range

The SMMU_(S_)VATOS_SEL.VMID field was programmed with a VMID that was too wide for this implementation (SMMU_IDR0.VMID16 == 0). DISPLAY %{{ssd_ns:(s-|ns-)}}VMID:{{vmid}} is out of range. Fields:

ssd_ns bool

The security state of the VATOS interface.

vmid unsigned int

The VMID programmed.

ArchMsg.Info.info

These are information messages about what is happening in the SMMU. DISPLAY %{{output}}. Fields:

output string

The stream output.

ArchMsg.Warning.CMD_RESUME_no_transaction_resumed

A CMD_RESUME was issued that matched no transaction. Fields:

stag unsigned int

STAG in the CMD_RESUME.

streamid unsigned int

StreamID in the CMD_RESUME.

streamid_ns bool

The StreamID was for the non-secure world.

ArchMsg.Warning.NoStreamID_bad_pas_or_mpam_sp

When RME is not supported then a NoStreamID transaction with PAS[1] == 1 or MPAM_SP[1] == 1 is treated as though PAS[1] == 0 and MPAM_SP[1] == 0. This is usually a system construction error and is not expected to occur. Fields:

address unsigned int

The Physical Address.

effective_mpam_sp enum

The effective MPAM_SP.

effective_pas enum

The effective PAS.

extendedid unsigned int

The ExtendedID of the transaction.

incoming_mpam_sp enum

The incoming MPAM_SP.

incoming_pas enum

The incoming PAS.

managerid64 unsigned int

The ManagerID64 of the transaction,.

userflags unsigned int

The UserFlags of the transaction.

ArchMsg.Warning.atc_inv_strange

Something was odd about the CMD_ATC_INV. DISPLAY CMD_ATC_INV strange as: %{why}. Fields:

cmd_id unsigned int

Command id.

ssd_of_cmdq enum

The SSD of the CMDQ.

why string

Why the CMD_ATC_INV was strange.

ArchMsg.Warning.bad_axi_stream_msi_addr_to_match_s

The signal was not driven at reset by the external system, but was driven after we first started using the default value we had assumed. The external system, if it is going to drive the signal, must do so at first reset so that both sides of the connection can agree on the value of this signal. Fields:

default_value unsigned int

The default value of the signal we have been assuming.

value unsigned int

The value of the signal.

ArchMsg.Warning.bad_conf_system_supports_bgptm

The signal was not driven at reset by the external system, but was driven after we first started using the default value we had assumed. The external system, if it is going to drive the signal, must do so at first reset so that both sides of the connection can agree on the value of this signal. Fields:

default_value bool

The default value of the signal we have been assuming.

value bool

The value of the signal.

ArchMsg.Warning.bad_10gptsz

The signal was not driven at reset by the external system, but was driven after we first started using the default value we had assumed. The external system, if it is going to drive the signal, must do so at first reset so that both sides of the connection can agree on the value of this signal. Fields:

default_value unsigned int

The default value of the signal we have been assuming.

value unsigned int

The value of the signal.

ArchMsg.Warning.bad_legacy_tz_en

The signal was not driven at reset by the external system, but was driven after we first started using the default value we had assumed. The external system, if it is going to drive the signal, must do so at first reset so that both sides of the connection can agree on the value of this signal. Fields:

default_value bool

The default value of the signal we have been assuming.

value bool

The value of the signal.

ArchMsg.Warning.bad_s_gbpa_abort_init

The signal was not driven at reset by the external system, but was driven after we first started using the default value we had assumed. The external system, if it is going to drive the signal, must do so at first reset so that both sides of the connection can agree on the value of this signal. Fields:

default_value bool

The default value of the signal we have been assuming.

value bool

The value of the signal.

ArchMsg.Warning.bad_sec_override

The signal was not driven at reset by the external system, but was driven after we first started using the default value we had assumed. The external system, if it is going to drive the signal, must do so at first reset so that both sides of the connection can agree on the value of this signal. Fields:

default_value bool

The default value of the signal we have been assuming.

value bool

The value of the signal.

ArchMsg.Warning.bad_sup_btm

The signal was not driven at reset by the external system, but was driven after we first started using the default value we had assumed. The external system, if it is going to drive the signal, must do so at first reset so that both sides of the connection can agree on the value of this signal. Fields:

default_value bool

The default value of the signal we have been assuming.

value bool

The value of the signal.

ArchMsg.Warning.bad_sup_cohacc

The signal was not driven at reset by the external system, but was driven after we first started using the default value we had assumed. The external system, if it is going to drive the signal, must do so at first reset so that both sides of the connection can agree on the value of this signal. Fields:

default_value bool

The default value of the signal we have been assuming.

value bool

The value of the signal.

ArchMsg.Warning.bad_sup_httu

The signal was not driven at reset by the external system, but was driven after we first started using the default value we had assumed. The external system, if it is going to drive the signal, must do so at first reset so that both sides of the connection can agree on the value of this signal. Fields:

default_value bool

The default value of the signal we have been assuming.

value bool

The value of the signal.

ArchMsg.Warning.bad_sup_sev

The signal was not driven at reset by the external system, but was driven after we first started using the default value we had assumed. The external system, if it is going to drive the signal, must do so at first reset so that both sides of the connection can agree on the value of this signal. Fields:

default_value bool

The default value of the signal we have been assuming.

value bool

The value of the signal.

ArchMsg.Warning.bad_tbu0_reset_in

The signal was not driven at reset by the external system, but was driven after we first started using the default value we had assumed. The external system, if it is going to drive the signal, must do so at first reset so that both sides of the connection can agree on the value of this signal. Fields:

default_value bool

The default value of the signal we have been assuming.

value bool

The value of the signal.

ArchMsg.Warning.bad_tbu_fmurstdisable[0]

The signal was not driven at reset by the external system, but was driven after we first started using the default value we had assumed. The external system, if it is going to drive the signal, must do so at first reset so that both sides of the connection can agree on the value of this signal. Fields:

default_value bool

The default value of the signal we have been assuming.

value bool

The value of the signal.

ArchMsg.Warning.bad_tcu_fmurstdisable

The signal was not driven at reset by the external system, but was driven after we first started using the default value we had assumed. The external system, if it is going to drive the signal, must do so at first reset so that both sides of the connection can agree on the value of this signal. Fields:

default_value bool

The default value of the signal we have been assuming.

value bool

The value of the signal.

ArchMsg.Warning.bad_tcu_reset_in

The signal was not driven at reset by the external system, but was driven after we first started using the default value we had assumed. The external system, if it is going to drive the signal, must do so at first reset so that both sides of the connection can agree on the value of this signal. Fields:

default_value bool

The default value of the signal we have been assuming.

value bool

The value of the signal.

ArchMsg.Warning.contig_bit_gives_too_large_region_for_TxSZ

If the contig bit was used then the size of the contig region would be larger than that indicated by TxSZ. Fields:

desc_kind enum

Descriptor kind.

el enum

Exception level.

input_address unsigned int

Input address.

log2_contig_region_size unsigned int

Log2(contig region size in bytes).

output_address unsigned int

Output address.

stage_and_level unsigned int

The stage in bits [7:4] and the level in [3:0]. Level -1 is 0xF.

vmid unsigned int

VMID if appropriate.

ArchMsg.Warning.contig_bit_has_inconsistent_input_and_output_address

If the contig bit was used then the some bits of the output address held in the descriptor and the input address must match. Fields:

desc_kind enum

Descriptor kind.

el enum

Exception level.

input_address unsigned int

Input address.

log2_contig_region_size unsigned int

Log2(contig region size in bytes).

match_mask unsigned int

Bits that must match.

output_address unsigned int

Output address.

stage_and_level unsigned int

The stage in bits [7:4] and the level in [3:0]. Level -1 is 0xF.

vmid unsigned int

VMID if appropriate.

ArchMsg.Warning.fetch_from_memory_type_not_supporting_httu

A descriptor fetch from an HTTU-enabled translation regime to an unsupported memory type was made. Whilst the fetch itself may succeed, if an update to the descriptor was attempted then it would fail. Fields:

address unsigned int

The address of the descriptor fetch.

desc_inner enum

Inner cacheability of descriptor.

desc_outer enum

Outer cacheability of descriptor.

desc_prot enum

The instruction/data, bus-ns/s and privileged/user marking.

desc_sh enum

Shareability of descriptor.

ssd enum

The SSD of the transaction.

ssd_ns bool

Is the Security State Determination of the transaction non-secure.

stage unsigned int

The stage at which we had a problem.

streamid unsigned int

The StreamID of the transaction.

substreamid unsigned int

The SubstreamID of the transaction (or ~0u if no substreamid).

trans_id unsigned int

Transaction id.

ArchMsg.Warning.gpt_read_invalid_descriptor

A GPT read has completed and not found something that looks like a descriptor. Fields:

abort enum

Non-zero if the access aborted/failed.

address_checking unsigned int

The address that we are checking.

address_gptdesc unsigned int

The address of the GPT descriptor.

data unsigned int

The data fetch if it didn't abort.

level unsigned int

Level of the walk.

pas_checking enum

The PAS of the address that we are checking.

trans_id unsigned int

The trans_id of the transaction we are checking if appropriate.

what_checking enum

What kind of access are we checking.

ArchMsg.Warning.msi_address_truncated

An MSI was generated, but the address was silently truncated due to the limited downstream address bus width. Fields:

pmcg_index_and_counter unsigned int

If this is a PMCG interrupt, then the top 16 bits are the PMCG index, the lower 16 bits are counter index.

ras_error_group_id_node_id_and_record_index_in_node unsigned int

If this is a RAS interrupt then this is the error group record id in bits[23:16], the node_id in [15:8] and the record index in the node in [7:0].

truncated_address unsigned int

The truncated address.

untruncated_address unsigned int

The untruncated address.

which enum

Which MSI this is.

ArchMsg.Warning.msi_lost

An MSI was attempted to be sent, but couldn't be sent. Fields:

id unsigned int

ID of this interrupt transaction.

kind enum

What kind of interrupt.

why enum

Why this interrupt was denied.

ArchMsg.Warning.pmcg_non_secure_world

A commentary on potential mis-programming of the PMCG when transitioning the PMCG between security states. Fields:

commentary string

The commentary.

ArchMsg.Warning.pmcg_programming_violates_security

A commentary on potential mis-programming of the PMCG when transitioning the PMCG between different security states. This is under the assumption that an agent from another security state could be writing to the PMCG. Fields:

commentary string

The commentary.

ArchMsg.Warning.priq_auto_response_failed_to_find_STE

The PRIQ was going to generate an auto-response, but failed to find an STE and so is returning a Failure message to the EndPoint which should disable the PRI interface of the EndPoint. Fields:

trans_id unsigned int

The ID of the PPR that is being lost.

why enum

Why the PPR is being lost.

ArchMsg.Warning.priq_overflow_bad_acking

Indicates that an overflow condition was acknowledged by writing to:-
SMMU_PRIQ_CONS.OVACKFLG but an overflow condition did not exist. Fields:

new_ovackflg bool

The new value of the SMMU_PRIQ_CONS.OVACKFLG.

ArchMsg.Warning.priq_smmuen_forces_effective_priqen_low

If SMMUEN == 0, then the effective value of PRIQEN is 0. This warning is triggered when
PRIQEN == 1 && SMMUEN == 0; which may not be what was intended. The PRIQ cannot
be active if SMMUEN == 0.

ArchMsg.Warning.rl_priq_auto_response_failed_to_find_STE

The PRIQ was going to generate an auto-response, but failed to find an STE and so is
returning a Failure message to the EndPoint which should disable the PRI interface of the
EndPoint. Fields:

trans_id unsigned int

The ID of the PPR that is being lost.

why enum

Why the PPR is being lost.

ArchMsg.Warning.rl_priq_overflow_bad_acking

Indicates that an overflow condition was acknowledged by writing to:-
SMMU_PRIQ_CONS.OVACKFLG but an overflow condition did not exist. Fields:

new_ovackflg bool

The new value of the SMMU_PRIQ_CONS.OVACKFLG.

ArchMsg.Warning.rl_priq_smmuen_forces_effective_priqen_low

If SMMUEN == 0, then the effective value of PRIQEN is 0. This warning is triggered when
PRIQEN == 1 && SMMUEN == 0; which may not be what was intended. The PRIQ cannot
be active if SMMUEN == 0.

ArchMsg.Warning.sev_lost

A SEV was lost because it isn't supported according to SMMU_IDRO.SEV. DISPLAY SEV was
lost because: %{why}. Fields:

why enum

Why the SEV was generated.

ArchMsg.Warning.smmu_pcie_rc_is_in_reset_ignoring_atc_invalidate

A CMD_PRI_RESP or CMD_ATC_INV was sent with a streamid that mapped to a PCIe
Root Complex that is in reset. The message will be discarded and act as though completed
successfully. Fields:

port_index unsigned int

The port index (node index) that is in reset but we would have sent it to.

streamid unsigned int

The StreamID that failed to map.

ArchMsg.Warning.smmu_pcie_rc_not_found_for_streamid

A CMD_PRI_RESP or CMD_ATC_INV was sent with a streamid that did not map to a PCIe Root Complex. This might be that the SW used an incorrect StreamID or it might be that the model has not been connected correctly. ATC Invalidate messages complete as though successful and PRI Requests are ignored. Fields:

streamid unsigned int

The StreamID that failed to map.

ArchMsg.Warning.smmu_pmusnapshot

Something strange happened on the pmusnapshot_req/pmusnapshot_ack interface. Fields:

pin_index unsigned int

If the PMCG index corresponds to an array of signals, this is the index in the array, or 0 otherwise.

pmcg_index enum

The PMCG index.

warning string

The warning message.

ArchMsg.Warning.suspicious_overlapping_entries

Two DPT TLB entries are overlapping but they differ in ways that are potentially a SW error. Fields:

new_AC enum

The value of the 'AC' field that controls access to this region.

new_FWB bool

The region is FWB.

new_VMID unsigned int

The VMID, if any, associated with this region.

new_entry_index unsigned int

The entry index of the new DPT TLB entry.

new_entry_is_vmsa_formed bool

The new entry was inserted as a result of VMSA information.

new_input_end_incl_address unsigned int

The end inclusive address of the input range that this matches.

new_input_start_address unsigned int

The start address of the input range that this matches.

new_output_pas enum

The output PAS of this DPT region.

new_ssd enum

The SSD of the streams this region captures.

new_writeable bool

True if this region is writeable.

old_AC enum

The value of the 'AC' field that controls access to this region.

old_FWB bool

The region is FWB.

old_VMID unsigned int

The VMID, if any, associated with this region.

old_entry_index unsigned int

The entry index of the old DPT TLB entry.

old_entry_is_vmsa_formed bool

The old entry was inserted as a result of VMSA information.

old_input_end_incl_address unsigned int

The end inclusive address of the input range that this matches.

old_input_start_address unsigned int

The start address of the input range that this matches.

old_output_pas enum

The output PAS of this DPT region.

old_ssd enum

The SSD of the streams this region captures.

old_writeable bool

True if this region is writeable.

pas_differ bool

The output PAS of the two entries differ.

vmids_differ bool

The VMIDs are used by at least one of the AC schemes and are different.

vmsa_formed_writeable_while_DPT_entry_is_not bool

The VMSA-formed entry is writeable but the DPT Entry says it is not writeable.

ArchMsg.Warning.warning

These messages are about unusual (but not necessarily incorrect) activity occurring on the SMMU. DISPLAY %{output}. Fields:

output string

The stream output.

ArchMsg.Warning.warning_effective_EOPD_differs_from_what_would_be_cached_in_TLB

Effective value of EOPD differs from what would be cached in the TLB DISPLAY transaction (%{transaction_id}), sid (%{sid}), ssid (%{ssid}), ssd (%{ssd}), effective EOPD (%{effective_EOPD}), cached EOPD (%{cached_EOPD}). Fields:

cached_EOPD bool

The EOPD value that would be cached in the TLB.

effective_EOPD bool

The effective value of EOPD.

ssd enum

SSD.

streamid unsigned int

StreamID or ~0ull if NoStreamID.

substreamid unsigned int

SubstreamID or ~0u if SSValid == false.

transaction_id unsigned int

The transaction ID.

DPTTLB.tlb_commentary

A commentary from the cache about what is going on. Fields:

output string

The stream output.

DPTTLB.tlb_entry_invalidated

An entry in the cache has been invalidated. Fields:

entry_id unsigned int

The entry id that is being invalidated.

reason enum

Why the entry is being invalidated.

EVENTQ_config

Configuration of interrupt updated. Fields:

address unsigned int

Address of the MSI.

data unsigned int

Data payload of the MSI.

irqen bool

Appropriate bit of SMMU_s_IRQ_CTRL controlling this interrupt.

memattr enum

Memory type.

ns bool

PAS of bus attribute is non-secure.

pas enum

PAS of the MSI.

ra bool

read-allocate attribute.

sh enum

Shareability.

smmu_idr0_msi bool

Are MSIs supported for this security world.

tr bool

transient attribute.

wa bool

write-allocate attributes.

EVENTQ_irqen

The IRQEN has been changed. Fields:

new_value bool

The new value of IRQEN.

EVENTQ_irqen_ack

The IRQEN change has been acked. Fields:

new_value bool

The new value of IRQEN.

GERROR_config

Configuration of interrupt updated. Fields:

address unsigned int

Address of the MSI.

data unsigned int

Data payload of the MSI.

irqen bool

Appropriate bit of SMMU_s_IRQ_CTRL controlling this interrupt.

memattr enum

Memory type.

ns bool

PAS of bus attribute is non-secure.

pas enum

PAS of the MSI.

ra bool

read-allocate attribute.

sh enum

Shareability.

smmu_idr0_msi bool

Are MSIs supported for this security world.

tr bool

transient attribute.

wa bool

write-allocate attributes.

GERROR_irqen

The IRQEN has been changed. Fields:

new_value bool

The new value of IRQEN.

GERROR_irqen_ack

The IRQEN change has been acked. Fields:

new_value bool

The new value of IRQEN.

GPTTLB.tlb_commentary

A commentary from the cache about what is going on. Fields:

output string

The stream output.

GPTTLB.tlb_entry_invalidated

An entry in the cache has been invalidated. Fields:

entry_id unsigned int

The entry id that is being invalidated.

reason enum

Why the entry is being invalidated.

HACDBS_PROCESSING_COMPLETE_config

Configuration of interrupt updated. Fields:

address unsigned int

Address of the MSI.

data unsigned int

Data payload of the MSI.

irqen bool

Appropriate bit of SMMU_s_IRQ_CTRL controlling this interrupt.

Copyright © 2023–2025 Arm Limited (or its affiliates). All rights reserved.

Non-Confidential

memattr enum

Memory type.

ns bool

PAS of bus attribute is non-secure.

pas enum

PAS of the MSI.

ra bool

read-allocate attribute.

sh enum

Shareability.

smmu_idr0_msi bool

Are MSIs supported for this security world.

tr bool

transient attribute.

wa bool

write-allocate attributes.

HACDBS_PROCESSING_COMPLETE_irqen

The IRQEN has been changed. Fields:

new_value bool

The new value of IRQEN.

HACDBS_PROCESSING_COMPLETE_irqen_ack

The IRQEN change has been acked. Fields:

new_value bool

The new value of IRQEN.

HDBSS_TABLE_FULL_config

Configuration of interrupt updated. Fields:

address unsigned int

Address of the MSI.

data unsigned int

Data payload of the MSI.

irqen bool

Appropriate bit of SMMU_s_IRQ_CTRL controlling this interrupt.

memattr enum

Memory type.

ns bool

PAS of bus attribute is non-secure.

pas enum

PAS of the MSI.

ra bool

read-allocate attribute.

sh enum

Shareability.

smmu_idr0_msi bool

Are MSIs supported for this security world.

tr bool

transient attribute.

wa bool

write-allocate attributes.

HDBSS_TABLE_FULL_irqen

The IRQEN has been changed. Fields:

new_value bool

The new value of IRQEN.

HDBSS_TABLE_FULL_irqen_ack

The IRQEN change has been acked. Fields:

new_value bool

The new value of IRQEN.

PRIQ_config

Configuration of interrupt updated. Fields:

address unsigned int

Address of the MSI.

data unsigned int

Data payload of the MSI.

irqen bool

Appropriate bit of SMMU_s_IRQ_CTRL controlling this interrupt.

memattr enum

Memory type.

ns bool

PAS of bus attribute is non-secure.

pas enum

PAS of the MSI.

ra bool

read-allocate attribute.

sh enum

Shareability.

smmu_idr0_msi bool

Are MSIs supported for this security world.

tr bool

transient attribute.

wa bool

write-allocate attributes.

PRIQ_irqen

The IRQEN has been changed. Fields:

new_value bool

The new value of IRQEN.

PRIQ_irqen_ack

The IRQEN change has been acked. Fields:

new_value bool

The new value of IRQEN.

R_EVENTQ_config

Configuration of interrupt updated. Fields:

address unsigned int

Address of the MSI.

data unsigned int

Data payload of the MSI.

irqen bool

Appropriate bit of SMMU_s_IRQ_CTRL controlling this interrupt.

memattr enum

Memory type.

ns bool

PAS of bus attribute is non-secure.

pas enum

PAS of the MSI.

ra bool

read-allocate attribute.

sh enum

Shareability.

smmu_idr0_msi bool

Are MSIs supported for this security world.

tr bool

transient attribute.

wa bool

write-allocate attributes.

R_EVENTQ_irqen

The IRQEN has been changed. Fields:

new_value bool

The new value of IRQEN.

R_EVENTQ_irqen_ack

The IRQEN change has been acked. Fields:

new_value bool

The new value of IRQEN.

R_ERROR_config

Configuration of interrupt updated. Fields:

address unsigned int

Address of the MSI.

data unsigned int

Data payload of the MSI.

irqen bool

Appropriate bit of SMMU_s_IRQ_CTRL controlling this interrupt.

memattr enum

Memory type.

ns bool

PAS of bus attribute is non-secure.

pas enum

PAS of the MSI.

ra bool

read-allocate attribute.

sh enum

Shareability.

smmu_idr0_msi bool

Are MSIs supported for this security world.

tr bool

transient attribute.

wa bool

write-allocate attributes.

R_ERROR_irqen

The IRQEN has been changed. Fields:

new_value bool

The new value of IRQEN.

R_ERROR_irqen_ack

The IRQEN change has been acked. Fields:

new_value bool

The new value of IRQEN.

R_HACDBS_PROCESSING_COMPLETE_config

Configuration of interrupt updated. Fields:

address unsigned int

Address of the MSI.

data unsigned int

Data payload of the MSI.

irqen bool

Appropriate bit of SMMU_s_IRQ_CTRL controlling this interrupt.

memattr enum

Memory type.

ns bool

PAS of bus attribute is non-secure.

pas enum

PAS of the MSI.

ra bool

read-allocate attribute.

sh enum

Shareability.

smmu_idr0_msi bool

Are MSIs supported for this security world.

tr bool

transient attribute.

wa bool

write-allocate attributes.

R_HACDBS_PROCESSING_COMPLETE_irqen

The IRQEN has been changed. Fields:

new_value bool

The new value of IRQEN.

R_HACDBS_PROCESSING_COMPLETE_irqen_ack

The IRQEN change has been acked. Fields:

new_value bool

The new value of IRQEN.

R_HDBSS_TABLE_FULL_config

Configuration of interrupt updated. Fields:

address unsigned int

Address of the MSI.

data unsigned int

Data payload of the MSI.

irqen bool

Appropriate bit of SMMU_s_IRQ_CTRL controlling this interrupt.

memattr enum

Memory type.

ns bool

PAS of bus attribute is non-secure.

pas enum

PAS of the MSI.

ra bool

read-allocate attribute.

sh enum

Shareability.

smmu_idr0_msi bool

Are MSIs supported for this security world.

tr bool

transient attribute.

wa bool

write-allocate attributes.

R_HDBSS_TABLE_FULL_irqen

The IRQEN has been changed. Fields:

new_value bool

The new value of IRQEN.

R_HDBSS_TABLE_FULL_irqen_ack

The IRQEN change has been acked. Fields:

new_value bool

The new value of IRQEN.

R_PRIQ_config

Configuration of interrupt updated. Fields:

address unsigned int

Address of the MSI.

data unsigned int

Data payload of the MSI.

irgen bool

Appropriate bit of SMMU_s_IRQ_CTRL controlling this interrupt.

memattr enum

Memory type.

ns bool

PAS of bus attribute is non-secure.

pas enum

PAS of the MSI.

ra bool

read-allocate attribute.

sh enum

Shareability.

smmu_idr0_msi bool

Are MSIs supported for this security world.

tr bool

transient attribute.

wa bool

write-allocate attributes.

R_PRIQ_irgen

The IRQEN has been changed. Fields:

new_value bool

The new value of IRQEN.

R_PRIQ_irgen_ack

The IRQEN change has been acked. Fields:

new_value bool

The new value of IRQEN.

SMMU_CR0ACK_SMMUEN_hazarded_by_priq

The SMMU_r_CR0ACK.SMMUEN cannot acknowledge the change to SMMUEN because there are outstanding PRIQ writes.

SMMU_CR0ACK_SMMUEN_update

The acknowledge to SMMU_CR0.SMMUEN was performed. Fields:

new_value bool

The new value of the register.

SMMU_CR0_SMMUEN_old_set_complete

A set of transactions associated with the old value of SMMU_CR0.SMMUEN completed.

Fields:

last bool

This is the last set completing.

old_value bool

The old value of the set completing.

SMMU_CR0_SMMUEN_write

A write to SMMU_CR0.SMMUEN occurred. Fields:

new_value bool

The new value of the register.

old_value bool

The old value of the register.

SMMU_GBPA_old_set_complete

A set of transactions associated with the old value of SMMU_GBPA completed. Fields:

last bool

This is the last set completing.

old_value unsigned int

The old value of the set completing.

SMMU_GBPA_update

The Update flag to SMMU_GBPA was lowered. Fields:

new_value unsigned int

The new value of the register.

SMMU_GBPA_write

A write to SMMU_GBPA occurred. Fields:

new_value unsigned int

The new value of the register.

old_value unsigned int

The old value of the register.

SMMU_ROOT_CR0ACK_ACCESSSEN_update

The acknowledge to SMMU_ROOT_CR0.ACCESSSEN was performed. Fields:

new_value bool

The new value of the register.

SMMU_ROOT_CR0ACK_GPCEN_update

The acknowledge to SMMU_ROOT_CR0.GPCEN was performed. Fields:

new_value bool

The new value of the register.

SMMU_ROOT_CR0_ACCESSEN_old_set_complete

A set of transactions associated with the old value of SMMU_ROOT_CR0.ACCESSSEN completed. Fields:

last bool

This is the last set completing.

old_value bool

The old value of the set completing.

SMMU_ROOT_CR0_ACCESSEN_write

A write to SMMU_ROOT_CR0.ACCESSSEN occurred. Fields:

new_value bool

The new value of the register.

old_value bool

The old value of the register.

SMMU_ROOT_CR0_GPCEN_old_set_complete

A set of transactions associated with the old value of SMMU_ROOT_CR0.GPCEN completed. Fields:

last bool

This is the last set completing.

old_value bool

The old value of the set completing.

SMMU_ROOT_CR0_GPCEN_write

A write to SMMU_ROOT_CR0.GPCEN occurred. Fields:

new_value bool

The new value of the register.

old_value bool

The old value of the register.

SMMU_R_CR0ACK_SMMUEN_hazarded_by_rl_priq

The SMMU_r_CR0ACK.SMMUEN cannot acknowledge the change to SMMUEN because there are outstanding PRIQ writes.

SMMU_R_CR0ACK_SMMUEN_update

The acknowledge to SMMU_R_CR0.SMMUEN was performed. Fields:

new_value bool

The new value of the register.

SMMU_R_CR0_SMMUEN_old_set_complete

A set of transactions associated with the old value of SMMU_R_CR0.SMMUEN completed.

Fields:

last bool

This is the last set completing.

old_value bool

The old value of the set completing.

SMMU_R_CR0_SMMUEN_write

A write to SMMU_R_CR0.SMMUEN occurred. Fields:

new_value bool

The new value of the register.

old_value bool

The old value of the register.

SMMU_S_CR0ACK_SMMUEN_update

The acknowledge to SMMU_S_CR0.SMMUEN was performed. Fields:

new_value bool

The new value of the register.

SMMU_S_CR0_SMMUEN_old_set_complete

A set of transactions associated with the old value of SMMU_S_CR0.SMMUEN completed.

Fields:

last bool

This is the last set completing.

old_value bool

The old value of the set completing.

SMMU_S_CR0_SMMUEN_write

A write to SMMU_S_CR0.SMMUEN occurred. Fields:

new_value bool

The new value of the register.

old_value bool

The old value of the register.

SMMU_S_GBPA_old_set_complete

A set of transactions associated with the old value of SMMU_S_GBPA completed. Fields:

last bool

This is the last set completing.

old_value unsigned int

The old value of the set completing.

SMMU_S_GBPA_update

The Update flag to SMMU_S_GBPA was lowered. Fields:

new_value unsigned int

The new value of the register.

SMMU_S_GBPA_write

A write to SMMU_S_GBPA occurred. Fields:

new_value unsigned int

The new value of the register.

old_value unsigned int

The old value of the register.

S_EVENTQ_config

Configuration of interrupt updated. Fields:

address unsigned int

Address of the MSI.

data unsigned int

Data payload of the MSI.

irqen bool

Appropriate bit of SMMU_s_IRQ_CTRL controlling this interrupt.

memattr enum

Memory type.

ns bool

PAS of bus attribute is non-secure.

pas enum

PAS of the MSI.

ra bool

read-allocate attribute.

sh enum

Shareability.

smmu_idr0_msi bool

Are MSIs supported for this security world.

tr bool

transient attribute.

wa bool

write-allocate attributes.

S_EVENTQ_irqen

The IRQEN has been changed. Fields:

new_value bool

The new value of IRQEN.

S_EVENTQ_irqen_ack

The IRQEN change has been acked. Fields:

new_value bool

The new value of IRQEN.

S_GERROR_config

Configuration of interrupt updated. Fields:

address unsigned int

Address of the MSI.

data unsigned int

Data payload of the MSI.

irqen bool

Appropriate bit of SMMU_s_IRQ_CTRL controlling this interrupt.

memattr enum

Memory type.

ns bool

PAS of bus attribute is non-secure.

pas enum

PAS of the MSI.

ra bool

read-allocate attribute.

sh enum

Shareability.

smmu_idr0_msi bool

Are MSIs supported for this security world.

tr bool

transient attribute.

wa bool

write-allocate attributes.

S_GERROR_irqen

The IRQEN has been changed. Fields:

new_value bool

The new value of IRQEN.

S_ERROR_irqen_ack

The IRQEN change has been acked. Fields:

new_value bool

The new value of IRQEN.

S_HACDBS_PROCESSING_COMPLETE_config

Configuration of interrupt updated. Fields:

address unsigned int

Address of the MSI.

data unsigned int

Data payload of the MSI.

irqen bool

Appropriate bit of SMMU_s_IRQ_CTRL controlling this interrupt.

memattr enum

Memory type.

ns bool

PAS of bus attribute is non-secure.

pas enum

PAS of the MSI.

ra bool

read-allocate attribute.

sh enum

Shareability.

smmu_idr0_msi bool

Are MSIs supported for this security world.

tr bool

transient attribute.

wa bool

write-allocate attributes.

S_HACDBS_PROCESSING_COMPLETE_irqen

The IRQEN has been changed. Fields:

new_value bool

The new value of IRQEN.

S_HACDBS_PROCESSING_COMPLETE_irqen_ack

The IRQEN change has been acked. Fields:

new_value bool

The new value of IRQEN.

S_HDBSS_TABLE_FULL_config

Configuration of interrupt updated. Fields:

address unsigned int

Address of the MSI.

data unsigned int

Data payload of the MSI.

irqen bool

Appropriate bit of SMMU_s_IRQ_CTRL controlling this interrupt.

memattr enum

Memory type.

ns bool

PAS of bus attribute is non-secure.

pas enum

PAS of the MSI.

ra bool

read-allocate attribute.

sh enum

Shareability.

smmu_idr0_msi bool

Are MSIs supported for this security world.

tr bool

transient attribute.

wa bool

write-allocate attributes.

S_HDBSS_TABLE_FULL_irqen

The IRQEN has been changed. Fields:

new_value bool

The new value of IRQEN.

S_HDBSS_TABLE_FULL_irqen_ack

The IRQEN change has been acked. Fields:

new_value bool

The new value of IRQEN.

TLB.tlb_commentary

A commentary from the cache about what is going on. Fields:

output string

The stream output.

TLB.tlb_entry_invalidated

An entry in the cache has been invalidated. Fields:

entry_id unsigned int

The entry id that is being invalidated.

reason enum

Why the entry is being invalidated.

atc_inv_nop

The CMD_ATC_INV command is ignored as a **NOP**. This may be emitted multiple times if the CMD_ATC_INV is being ignored for multiple reasons. Fields:

address unsigned int

Untranslated Address to invalidate.

cmd_id unsigned int

Command id.

cons unsigned int

CONS of the command.

global bool

Global flag.

log2_size_in_bytes unsigned int

Log 2 of the size in bytes of the region to invalidate.

ssd_of_queue enum

The SSD of the CMDQ.

ssd_of_streamid enum

The SSD of the StreamID.

streamid unsigned int

StreamID in the invalidate command.

substreamid unsigned int

SubstreamID or ~0u if SSValid == false.

why enum

Why was NOPped.

atos_complete_fault

The ATOS operation completed with a fault. Fields:

effective_st1translate bool

Because of the settings and/or the ATOS type then the effective st1translate can be different.

fault_faddr unsigned int

The fault FADDR.

fault_faultcode enum

The fault code.

fault_reason enum

The fault reason.

ssd_ns bool

This is a non-secure ATOS operation.

st1translate bool

The StreamID/SubstreamID combination should have been translated by stage 1.

st2translate bool

The translation suffered a stage 2 translation.

state enum

The final transaction state of the ATOS operation.

vatos_vmid unsigned int

The VMID of the VATOS operation, or ~0u if not a VATOS operation.

atos_complete_fault_inv_req

The ATOS operation completed, faulted and generated an INV_REQ response. Fields:

ssd_ns bool

This is a non-secure ATOS operation.

vatos_vmid unsigned int

The VMID of the VATOS operation, or ~0u if not a VATOS operation.

why enum

Why the ATOS operation generated an INV_REQ.

atos_complete_success

The ATOS operation completed successfully. Fields:

base_addr unsigned int

The actual base address of region.

effective_st1translate bool

Because of the settings and/or the ATOS type then the effective st1translate can be different.

par_addr unsigned int

The PAR.ADDR field.

par_mair unsigned int

The memory attributes encoded as a MAIR.

par_ns bool

The PAR.NS field, for an SSD-ns request then this will always be 0.

par_sh enum

Shareability.

par_size bool

The PAR.Size field.

size_in_bytes unsigned int

The actual size in bytes of the region.

ssd_ns bool

This is a non-secure ATOS operation.

st1translate bool

The StreamID/SubstreamID combination should have been translated by stage 1.

st2translate bool

The translation suffered a stage 2 translation.

state enum

The final transaction state of the ATOS operation.

vatos_vmid unsigned int

The VMID of the VATOS operation, or ~0u if not a VATOS operation.

atos_run_set

The SMMU_s_GATOS_CTRL.RUN field was set to start the ATOS operation. Fields:

ssd_ns bool

This is a non-secure ATOS operation.

vatos bool

This is a VATOS operation.

atos_starting

The ATOS operation is starting. Fields:

addr unsigned int

The input address to the ATOS operation.

httui bool

Inhibit HTTU update.

ind bool

Instruction Data.

pnu bool

Privileged not User.

rnw bool

Read not Write.

ssd_ns bool

This is a non-secure ATOS operation.

ssec bool

If this is a secure ATOS operation then this is if it is secure or not.

streamid unsigned int

The StreamID requested.

substreamid unsigned int

The SubstreamID requested, or ~0u if no SubstreamID.

type enum

The requested ATOS type.

vatos_vmid unsigned int

The VMID of the VATOS operation, or ~0u if not a VATOS operation.

axi_stream_msi_addr_to_match_s

Address to use to send SMMU originated MSIs directly to the GIC. This pin is sampled on the negedge of the reset pin and has to be driven beforehand for the value to take effect. Fields:

value unsigned int

The value of the signal.

cd_cc.CD_commentary

A commentary from the cache about what is going on. Fields:

output string

The stream output.

cd_cc.CD_entry_invalidated

An entry in the cache has been invalidated. Fields:

entry_id unsigned int

The entry id that is being invalidated.

reason enum

Why the entry is being invalidated.

cd_entry_allocated

An CD entry has been allocated. Fields:

AssuredTranslation bool

The CD (and any L1CD) was fetched from an AssuredOnly stage 2 page and so might be capable of creating AssuredTranslations.

cd string

A textual description of the CD.

entry_id unsigned int

The id of the cache entry. Top bit is set if is a debug entry.

ns enum

For the non-secure world.

ssd enum

The SSD of the entry.

streamid unsigned int

The StreamID of the entry this will match.

substreamid unsigned int

The SubstreamID of the entry this will match. This may be zero for transactions without a SubstreamID.

conf_system_supports_bgptm

The pin is driven. This indicates system supports broadcast TLBI by PA operations. This pin is sampled on the negedge of the reset pin and has to be driven beforehand for the value to take effect. Fields:

value bool

The value of the signal.

dpttlb_entry_allocated

A DPT TLB entry has been allocated. Fields:

AC enum

The value of the 'AC' field that controls access to this region.

FWB bool

The region is FWB.

VMID unsigned int

The VMID, if any, associated with this region.

index unsigned int

Index of the TLB entry.

input_end_incl_address unsigned int

The end inclusive address of the input range that this matches.

input_start_address unsigned int

The start address of the input range that this matches.

output_pas enum

The output PAS of this DPT region.

ssd enum

The SSD of the streams this region captures.

trans_id unsigned int

The trans_id of the transaction that caused this allocation.

vmsa_formed bool

The entry was formed from VMSA information rather than from a DPT walk.

writeable bool

True if this region is writeable.

dpttlb_invalidate_intersects_but_does_not_cover_entry_range

ENCODED_SIZE < size of the region covered by the DPT entry so invalidation is not architecturally guaranteed. No invalidation is performed. Fields:

dpttlb_entry_id unsigned int

ID of the TLB entry.

entry_end_incl_address unsigned int

Last address covered by the TLB entry.

entry_start_address unsigned int

First address covered by the TLB entry.

invalidate_end_incl_address unsigned int

Last address covered by the invalidation range.

invalidate_start_address unsigned int

First address covered by the invalidation range.

ssd enum

The security state of the TLB entry.

dpttlb_overlapping_entries

Two DPT TLB entries are overlapping. Fields:

new_AC enum

The value of the 'AC' field that controls access to this region.

new_FWB bool

The region is FWB.

new_VMID unsigned int

The VMID, if any, associated with this region.

new_entry_index unsigned int

The entry index of the new DPT TLB entry.

new_entry_is_vmsa_formed bool

The new entry was inserted as a result of VMSA information.

new_input_end_incl_address unsigned int

The end inclusive address of the input range that this matches.

new_input_start_address unsigned int

The start address of the input range that this matches.

new_output_pas enum

The output PAS of this DPT region.

new_ssd enum

The SSD of the streams this region captures.

new_writeable bool

True if this region is writeable.

old_AC enum

The value of the 'AC' field that controls access to this region.

old_FWB bool

The region is FWB.

old_VMID unsigned int

The VMID, if any, associated with this region.

old_entry_index unsigned int

The entry index of the old DPT TLB entry.

old_entry_is_vmsa_formed bool

The old entry was inserted as a result of VMSA information.

old_input_end_incl_address unsigned int

The end inclusive address of the input range that this matches.

old_input_start_address unsigned int

The start address of the input range that this matches.

old_output_pas enum

The output PAS of this DPT region.

old_ssd enum

The SSD of the streams this region captures.

old_writeable bool

True if this region is writeable.

dvm_tlbinvalidate_complete

The DVM TLB Invalidate message completed. Fields:

id unsigned int

The unique id of this DVM message.

ok bool

The DVM message was OK.

dvm_tlbinvalidate_received

A DVM message for a TLB Invalidate has been received. Fields:

address unsigned int

The VA or IPA to use if match_address.

asid unsigned int

The ASID to match if match_asid.

by_ipa bool

The operation is for an IPA operation if match_address.

id unsigned int

The unique id of this DVM message.

ignored enum

The DVM message was ignored.

last_level bool

The operation is for last level if supported.

match_address bool

Match the address field.

match_asid bool

Match the asid field.

match_vmid bool

Match the vmid field.

num unsigned int

If a range operation, the NUM field. If a single-address operation this is 0.

prot enum

The protection level for which this TLB Invalidate will operate on.

security_world enum

The security world that this will apply to.

smmu_scale unsigned int

If a range operation, then the SCALE field with the meaning in the SMMU architecture which is different to the PE architecture. If a single-address operation this is 0.

stage1_only bool

The operation is for stage 1 only if supported.

tg enum

If a single-address or address-range operation, then the Translation Granule hint. Address-range operations always supply a Translation Granule.

translation_table_level enum

The leaf level of the translation table.

vmid unsigned int

The VMID to match if match_vmid.

found_tlb_entry_has_different_aset

Architecturally, a particular ASID either should be ASET0 or ASET1. However, we have managed to find a TLB entry that has a different ASET than that which we were searching for. This indicates a programming error. You should examine all contexts with this particular ASID/VMID and ensure they are consistent. Fields:

aset unsigned int

The ASET of this entry.

asid unsigned int

ASID if appropriate.

index unsigned int

Index of TLB entry.

input_start_address unsigned int

The start address of the input range that this matches.

scheme enum

The tagging scheme used.

ssd enum

The SSD of the tagging scheme.

ssd_ns enum

Security State.

vmid unsigned int

VMID if appropriate.

gpf_far

An attempt was made to record a fault in SMMU_ROOT_GPF_FAR. Fields:

fault_address unsigned int

The PA input address that faulted.

fpas enum

The PAS of the input PA that suffered the fault.

is_debug bool

The record is due to a debug transaction.

lost bool

The record is lost because an existing record is active. If `is_debug`, then the record is never written, but this reports if it would have been lost if it was a real transaction.

syndrome enum

The syndrome for the fault.

gpt_cfg_far

An attempt was made to record a fault in SMMU_ROOT_GPT_CFG_FAR. Fields:

cfg_err enum

The configuration syndrome.

fault_address unsigned int

The PA input address that faulted.

fpas enum

The PAS of the input PA that suffered the fault.

is_debug bool

The record is due to a debug transaction.

lost bool

The record is lost because an existing record is active. If `is_debug`, then the record is never written, but this reports if it would have been lost if it was a real transaction.

syndrome enum

The syndrome for the fault.

gpt_read_block_descriptor

A GPT read has completed and found an LOGPT block descriptor. Fields:

address_checking unsigned int

The address that we are checking.

address_gptdesc unsigned int

The address of the GPT descriptor.

begin_address unsigned int

The address range this block covers.

data unsigned int

The data fetch if it didn't abort.

end_incl_address unsigned int

The address range this block covers.

gpi enum

GPI.

gpi_violation bool

The descriptor is invalid because the GPI value is invalid.

has_res0_violation bool

The descriptor is invalid because of **RES0** violation.

pas_checking enum

The PAS of the address that we are checking.

trans_id unsigned int

The `trans_id` of the transaction we are checking if appropriate.

what_checking enum

What kind of access are we checking.

gpt_read_contig_descriptor

A GPT read has completed and found an L1GPT contig descriptor. Fields:

address_checking unsigned int

The address that we are checking.

address_gptdesc unsigned int

The address of the GPT descriptor.

begin_address unsigned int

The address range this block covers.

data unsigned int

The data fetch if it didn't abort.

end_incl_address unsigned int

The address range this block covers.

gpi enum

GPI.

gpi_violation bool

The descriptor is invalid because the GPI value is invalid.

has_res0_violation bool

The descriptor is invalid because of **RES0** violation.

pas_checking enum

The PAS of the address that we are checking.

size_violation bool

The encoded size was not a valid value.

trans_id unsigned int

The trans_id of the transaction we are checking if appropriate.

what_checking enum

What kind of access are we checking.

gpt_read_granules_descriptor

A GPT read has completed and found an L1GPT granules descriptor. Fields:

address_checking unsigned int

The address that we are checking.

address_gptdesc unsigned int

The address of the GPT descriptor.

begin_address unsigned int

The address range this block covers.

data unsigned int

The data fetch if it didn't abort.

end_incl_address unsigned int

The address range this block covers.

gpi enum

GPI.

gpi_violation bool

The descriptor is invalid because the GPI value is invalid.

pas_checking enum

The PAS of the address that we are checking.

trans_id unsigned int

The trans_id of the transaction we are checking if appropriate.

what_checking enum

What kind of access are we checking.

which_gpi_granule unsigned int

Which GPI granule the address being checked will use.

gpt_read_table_descriptor

A GPT read has completed and found an LOGPT table descriptor. Fields:

address_checking unsigned int

The address that we are checking.

address_gptdesc unsigned int

The address of the GPT descriptor.

data unsigned int

The data fetch if it didn't abort.

has_res0_violation bool

The descriptor is invalid because of **RES0** violation.

misaligned_violation bool

The descriptor is invalid because the base address is misaligned.

out_of_range_violation bool

The descriptor is invalid because the base address is out of range of the PPS.

pas_checking enum

The PAS of the address that we are checking.

table_address unsigned int

The table address.

trans_id unsigned int

The trans_id of the transaction we are checking if appropriate.

what_checking enum

What kind of access are we checking.

gpttlb_entry_allocated

A GPT TLB entry has been allocated. Fields:

GPI enum

The GPI of the entry.

index unsigned int

Index of TLB entry.

input_end_incl_address unsigned int

The end inclusive address of the input range that this matches.

input_start_address unsigned int

The start address of the input range that this matches.

origin enum

How this entry was formed.

httu_update_abandoned_update

The HTTU update of a descriptor in memory was potentially possible, but it was behind an update that failed to apply cleanly. Fields:

AF enum

Whether an AF update should/could occur.

DBM enum

Whether a DBM update should/could occur.

address unsigned int

Address of HTTU update.

httu_inner enum

Inner cacheability of descriptor to update.

httu_outer enum

Outer cacheability of descriptor to update.

httu_prot enum

The instruction/data, bus-ns/s and privileged/user marking.

httu_sh enum

Shareability of descriptor to update.

mpam_pmg_and_partid unsigned int

Bits[23:16] are the MPAM PMG. Bits[15:0] are the physical PARTID.

mpam_sp enum

The MPAM_SP of the transaction.

original_descriptor unsigned int

The original descriptor that the model observed.

stage_and_level unsigned int

The stage (bits[7:4]) and level (bits[3:0]). A level of 0xF means -1.

trans_id unsigned int

The transaction id that this HTTU update is associated with.

what enum

What this descriptor represents.

httu_update_end_update

The attempted update of a descriptor in memory has occurred. Fields:

is_big_endian bool

The descriptor is big-endian in memory.

original_descriptor unsigned int

The original descriptor value.

result enum

The result of the attempt to update.

trans_id unsigned int

The transaction id that this HTTU update is associated with.

try_to_change_to_descriptor unsigned int

The new descriptor value that tried to replace the original.

value_that_was_in_memory unsigned int

The value that the compare-and-swap operation returned as the value that was in memory.

httu_update_not_done

A discretionary HTTU update could occur and the implementation choose not to do it. A mandatory_do_if_linked_performed means that this is a stage 2 leaf descriptor that maps a subsequent stage 1 leaf descriptor whose update is discretionary and if that subsequent discretionary update is going to occur then this update becomes mandatory. Fields:

AF enum

Whether an AF update should/could occur.

DBM enum

Whether a DBM update should/could occur.

address unsigned int

Address of HTTU update.

httu_inner enum

Inner cacheability of descriptor to update.

httu_outer enum

Outer cacheability of descriptor to update.

httu_prot enum

The instruction/data, bus-ns/s and privileged/user marking.

httu_sh enum

Shareability of descriptor to update.

mpam_pmg_and_partid unsigned int

Bits[23:16] are the MPAM PMG. Bits[15:0] are the physical PARTID.

mpam_sp enum

The MPAM_SP of the transaction.

original_descriptor unsigned int

The original descriptor that the model observed.

stage_and_level unsigned int

The stage (bits[7:4]) and level (bits[3:0]). A level of 0xF means -1.

trans_id unsigned int

The transaction id that this HTTU update is associated with.

what enum

What this descriptor represents.

httu_update_start_update

An HTTU update could occur and the implementation chose to try it. A mandatory_do_if_linked_performed means that this is a stage 2 leaf descriptor that maps a subsequent stage 1 leaf descriptor whose update is discretionary and if that subsequent discretionary update is going to occur then this update becomes mandatory. Fields:

AF enum

Whether an AF update should/could occur.

DBM enum

Whether a DBM update should/could occur.

address unsigned int

Address of HTTU update.

httu_inner enum

Inner cacheability of descriptor to update.

httu_outer enum

Outer cacheability of descriptor to update.

httu_prot enum

The instruction/data, bus-ns/s and privileged/user marking.

httu_sh enum

Shareability of descriptor to update.

is_big_endian bool

The descriptor will be written to memory as big-endian.

mecid unsigned int

The masked MECID used for the update transaction, or ~0u if not appropriate.

mpam_pmg_and_partid unsigned int

Bits[23:16] are the MPAM PMG. Bits[15:0] are the physical PARTID.

mpam_sp enum

The MPAM_SP of the transaction.

original_descriptor unsigned int

The original descriptor that the model observed.

stage_and_level unsigned int

The stage (bits[7:4]) and level (bits[3:0]). A level of 0xF means -1.

trans_id unsigned int

The transaction id that this HTTU update is associated with.

try_to_change_to_descriptor unsigned int

The new descriptor value that is going to try replace the original.

what enum

What this descriptor represents.

will_do_AF bool

What the implementation chose to do for the AF update.

will_do_DBM bool

What the implementation chose to do for the DBM update.

integration_mode_end_ras_level_interrupt_restored

RAS level sensitive interrupt restored due to integration mode ending. Fields:

is_tcu bool

Is TCU integration mode.

ras_interrupt enum

RAS interrupt being restored.

tbu_index unsigned int

TBU index. ~0u if it's TCU.

value bool

Level of the interrupt signal.

integration_mode_pmcg_interrupt_lost

PMCG interrupt lost due to being in integration mode. Fields:

is_tcu enum

Is TCU integration mode.

pmcg_interrupt enum

PMCG interrupt being dropped.

tbu_index unsigned int

TBU index. ~0u if it's TCU.

integration_mode_pmusnapshot_ack_lost

pmusnapshot_ack lost due to being in integration mode. Fields:

is_tcu enum

Is TCU integration mode.

tbu_index unsigned int

TBU index. ~0u if it's TCU.

integration_mode_ras_interrupt_lost

RAS interrupt lost due to being in integration mode. Fields:

is_tcu enum

Is TCU integration mode.

ras_interrupt enum

RAS interrupt being dropped.

tbu_index unsigned int

TBU index. ~0u if it's TCU.

integration_mode_ras_level_interrupt_lost

RAS level sensitive interrupt lost due to being in integration mode. Fields:

is_tcu bool

Is TCU integration mode.

ras_interrupt enum

RAS interrupt being dropped.

tbu_index unsigned int

TBU index. ~0u if it's TCU.

value bool

Level of the interrupt signal.

integration_mode_start_ras_level_interrupt_cleared

RAS level sensitive interrupt cleared due to integration mode starting. Fields:

is_tcu bool

Is TCU integration mode.

ras_interrupt enum

RAS interrupt being cleared.

tbu_index unsigned int

TBU index. ~0u if it's TCU.

integration_mode_tcu_evento_lost

Evento lost due to being in integration mode.

integration_mode_tcu_interrupt_lost

Interrupt lost due to being in integration mode. Fields:

interrupt enum

Interrupt that is being dropped.

integration_register_change

An integration register write occurred and it's modified, which drives a signal. Fields:

is_tcu enum

Is TCU integration mode.

new_value bool

New value of the signal.

old_value bool

Old value of the signal.

register enum

The register being modified.

register_value unsigned int

Value written to the register.

signal string

Signal affected by the register write.

tbu_index unsigned int

TBU index. ~0u if it's TCU.

interrupt_returned

An interrupt/MSI returned from downstream. Fields:

InD_NS_PnU enum

Instruction/Data, Non-secure/Secure/Root/Realm, Privileged/User. Bit[1] = PAS[0], and bit[3] = PAS[1].

address unsigned int

Address of the MSI.

data unsigned int

Data payload of the MSI.

dcmdqcp_and_index unsigned int

If this not a DCMDQ MSI CMD_SYNC, then this is 0xFFFFffff. If it is then the DCMDQ QCP index is in bits[31:16] and the index in the QCP is in [7:0].

gpc_denied_msi bool

True if the MSI was denied as it failed its GPC checks. Thus the field 'ok' will be false.

id unsigned int

ID of this interrupt transaction.

inner_cache enum

The actual attributes of the access.

kind enum

What kind of interrupt.

mecid unsigned int

The MECID of the MSI, if applicable.

mpam_ns bool

For a non-RME system, this is true if the MPAM space is Non-secure. For an RME system, this is MPAM_SP[0].

mpam_pmg_and_partid unsigned int

Bits[23:16] are the MPAM PMG. Bits[15:0] are the physical PARTID.

mpam_sp enum

The MPAM_SP security state of the MPAM partition. 0 – secure, 1 – non-secure, 2 – root, 3 – realm.

msi_valid bool

MSI will attempt to be sent.

ok bool

Did the access return OK or an abort?.

outer_cache enum

The actual attributes of the access.

pas enum

The PAS of the MSI, if applicable.

pmcg_index_and_counter unsigned int

If this is a PMCG interrupt, then the top 16 bits are the PMCG index, the lower 16 bits are counter index.

ras_error_group_id_node_id_and_record_index_in_node unsigned int

If this is a RAS interrupt then this is the error group record id in bits[23:16], the node_id in [15:8] and the record index in the node in [7:0].

sh enum

Shareability.

interrupt_sent

An interrupt is raised. If it sends an MSI then this is *after* any device-dependent transform on the architectural attributes and so may differ from what is programmed. Fields:

InD_NS_PnU enum

Instruction/Data, Non-secure/Secure/Root/Realm, Privileged/User. Bit[1] = PAS[0], and bit[3] = PAS[1].

address unsigned int

Address of the MSI.

data unsigned int

Data payload of the MSI.

dcmdq_device_id unsigned int

If this is a DCMDQ then this is the DeviceID of the MSI write. Otherwise, 0xFFFF'ffff.

dcmdq_qcp_and_index unsigned int

If this not a DCMDQ MSI CMD_SYNC, then this is 0xFFFF'ffff. If it is then the DCMDQ QCP index is in bits[31:16] and the index in the QCP is in [7:0].

id unsigned int

ID of this interrupt transaction.

inner_cache enum

The actual attributes of the access.

kind enum

What kind of interrupt.

mecid unsigned int

The MECID of the MSI, if applicable.

mpam_ns bool

For a non-RME system, this is true if the MPAM space is Non-secure. For an RME system, this is MPAM_SP[0].

mpam_pmg_and_partid unsigned int

Bits[23:16] are the MPAM PMG. Bits[15:0] are the physical PARTID.

mpam_sp enum

The MPAM_SP security state of the MPAM partition. 0 – secure, 1 – non-secure, 2 – root, 3 – realm.

msi_valid bool

MSI will attempt to be sent.

outer_cache enum

The actual attributes of the access.

pas enum

The PAS of the MSI, if applicable.

pmcg_index_and_counter unsigned int

If this is a PMCG interrupt, then the top 16 bits are the PMCG index, the lower 16 bits are counter index.

qSID unsigned int

If this is a DCMDQ then this is the qSID of the MSI write. Otherwise, 0xFFFF'ffff.

ras_error_group_id_node_id_and_record_index_in_node unsigned int

If this is a RAS interrupt then this is the error group record id in bits[23:16], the node_id in [15:8] and the record index in the node in [7:0].

sh enum

Shareability.

logptsz

Log2size in bytes of the region covered by an LOGPT entry. The default value is rme_logpt_entry_covers_log2size_in_bytes. This is the value reported in SMMU_ROOT_GPT_BASE_CFG.LOGPTSZ. This pin is sampled on the negedge of the reset pin and has to be driven beforehand for the value to take effect. Fields:

value unsigned int

The value of the signal.

l1cd_cc.L1CD_commentary

A commentary from the cache about what is going on. Fields:

output string

The stream output.

l1cd_cc.L1CD_entry_invalidated

An entry in the cache has been invalidated. Fields:

entry_id unsigned int

The entry id that is being invalidated.

reason enum

Why the entry is being invalidated.

l1cd_entry_allocated

An L1 CD entry has been allocated. Fields:

AssuredTranslation bool

The L1CD was fetched from an AssuredOnly stage 2 page and so might be capable of creating AssuredTranslations.

entry_id unsigned int

The id of the cache entry. Top bit is set if is a debug entry.

normalised_substreamid unsigned int

The first SubstreamID of the range of SubstreamIDs that this L1CD entry will match.

ns enum

For the non-secure world.

pa_12 unsigned int

The PA of the L2 CD table. This is L2Ptr << 12.

ssd enum

The SSD of the entry.

streamid unsigned int

The StreamID this CD is for.

valid bool

Is the entry valid.

l1ste_cc.L1STE_commentary

A commentary from the cache about what is going on. Fields:

output string

The stream output.

l1ste_cc.L1STE_entry_invalidated

An entry in the cache has been invalidated. Fields:

entry_id unsigned int

The entry id that is being invalidated.

reason enum

Why the entry is being invalidated.

l1ste_entry_allocated

An L1 STE entry has been allocated. Fields:

entry_id unsigned int

The id of the cache entry. Top bit is set if is a debug entry.

first_streamid_of_range unsigned int

The first StreamID of the range of StreamIDs that this L1STE entry will match.

ns enum

For the non-secure world.

num_entries_in_l2 unsigned int

The number of entries in the L2 table. This is $2^{(\text{Span}-1)}$ or 0 if invalid.

pa_l2 unsigned int

The PA of the L2 ST table. This is $\text{L2Ptr} \ll 6$ and aligned to the size of the table.

ssd enum

The SSD of the entry.

legacy_tz_en

The pin is driven. If high then the SMMU does not support RME. If low and the ID codes indicate support, then the SMMU supports RME. This pin is sampled on the negedge of the reset pin and has to be driven beforehand for the value to take effect. Fields:

value bool

The value of the signal.

level_interrupt_sent

A level interrupt changed state. Fields:

kind enum

What kind of interrupt.

ras_error_group_id_node_id_and_record_index_in_node unsigned int

If this is a level RAS interrupt then this is the error group record id in bits[23:16], the node_id in [15:8] and the record index in the node in [7:0].

set_high bool

Level interrupt state.

ns_cmd_sync_completed_irq

"Non-secure" CMD_SYNC completed its IRQ action. Fields:

cmd_id unsigned int

The ID of the command being issued.

cons_incl_wrap unsigned int

CONS pointer of CMD_SYNC.

is_synthetic bool

This is a synthetic sync. The cons_incl_wrap is 0xFFFF'ffff.

ns_cmd_sync_completed_no_action

Non-secure CMD_SYNC completed, there was no CMD_SYNC action (SIG_NONE). Fields:

cmd_id unsigned int

The ID of the command being issued.

cons_incl_wrap unsigned int

CONS pointer of CMD_SYNC.

is_synthetic bool

This is a synthetic sync. The cons_incl_wrap is 0xFFFF'ffff.

ns_cmd_sync_completed_sev

Non-secure CMD_SYNC completed its action of SIG_SEV. Fields:

cmd_id unsigned int

The ID of the command being issued.

cons_incl_wrap unsigned int

CONS pointer of CMD_SYNC.

is_synthetic bool

This is a synthetic sync. The cons_incl_wrap is 0xFFFF'ffff.

ns_cmd_sync_completed_with_error

Non-secure CMD_SYNC completed, however, there was an error associated with the completion of the CMD_SYNC. Fields:

error enum

The error associated with the CMD_SYNC.

cmd_id unsigned int

The ID of the command being issued.

cons_incl_wrap unsigned int

CONS pointer of CMD_SYNC.

is_synthetic bool

This is a synthetic sync. The cons_incl_wrap is 0xFFFF'ffff.

ns_cmd_sync_issued

Non-secure CMD_SYNC has now completed issuing and we now know if something is hazarding the CMD_SYNC from completing. Fields:

cmd_id unsigned int

The ID of the command being issued.

cons_incl_wrap unsigned int

CONS pointer of CMD_SYNC.

hazarded bool

The CMD_SYNC is currently hazarded from completing.

is_synthetic bool

This is a synthetic sync. The cons_incl_wrap is 0xFFFF'ffff.

ns_cmd_sync_issuing

Non-secure CMD_SYNC is starting to issue, we are going to start accounting for which things have to be part of the prior set of things that have to complete before the CMD_SYNC can complete. Fields:

cmd_id unsigned int

The ID of the command being issued.

cons_incl_wrap unsigned int

CONS pointer of CMD_SYNC.

is_synthetic bool

This is a synthetic sync. The cons_incl_wrap is 0xFFFF'ffff.

ns_cmd_sync_starting_completion_action_irq

Non-secure CMD_SYNC starting IRQ completion action. Fields:

cmd_id unsigned int

The ID of the command being issued.

cons_incl_wrap unsigned int

CONS pointer of CMD_SYNC.

is_synthetic bool

This is a synthetic sync. The cons_incl_wrap is 0xFFFF'ffff.

msiaddress unsigned int

The MSI address to use (or 0 if not sending an MSI).

msidata unsigned int

The MSI data to send.

ns_cmd_sync_starting_completion_action_sev

Non-secure CMD_SYNC starting its completion action of SIG_SEV. Fields:

cmd_id unsigned int

The ID of the command being issued.

cons_incl_wrap unsigned int

CONS pointer of CMD_SYNC.

is_synthetic bool

This is a synthetic sync. The cons_incl_wrap is 0xFFFF'ffff.

ns_cmdq_cmd_sync_error

The CMD_SYNC can fault if an ATC Invalidate timeout occurs. The timeout will be attached to the next CMD_SYNC on the queue on which the invalidate was launched. Fields:

error enum

The CERROR to attach.

cmd_queue unsigned int

The top bits indicate the CMDQ id. These bits are attached to all cmd_ids generated for this queue. The bits [63:44] contain the queue ID. [63:62] 0: s-ssd, 1: reserved, 2: ns-ssd. 3: rl-ssd [61] is ECMDQ [59:52] QCP index [51:44] ECMDQ index in QCP.

herror enum

The HERROR to attach, or HERROR_NONE if we are attaching a CERROR.

ns_cmdq_cmd_sync_error_delivered

The CMD_SYNC can fault if an ATC Invalidate timeout occurs. This is the CMD_SYNC delivering that fault. Fields:

cerror enum

The CERROR to attach.

cmd_id unsigned int

Command ID of the CMD_SYNC.

ns_cmdq_ecmdq_enack

Trace the changes to ENACK. Fields:

cerror enum

The error being exposed, if any.

cmd_queue unsigned int

The top bits indicate the CMDQ id. These bits are attached to all cmd_ids generated for this queue. The bits [63:44] contain the queue ID. [63:62] 0: s-ssd, 1: reserved, 2: ns-ssd. 3: rl-ssd [61] is ECMDQ [59:52] QCP index [51:44] ECMDQ index in QCP.

enack bool

The value of ENACK.

forced_error bool

The queue was enabled with ERR != ERRACK and so the implementation chooses to trigger the error reporting mechanism.

ns_cmdq_issue

A command is actually being executed. Fields:

cmd_id unsigned int

Command ID of the command being executed.

cons unsigned int

CONS of the command.

what string

What is being executed.

ns_cmdq_pointers_state

The command queue pointers. Fields:

cmd_queue unsigned int

The top bits indicate the CMDQ id. These bits are attached to all cmd_ids generated for this queue. The bits [63:44] contain the queue ID. [63:62] 0: s-ssd, 1: reserved, 2: ns-ssd. 3: rl-ssd [61] is ECMDQ [59:52] QCP index [51:44] ECMDQ index in QCP.

disable_fetch bool

The model has decided that as you programmed PROD/CONS inconsistently that it would not fetch as it has no confidence as to which are commands and which are not.

q_issuable_and_issued_cons unsigned int

The CONS pointer of the region that is issuable and/or issued (including wrap).

q_issuable_and_issued_prod unsigned int

The PROD pointer of the region that is issuable and/or issued (including wrap).

q_programmer_cons unsigned int

The CONS pointer as seen by the programmer (including wrap).

q_programmer_prod unsigned int

The PROD pointer as seen by the programmer (including wrap).

q_unfetched_cons unsigned int

The CONS pointer of the region unfetched (including wrap).

q_unfetched_prod unsigned int

The PROD pointer of the region unfetched (including wrap).

ns_cmdq_state

A description of what the CMDQ can do now. Fields:

can_cerror bool

There is a CERROR that can be recognised.

can_cmd_sync bool

There is a CMD_SYNC that can be recognised.

can_fetch bool

The SMMU can fetch.

can_herror bool

There is an HERROR that can be recognised.

can_issue bool

There are fetched commands that are eligible to be issued.

can_update_cons bool

The programmer view of CMDQ_CONS can be moved to say some have been consumed.

cmd_queue unsigned int

The top bits indicate the CMDQ id. These bits are attached to all cmd_ids generated for this queue. The bits [63:44] contain the queue ID. [63:62] 0: s-ssd, 1: reserved, 2: ns-ssd. 3: rl-ssd [61] is ECMDQ [59:52] QCP index [51:44] ECMDQ index in QCP.

cmd_sync enum

Why a CMD_SYNC cannot be issued.

cmdqen bool

CMDQEN, when enabled then the CMDQ can fetch commands.

cmdqen_1_to_0_unacknowledged bool

CMDQEN has been set to 0, but we have yet to acknowledge it.

current-real-cerror enum

The current CERROR that would be visible to the programmer if there was an error.

fetch-disabled_due_to_programmer_error bool

The PROD/CONS pointer was inconsistent and the model decided to stop processing commands.

fetch-number_of_unfetched_commands unsigned int

The number of commands that have yet to be fetched from memory, or ~0u if CMDQEN == 0.

fetch-state enum

The state of the fetch state machine.

fetch-too_many_commands_outstanding_so_not_fetching_any_more bool

If there are too many issued commands, or pending to be issued then we won't fetch any more.

issue-number_of_issuable_commands unsigned int

The number of commands that have been fetched but not yet issued.

issue-number_of_unissuable_commands unsigned int

This might include commands behind an illegal command, or behind a CMD_SYNC. Under some circumstances this can include a command representing an external abort.

issue-state enum

The internal issued state machine state.

outstanding_fetches_since_reset unsigned int

The number of outstanding fetches.

synthetic_sync_can_be_completed bool

The synthetic sync that is outstanding can be completed.

waiting_for_synthetic_sync bool

There is a synthetic sync that is outstanding.

ns_eventq_adding_to_cmd_sync_prior_set

The event created by the specified transaction id must become visible to the programmer before the CMD_SYNC can complete. Fields:

trans_id unsigned int

The transaction id that caused the event.

ns_eventq_auto_retry_stalled_transaction

A transaction that was stalled but not yet reported on the Event Queue was retried. The retry can happen if the transaction didn't write because it was de-duplicated behind a reported stalling transaction and that stalling transaction was CMD_RESUMEd, or if the queue was unwriteable and the queue is now writeable. Fields:

trans_id unsigned int

Transaction id of the auto-retried transaction.

why enum

Why the transaction retried.

ns_eventq_cmd_sync_unhazarded

The CMD_SYNC has been unhazarded as the prior set is empty and as much of it as required is visible in the programmers' view. Fields:

prod_incl_wrap unsigned int

The prod index including wrap bit that was required to become visible.

prog_prod_incl_wrap unsigned int

The current programmer-visible prod including wrap bit.

ns_eventq_eventqen_and_eventqenack

Trace the values of EVENTQEN and EVENTQENACK as they change. Fields:

about enum

Some more information about the queue state.

cons_incl_wrap unsigned int

The CONS pointer (including wrap bits).

eventqen bool

EVENTQEN.

eventqenack bool

EVENTQENACK.

prod_incl_wrap unsigned int

The PROD pointer (including wrap bits).

ns_eventq_external_abort

A particular eventq record got an external abort. The event will subsequently appear in the ns/s_eventq_lost_event_records trace. Fields:

prod unsigned int

The prod pointer (excluding wrap bit) of the event record that aborted.

trans_id unsigned int

The transaction id that produced the event record that aborted.

ns_eventq_lost_event_records

Event records that are lost and never record appear on this trace. Fields:

dword0 unsigned int

Double word 0 of event.

dword1 unsigned int

Double word 1 of event.

dword2 unsigned int

Double word 2 of event.

dword3 unsigned int

Double word 3 of event.

event enum

Event code that was thrown away.

prod unsigned int

The PROD index of the event if appropriate or ~0u otherwise.

reason enum

The reason the record was lost.

trans_id unsigned int

The transaction id that this corresponds to.

ns_eventq_overflow_acknowledged

When events are lost then they appear on the eventq_lost_event_records_* trace. This will activate the overflow flag in the corresponding SMMU_s_EVENTQ_PROD register. Software can then acknowledge that flag using the overflow acknowledge flag in SMMU_s_EVENTQ_CONS. Fields:

bad bool

If true then the programmer has attempted to acknowledge an overflow condition that didn't exist.

count_of_events_lost_due_to_overflow unsigned int

This is the number of events that were lost because the event queue was full. This does not include any lost because of an EVENTQ_ABT_ERR.

ns_eventq_pending_event_records

Event records that are pending to be dealt with by the eventq state machine. A specific event can be traced multiple times if it becomes part of the prior set of transactions of a CMD_SYNC that has to complete before the CMD_SYNC is allowed to be completed. Fields:

dword0 unsigned int

Double word 0 of event.

dword1 unsigned int

Double word 1 of event.

dword2 unsigned int

Double word 2 of event.

dword3 unsigned int

Double word 3 of event.

event enum

Event code.

is_prior_set bool

The transaction forms part of the prior set of a CMD_SYNC that have to become visible (or thrown away if necessary) before the CMD_SYNC can complete.

trans_id unsigned int

The transaction id that this corresponds to.

ns_eventq_prod_cons_updated

This records when the programmer visible PROD/CONS pointer is updated. More can be visible in memory than are advertised in the programmer-visible PROD pointer. This also traces the latest PROD that could have been advertised. Fields:

mem_prod unsigned int

The PROD pointer value for all records that are visible in the memory.

num_events_available unsigned int

The number of events available to the programmer.

prog_cons unsigned int

The programmer-visible CONS pointer.

prog_prod unsigned int

The programmer-visible PROD pointer.

what enum

What changed?.

ns_eventq_raising_eventq_abt_err

EVENTQ_ABT_ERR can now be raised.

ns_eventq_removing_from_cmd_sync_prior_set

The event captured by a CMD_SYNC and created by the specified transaction id is now in memory. Fields:

prod unsigned int

The prod index (no wrap bit) that must become visible to the programmer to see this record. If not head_of_line then the record will not become visible and this field is meaningless.

reason enum

The reason the event was removed.

trans_id unsigned int

The transaction id that caused the event.

ns_eventq_stashing_unreported_stalled_transaction_for_auto_retry

A stall event record would have been lost, instead it is stashed away to auto-retry when it can. Fields:

dword0 unsigned int

Double word 0 of event.

dword1 unsigned int

Double word 1 of event.

dword2 unsigned int

Double word 2 of event.

dword3 unsigned int

Double word 3 of event.

event enum

Event code that was thrown away.

reason enum

The reason the record was lost.

trans_id unsigned int

The transaction id that this corresponds to.

ns_eventq_write_event

An event is committed to being written. Fields:

dword0 unsigned int

Double word 0 of event.

dword1 unsigned int

Double word 1 of event.

dword2 unsigned int

Double word 2 of event.

dword3 unsigned int

Double word 3 of event.

event enum

Event code that was thrown away.

prod unsigned int

The PROD pointer (without wrap) that this event will be written to.

stall_event bool

Describes an event for a stalled transaction.

trans_id unsigned int

Transaction ID that generated this event.

ns_eventq_write_event_visible_in_memory

An event is now visible in memory. Fields:

abort bool

The write aborted.

dword0 unsigned int

Double word 0 of event.

dword1 unsigned int

Double word 1 of event.

dword2 unsigned int

Double word 2 of event.

dword3 unsigned int

Double word 3 of event.

event enum

Event code that was thrown away.

prod unsigned int

The PROD pointer (without wrap) that this event will be written to.

stall_event bool

Describes an event for a stalled transaction.

trans_id unsigned int

Transaction ID that generated this event.

ns_gerror_inverted

A GERROR bit was attempted to be inverted to record a fault. Fields:

already_different_to_gerrorn bool

True if the bit was already different to the corresponding bit in GERRORN and so it was not inverted.

gerrorn unsigned int

GERRORN register value.

interrupt_potentially_sent bool

An interrupt is potentially sent, depending on IRQEN.
GERROR(N).MSI_GERROR_ABT_ERR does not send an interrupt.

new_gerror unsigned int

GERROR register value after any inversion.

which enum

Which bit was attempted to be inverted.

ns_gerrorn_acknowledge

A GERROR was acknowledged by SW. Fields:

which enum

Which GERROR was acknowledged.

pmcg_irq_config

The interrupt configuration of the Performance Monitor Counter Group (PMCG) changed.
Fields:

address unsigned int

Address of the MSI.

data unsigned int

Data payload of the MSI.

irqen bool

SMMU_PMCG_CTRL.IRQEN.

memattr enum

Memory type.

mpam_ns bool

The NS state of the MPAM PARTID and MPAM PMG.

mpam_pmg_and_partid unsigned int

Bits[23:16] are the MPAM PMG. Bits[15:0] are the physical PARTID.

msi_supported bool

Are MSIs supported for this security world.

ns bool

Non-secure bus attribute.

number_of_interrupts_in_flight unsigned int

The number of interrupts that have been committed to be produced or in flight.

pmcg_index unsigned int

Index of the PMCG.

sh enum

Shareability.

smmu_pmcg_gmpam_Update bool

The SMMU_PMCG_GMPAM.Update flag. Only when this is zero are writes predictable.

ssd_ns bool

True if this PMCG controlled by the non-secure world.

pmcg_irqen

A trace of SMMU_PMCG_IRQ_CTRL.IRQEN. Fields:

new_value bool

The new value of IRQEN.

pmcg_index unsigned int

Index of the PMCG.

ssd_ns bool

True if this PMCG controlled by the non-secure world.

pmcg_irqen_ack

A trace of SMMU_PMCG_IRQ_CTRLACK.IRQEN. Fields:

new_value bool

The new value of IRQEN.

pmcg_index unsigned int

Index of the PMCG.

ssd_ns bool

True if this PMCG controlled by the non-secure world.

pmcg_merging_interrupts

An interrupt was wanted to be generated, but one was already pending so the two were merged together. Fields:

pmcg_index unsigned int

Index of the PMCG.

ssd_ns bool

True if this PMCG controlled by the non-secure world.

pmcg_trigger

A PMCG counter has been triggered. Fields:

NoStreamID bool

True if the transaction is a NoStreamID transaction.

counter_index unsigned int

The index of the counter within the PMCG.

event_id unsigned int

The event id that has been triggered.

ns_event bool

Is the event associated with non-secure state.

pmcg_index unsigned int

Index of the PMCG.

prior_counter_value unsigned int

The Counter value *before* the event has incremented it.

ssd enum

The security state associated with the event.

streamid unsigned int

The StreamID associated with the event, if there is one.

tbu_index unsigned int

The TBU index of the transaction, or ~0u if not applicable.

pmu_active_counter

Traces what active counters are in a PMCG and what StreamIDs it might filter on. Those counters that trace StreamIDs for multiple security states, or those that are not filtered by StreamID, will appear multiple times, once for each security state. All active counters for a PMCG are traced one after another. Fields:

NoStreamID bool

True if NoStreamID transactions will be traced.

begin_streamid unsigned int

The start StreamID to filter on.

counter_index unsigned int

The counter index within the PMCG.

end_incl_streamid unsigned int

The end inclusive StreamID to filter on.

evcnt unsigned int

The current count.

event_id unsigned int

The event ID to filter.

ns bool

Are the StreamIDs non-secure?.

pmcg_index unsigned int

The index of the PMCG that this counter belongs to.

ssd enum

SSD of the StreamID.

tbu_index_to_match unsigned int

The TBU index that must match, or ~0u if no matching applicable.

pmu_all_counters_in_pmcg_became_inactive

The PMCG was tracing some events and now is not tracing any. Fields:

pmcg_index unsigned int

The index of the PMCG.

pmu_capture

For some reason, a capture event occurred. Fields:

pmcg_index unsigned int

The index of the PMCG that the capture occurred on.

why enum

Why did the capture occur?.

pmu_counter_configured_to_use_unsupported_event

An enabled counter was configured to use a unsupported event. Fields:

counter_index unsigned int

The counter index within the PMCG.

event_id unsigned int

The unsupported event id.

pmcg_index unsigned int

The index of the PMCG that this counter belongs to.

pmu_counter_overflowed

A counter in a particular PMCG overflowed. Fields:

already_overflowed bool

True if the overflow flag was already set.

capture bool

True if it captured the other counter values.

counter_index unsigned int

The counter index within the PMCG.

interrupt bool

True if going to attempt to generate an interrupt.

interrupt_action enum

The interrupt action that is going to occur.

pmcg_index unsigned int

The index of the PMCG that this counter belongs to.

ssd_ns bool

The PMCG is controlled by the Non-secure security state.

priq_auto_response

A PRIQ auto response is generated. Fields:

prgindex unsigned int

The PRG Index of the response.

resp enum

The response to send.

streamid unsigned int

The StreamID of the response.

substreamid unsigned int

The PASID/SubstreamID attached to the response, or ~0u if there is none.

trans_id unsigned int

The ID of the PPR that is being lost.

why enum

Why the PPR is being lost.

priq_lost_ppr

The PRIQ is not reporting a PPR into the queue. It may or may not generate an auto-response. Fields:

is_stop_marker bool

The PPR being lost is a Stop Marker.

resp enum

What, if any, auto-response will be.

trans_id unsigned int

The ID of the PPR that is being lost.

why enum

Why the PPR is being lost.

priq_overflow_acking

Indicates that an overflow condition was acknowledged by writing to:- SMMU_PRIQ_CONS.OVACKFLG. Fields:

new_ovackflg bool

The new value of the SMMU_PRIQ_CONS.OVACKFLG.

priq_overflow_asserting

Indicates that we are toggling the SMMU_PRIQ_PROD.OVFLG because we lost a PRI request due to the PRIQ being full and an existing overflow condition does not already exist. Fields:

new_ovflg bool

The new value of the SMMU_PRIQ_PROD.OVFLG.

trans_id unsigned int

The transaction ID of the PPR that caused the overflow.

priq_received

A PRIQ Request has been received and is queued waiting processing. This does not mean that it has been written to the PRIQ but has been placed in a pending queue awaiting a decision about what to do about it. Fields:

L bool

The Last in the Page Request Group (PRG).

R bool

Read permission is requested.

W bool

Read permission is requested.

prgindex unsigned int

The Page Request Group (PRG) index that is being used to identify this request to the EndPoint.

stop_marker bool

If LRW == 0b100 then this indicates this is a stop marker.

streamid unsigned int

The StreamID this request is associated with.

substreamid unsigned int

The SubstreamID (PASID) this request is associated with, of ~0u if no SubstreamID.

substreamid_P bool

If has a SubstreamID (!= ~0u) then Privileged Mode Requested.

substreamid_X bool

If has a SubstreamID (!= ~0u) then Execution Requested (R must also be 1).

trans_id unsigned int

The transaction ID that identifies this request.

untranslated_address unsigned int

The Untranslated Address that this is requesting.

priq_state

The trace of various fields that indicate the state of the PRIQ. Fields:

cons_incl_wrap unsigned int

The value of SMMU_PRIQ.CONS.RD_and_RD_wrap.

number_of_pprs unsigned int

The number of PPRs as indicated by the CONS/PROD.

number_of_pprs_still_to_deal_with unsigned int

This is the number of PPRs that are currently waiting to either be written to the PRIQ, or auto-responded to.

number_of_priq_writes_in_flight unsigned int

The number of writes to the PRIQ that are currently in flight.

ovackflg bool

The OVACKFLG which if different to OVFLG is used to indicate that the PRIQ overflowed.

ovflg bool

The OVFLG which if different to OVACKFLG is used to indicate that the PRIQ overflowed.

priq_abt_err bool

There is an active SMMU_GERROR{N}.PRIQ_ABT_ERR.

priqen bool

The value of SMMU_CRO.PRIQEN. The *effective* value is 0 if SMMUEN == 0.

priqenack bool

The value of SMMU_CROACK.PRIQEN.

prod_incl_wrap unsigned int

The value of SMMU_PRIQ.CONS.WR_and_WR_wrap.

queue_disabled_due_to_prior_programming_error bool

The queue was disabled as the programmer got CONS/PROD into an inconsistent state. The model will disable the PRIQ until SW disables and re-enables the queue via SMMU_CR0.PRIQEN.

smmuen bool

The value of SMMU_CR0.SMMUEN. If this is 0 then the effective PRIQEN is 0.

smmuenack bool

The value of SMMU_CR0.SMMUEN. If this is 0 then the effective PRIQEN is 0.

table_size_in_elements unsigned int

The size of the table in the number of items it can hold.

priq_write_aborted

A PRIQ write aborted. The PRIQ now goes into an error state and will start auto-responding to PRI requests. Fields:

prod_incl_wrap unsigned int

The PROD including the Wrap bit where we were trying to write.

trans_id unsigned int

The transaction ID of the PPR that aborted.

priq_write_ok

A PRIQ write completed OK. Fields:

prod_incl_wrap unsigned int

The PROD including the Wrap bit where we wrote..

trans_id unsigned int

The transaction ID of the PPR.

priq_write_start

A PRIQ request has been received and is going to be attempt to be written to the queue. Fields:

L bool

The Last in the Page Request Group (PRG).

R bool

Read permission is requested.

W bool

Read permission is requested.

prgindex unsigned int

The Page Request Group (PRG) index that is being used to identify this request to the EndPoint.

prod_incl_wrap unsigned int

PROD position including the wrap bit.

stop_marker bool

If LRW == 0b100 then this indicates this is a stop marker.

streamid unsigned int

The StreamID this request is associated with.

substreamid unsigned int

The SubstreamID (PASID) this request is associated with, of ~0u if no SubstreamID.

substreamid_P bool

If has a SubstreamID (!= ~0u) then Privileged Mode Requested.

substreamid_X bool

If has a SubstreamID (!= ~0u) then Execution Requested (R must also be 1).

trans_id unsigned int

The transaction ID that identifies this request.

untranslated_address unsigned int

The Untranslated Address that this is requesting.

ptw_read

Page Table Walk (read). This is the result of the physical access that the SMMU is making.

Fields:

abort enum

Non-zero if the access aborted/failed.

adomain enum

The actual attributes of the access that was used, after IMP DEF mangling.

aprot enum

The actual attributes of the access that was used, after IMP DEF mangling. PAS[1] (bit[3]), Instruction/Data (bit[2]), PAS[0] (bit[1]), Privileged/User (bit[0]).

data unsigned int

The data fetch if it didn't abort.

desckind enum

The kind of descriptor.

inner_cache enum

The actual attributes of the access that was used, after IMP DEF mangling.

input_address unsigned int

Input address this lookup is for (after TBI normalisation).

mecid unsigned int

The MECID of the TTD fetch.

ns enum

Non-Secure on the bus.

outer_cache enum

The actual attributes of the access that was used, after IMP DEF mangling.

pa_address unsigned int

Physical address.

pas enum

The PAS of the TTD fetch.

ssd enum

SSD of the transaction.

ssd_ns enum

Security state of stream.

streamid unsigned int

The StreamID this is for.

substreamid unsigned int

The SubstreamID this is for.

trans_id unsigned int

Transaction id.

ttb_grain_stage_and_level unsigned int

Which TTB used in bits[31:24]. Which Grain size used in bit[23:16]:- 0 – 4 KiB 1 – 16 KiB 2 – 64 KiB 3 – 64 KiB ARMv8.2 where descriptor bits [15:12] for bits [51:48] of output address 4 – 4 KiB FEAT_LPA2 where address is {bits[9:8], bits[49:12]} 5 – 16 KiB FEAT_LPA2 where address is {bits[9:8], bits[49:14], 2'b00} 6 – 64 KiB ARMv8.2 where descriptor bits [15:12] are ignored Stage in bits[15:8]. Level in bits[7:0], level -1 is 0xF.

ptw_read_st1_invalid_long_descriptor

Page Table Walk (read). The descriptor that the PTW fetched is invalid. Fields:

desckind enum

The kind of descriptor.

input_address unsigned int

Input address this lookup is for (after TBI normalisation).

mecid unsigned int

The MECID of the TTD fetch.

ns enum

Non-Secure on the bus.

pa_address unsigned int

Physical address.

pas enum

The PAS of the TTD fetch.

ssd enum

SSD of the transaction.

ssd_ns enum

Security state of stream.

streamid unsigned int

The StreamID this is for.

substreamid unsigned int

The SubstreamID this is for.

trans_id unsigned int

Transaction id.

ttb_grain_stage_and_level unsigned int

Which TTB used in bits[31:24]. Which Grain size used in bit[23:16]:- 0 – 4 KiB 1 – 16 KiB 2 – 64 KiB 3 – 64 KiB ARMv8.2 where descriptor bits [15:12] for bits [51:48] of output address 4 – 4 KiB FEAT_LPA2 where address is {bits[9:8], bits[49:12]} 5 – 16 KiB FEAT_LPA2 where address is {bits[9:8], bits[49:14], 2'b00} 6 – 64 KiB ARMv8.2 where descriptor bits [15:12] are ignored Stage in bits[15:8]. Level in bits[7:0], level -1 is 0xF.

ptw_read_st1_leaf_long_descriptor

Page Table Walk (read). The descriptor that the PTW fetched is a block or page and this is the decode. Fields:

AF bool

Access Flag.

AMEC enum

Use Primary or Alternative MECID.

AP21 enum

The access permissions.

AttrIndx210 unsigned int

The attribute index into the MAIR0/1. If AIE is implemented then this is the full index AttrIndx[3:0].

DBM bool

Dirty Bit Modifier. This only has meaning if running in AArch64 mode.

NS bool

The encoding is for non-secure if this is a secure fetch.

PIE_Dirty enum

The Dirty bit if S2PIE is in use.

PIE_PIIIndex unsigned int

The S2PIIndex if S2PIE is in use, or 0xFFFF if not.

PXN bool

Privileged eXecute Never.

Protected enum

Is the descriptor producing an AssuredTranslation.

SH10 enum

The shareability.

XN bool

eXecute Never.

contiguous bool

Contiguous hint.

desckind enum

The kind of descriptor.

hwu_pbha unsigned int

Top four bits are appropriate CD.HWU, *bottom bits[62:59] of descriptor. Page Based Hardware Attributes: only valid on a bit-per-bit basis enabled by CD.HWU0/CD.HWU1**.

input_address unsigned int

Input address this lookup is for (after TBI normalisation).

mecid unsigned int

The MECID of the TTD fetch.

nG bool

not Global.

nT bool

SMMUv3.2: If true then do not cache this entry in such a way that it can cause a TLB conflict abort (F_TLB_CONFLICT). The entry must produce a consistent result.

ns enum

Non-Secure on the bus.

output_address unsigned int

Output address.

pa_address unsigned int

Physical address.

pas enum

The PAS of the TTD fetch.

ssd enum

SSD of the transaction.

ssd_ns enum

Security state of stream.

streamid unsigned int

The StreamID this is for.

substreamid unsigned int

The SubstreamID this is for.

trans_id unsigned int

Transaction id.

ttb_grain_stage_and_level unsigned int

Which TTB used in bits[31:24]. Which Grain size used in bit[23:16]:- 0 – 4 KiB 1 – 16 KiB 2 – 64 KiB 3 – 64 KiB ARMv8.2 where descriptor bits [15:12] for bits [51:48] of output address 4 – 4 KiB FEAT_LPA2 where address is {bits[9:8], bits[49:12]} 5 – 16 KiB FEAT_LPA2 where address is {bits[9:8], bits[49:14], 2'b00} 6 – 64 KiB ARMv8.2 where descriptor bits [15:12] are ignored Stage in bits[15:8]. Level in bits[7:0], level -1 is 0xF.

ptw_read_st1_table_long_descriptor

Page Table Walk (read). The descriptor that the PTW fetched is a Table, this decodes the fields. Fields:

AF enum

.

APTable enum

Remove permissions independently of subsequent descriptors.

NSTable bool

The next level table descriptor is forced to non-secure.

PXNTable bool

Force PXN independently of subsequent descriptors.

Protected enum

Is the descriptor capable of producing an AssuredTranslation.

TableAddress unsigned int

Address of the next table.

XNTable bool

Force XN independently of subsequent descriptors.

desckind enum

The kind of descriptor.

input_address unsigned int

Input address this lookup is for (after TBI normalisation).

mecid unsigned int

The MECID of the TTD fetch.

ns enum

Non-Secure on the bus.

pa_address unsigned int

Physical address.

pas enum

The PAS of the TTD fetch.

ssd enum

SSD of the transaction.

ssd_ns enum

Security state of stream.

streamid unsigned int

The StreamID this is for.

substreamid unsigned int

The SubstreamID this is for.

trans_id unsigned int

Transaction id.

tth_grain_stage_and_level unsigned int

Which TTB used in bits[31:24]. Which Grain size used in bit[23:16]:- 0 – 4 KiB 1 – 16 KiB 2 – 64 KiB 3 – 64 KiB ARMv8.2 where descriptor bits [15:12] for bits [51:48] of output address 4 – 4 KiB FEAT_LPA2 where address is {bits[9:8], bits[49:12]} 5 – 16 KiB FEAT_LPA2 where address is {bits[9:8], bits[49:14], 2'b00} 6 – 64 KiB ARMv8.2 where descriptor bits [15:12] are ignored Stage in bits[15:8]. Level in bits[7:0], level -1 is 0xF.

ptw_read_st2_invalid_descriptor

Page Table Walk (read). The descriptor that the PTW fetched is invalid. Fields:

desckind enum

The kind of descriptor.

input_address unsigned int

Input address this lookup is for (after TBI normalisation).

mecid unsigned int

The MECID of the TTD fetch.

ns enum

Non-Secure on the bus.

pa_address unsigned int

Physical address.

pas enum

The PAS of the TTD fetch.

ssd enum

SSD of the transaction.

ssd_ns enum

Security state of stream.

streamid unsigned int

The StreamID this is for.

substreamid unsigned int

The SubstreamID this is for.

trans_id unsigned int

Transaction id.

ttb_grain_stage_and_level unsigned int

Which TTB used in bits[31:24]. Which Grain size used in bit[23:16]:- 0 – 4 KiB 1 – 16 KiB 2 – 64 KiB 3 – 64 KiB ARMv8.2 where descriptor bits [15:12] for bits [51:48] of output address 4 – 4 KiB FEAT_LPA2 where address is {bits[9:8], bits[49:12]} 5 – 16 KiB FEAT_LPA2 where address is {bits[9:8], bits[49:14], 2'b00} 6 – 64 KiB ARMv8.2 where descriptor bits [15:12] are ignored Stage in bits[15:8]. Level in bits[7:0], level -1 is 0xF.

ptw_read_st2_leaf_descriptor

Page Table Walk (read). The descriptor that the PTW fetched is a block or page and this is the decode. Fields:

AF bool

Access Flag.

AMEC enum

Use Primary or Alternative MECID.

AssuredOnly enum

The descriptor is marked as AssuredOnly.

DBM bool

Dirty Bit Modifier. This only has meaning if running in AArch64 mode.

HAP21 enum

The access permissions.

MemAttr3_0 enum

The memory attributes.

NS enum

Whether this descriptor forces NS.

PIE_Dirty enum

The Dirty bit if S2PIE is in use.

PIE_PIndex unsigned int

The S2PIndex if S2PIE is in use, or 0xFFFF if not.

POE_POIndex unsigned int

The S2POIndex if S2POE is in use, or 0xFFFF if not.

SH10 enum

The shareability.

XN enum

eXecute Never.

contiguous bool

Contiguous hint.

desckind enum

The kind of descriptor.

input_address unsigned int

Input address this lookup is for (after TBI normalisation).

mecid unsigned int

The MECID of the TTD fetch.

nT bool

SMMUv3.2: If true then do not cache this entry in such a way that it can cause a TLB conflict abort (F_TLB_CONFLICT). The entry must produce a consistent result.

ns enum

Non-Secure on the bus.

output_address unsigned int

Output address.

pa_address unsigned int

Physical address.

pas enum

The PAS of the TTD fetch.

s2hwi_pbha unsigned int

Top four bits are STE.S2HWU, bottom bits[62:59] of descriptor. *Page Based Hardware Attributes: only valid on a bit-per-bit basis enabled by STE.S2HWU.*

ssd enum

SSD of the transaction.

ssd_ns enum

Security state of stream.

streamid unsigned int

The StreamID this is for.

substreamid unsigned int

The SubstreamID this is for.

trans_id unsigned int

Transaction id.

ttb_grain_stage_and_level unsigned int

Which TTB used in bits[31:24]. Which Grain size used in bit[23:16]:- 0 – 4 KiB 1 – 16 KiB 2 – 64 KiB 3 – 64 KiB ARMv8.2 where descriptor bits [15:12] for bits [51:48] of output address 4 – 4 KiB FEAT_LPA2 where address is {bits[9:8], bits[49:12]} 5 – 16 KiB FEAT_LPA2 where address is {bits[9:8], bits[49:14], 2'b00} 6 – 64 KiB ARMv8.2 where descriptor bits [15:12] are ignored Stage in bits[15:8]. Level in bits[7:0], level -1 is 0xF.

ptw_read_st2_table_descriptor

Page Table Walk (read). The descriptor that the PTW fetched is a Table, this decodes the fields. Fields:

AF enum

.

APTable enum

Remove permissions independently of subsequent descriptors.

PXNTable bool

Force PXN independently of subsequent descriptors.

TableAddress unsigned int

Address of the next table.

XNTable bool

Force XN independently of subsequent descriptors.

desckind enum

The kind of descriptor.

input_address unsigned int

Input address this lookup is for (after TBI normalisation).

mecid unsigned int

The MECID of the TTD fetch.

ns enum

Non-Secure on the bus.

pa_address unsigned int

Physical address.

pas enum

The PAS of the TTD fetch.

ssd enum

SSD of the transaction.

ssd_ns enum

Security state of stream.

streamid unsigned int

The StreamID this is for.

substreamid unsigned int

The SubstreamID this is for.

trans_id unsigned int

Transaction id.

tbtb_grain_stage_and_level unsigned int

Which TTB used in bits[31:24]. Which Grain size used in bit[23:16]:- 0 – 4 KiB 1 – 16 KiB 2 – 64 KiB 3 – 64 KiB ARMv8.2 where descriptor bits [15:12] for bits [51:48] of

output address 4 – 4 KiB FEAT_LPA2 where address is {bits[9:8], bits[49:12]} 5 – 16 KiB FEAT_LPA2 where address is {bits[9:8], bits[49:14], 2'b00} 6 – 64 KiB ARMv8.2 where descriptor bits [15:12] are ignored Stage in bits[15:8]. Level in bits[7:0], level -1 is 0xF.

raw_register_end_read

The raw register read transaction. This is the transaction as directed to the register port.

Fields:

address unsigned int

The address of the transaction.

id unsigned int

Id of this transaction.

ok bool

Was the access OK? The bus response will always be OK, but is the register access reasonable.

pas enum

The PAS of the transaction.

read_data unsigned int

The data read.

raw_register_end_write

The raw register write transaction. This is the transaction as directed to the register port.

Fields:

address unsigned int

The address of the transaction.

id unsigned int

Id of this transaction.

ok bool

Was the access OK? The bus response will always be OK, but is the register access reasonable.

pas enum

The PAS of the transaction.

raw_register_start_read

The raw register read transaction. This is the transaction as directed to the register port.

Fields:

address unsigned int

Address of transaction.

id unsigned int

Id of this transaction. Top bit is set if is a debug transaction.

ns enum

Is the transaction non-secure?.

number_of_beats unsigned int

Number of beats.

pas enum

The PAS of the transaction.

size_of_beat_in_bytes unsigned int

Size of each beat in bytes.

raw_register_start_write

The raw register write transaction. This is the transaction as directed to the register port.

Fields:

address unsigned int

Address of transaction.

id unsigned int

Id of this transaction. Top bit is set if is a debug transaction.

ns enum

Is the transaction non-secure?.

number_of_beats unsigned int

Number of beats.

pas enum

The PAS of the transaction.

size_of_beat_in_bytes unsigned int

Size of each beat in bytes.

write_data unsigned int

The data to write.

register_disallowed_read_string

A text representation of the read of a register that was disallowed. Fields:

out string

The text description of the register value read.

register_disallowed_write_string

A text representation of the write of a register write that was disallowed. Fields:

in string

The text description of the register value written.

register_read_reserved

A text representation of an access to a register address that is reserved. Fields:

in string

The text description of the register value.

register_read_string

A text representation of the read of a register. Fields:

out string

The text description of the register value read.

register_write_reserved

A text representation of an access to a register address that is reserved or a write to a **RES0** field in a register. Fields:

in string

The text description of the register value.

register_write_string

A text representation of the write of a register. Fields:

in string

The text description of the register value written.

rl_cmd_sync_completed_irq

“Realm” CMD_SYNC completed its IRQ action. Fields:

cmd_id unsigned int

The ID of the command being issued.

cons_incl_wrap unsigned int

CONS pointer of CMD_SYNC.

is_synthetic bool

This is a synthetic sync. The cons_incl_wrap is 0xFFFF'ffff.

rl_cmd_sync_completed_no_action

Realm CMD_SYNC completed, there was no CMD_SYNC action (SIG_NONE). Fields:

cmd_id unsigned int

The ID of the command being issued.

cons_incl_wrap unsigned int

CONS pointer of CMD_SYNC.

is_synthetic bool

This is a synthetic sync. The cons_incl_wrap is 0xFFFF'ffff.

rl_cmd_sync_completed_sev

Realm CMD_SYNC completed its action of SIG_SEV. Fields:

cmd_id unsigned int

The ID of the command being issued.

cons_incl_wrap unsigned int

CONS pointer of CMD_SYNC.

is_synthetic bool

This is a synthetic sync. The cons_incl_wrap is 0xFFFF'ffff.

rl_cmd_sync_completed_with_error

Realm CMD_SYNC completed, however, there was an error associated with the completion of the CMD_SYNC. Fields:

error enum

The error associated with the CMD_SYNC.

cmd_id unsigned int

The ID of the command being issued.

cons_incl_wrap unsigned int

CONS pointer of CMD_SYNC.

is_synthetic bool

This is a synthetic sync. The cons_incl_wrap is 0xFFFF'ffff.

rl_cmd_sync_issued

Realm CMD_SYNC has now completed issuing and we now know if something is hazarding the CMD_SYNC from completing. Fields:

cmd_id unsigned int

The ID of the command being issued.

cons_incl_wrap unsigned int

CONS pointer of CMD_SYNC.

hazarded bool

The CMD_SYNC is currently hazarded from completing.

is_synthetic bool

This is a synthetic sync. The cons_incl_wrap is 0xFFFF'ffff.

rl_cmd_sync_issuing

Realm CMD_SYNC is starting to issue, we are going to start accounting for which things have to be part of the prior set of things that have to complete before the CMD_SYNC can complete. Fields:

cmd_id unsigned int

The ID of the command being issued.

cons_incl_wrap unsigned int

CONS pointer of CMD_SYNC.

is_synthetic bool

This is a synthetic sync. The cons_incl_wrap is 0xFFFF'ffff.

rl_cmd_sync_starting_completion_action_irq

Realm CMD_SYNC starting IRQ completion action. Fields:

cmd_id unsigned int

The ID of the command being issued.

cons_incl_wrap unsigned int

CONS pointer of CMD_SYNC.

is_synthetic bool

This is a synthetic sync. The cons_incl_wrap is 0xFFFF'ffff.

msiaddress unsigned int

The MSI address to use (or 0 if not sending an MSI).

msidata unsigned int

The MSI data to send.

rl_cmd_sync_starting_completion_action_sev

Realm CMD_SYNC starting its completion action of SIG_SEV. Fields:

cmd_id unsigned int

The ID of the command being issued.

cons_incl_wrap unsigned int

CONS pointer of CMD_SYNC.

is_synthetic bool

This is a synthetic sync. The cons_incl_wrap is 0xFFFF'ffff.

rl_cmdq_cmd_sync_error

The CMD_SYNC can fault if an ATC Invalidate timeout occurs. The timeout will be attached to the next CMD_SYNC on the queue on which the invalidate was launched. Fields:

cerror enum

The CERROR to attach.

cmd_queue unsigned int

The top bits indicate the CMDQ id. These bits are attached to all cmd_ids generated for this queue. The bits [63:44] contain the queue ID. [63:62] 0: s-ssd, 1: reserved, 2: ns-ssd. 3: rl-ssd [61] is ECMDQ [59:52] QCP index [51:44] ECMDQ index in QCP.

herror enum

The HERROR to attach, or HERROR_NONE if we are attaching a CERROR.

rl_cmdq_cmd_sync_error_delivered

The CMD_SYNC can fault if an ATC Invalidate timeout occurs. This is the CMD_SYNC delivering that fault. Fields:

cerror enum

The CERROR to attach.

cmd_id unsigned int

Command ID of the CMD_SYNC.

rl_cmdq_ecmdq_enack

Trace the changes to ENACK. Fields:

error enum

The error being exposed, if any.

cmd_queue unsigned int

The top bits indicate the CMDQ id. These bits are attached to all cmd_ids generated for this queue. The bits [63:44] contain the queue ID. [63:62] 0: s-ssd, 1: reserved, 2: ns-ssd. 3: rl-ssd [61] is ECMDQ [59:52] QCP index [51:44] ECMDQ index in QCP.

enack bool

The value of ENACK.

forced_error bool

The queue was enabled with ERR != ERRACK and so the implementation chooses to trigger the error reporting mechanism.

rl_cmdq_issue

A command is actually being executed. Fields:

cmd_id unsigned int

Command ID of the command being executed.

cons unsigned int

CONS of the command.

what string

What is being executed.

rl_cmdq_pointers_state

The command queue pointers. Fields:

cmd_queue unsigned int

The top bits indicate the CMDQ id. These bits are attached to all cmd_ids generated for this queue. The bits [63:44] contain the queue ID. [63:62] 0: s-ssd, 1: reserved, 2: ns-ssd. 3: rl-ssd [61] is ECMDQ [59:52] QCP index [51:44] ECMDQ index in QCP.

disable_fetch bool

The model has decided that as you programmed PROD/CONS inconsistently that it would not fetch as it has no confidence as to which are commands and which are not.

q_issuable_and_issued_cons unsigned int

The CONS pointer of the region that is issuable and/or issued (including wrap).

q_issuable_and_issued_prod unsigned int

The PROD pointer of the region that is issuable and/or issued (including wrap).

q_programmer_cons unsigned int

The CONS pointer as seen by the programmer (including wrap).

q_programmer_prod unsigned int

The PROD pointer as seen by the programmer (including wrap).

q_unfetched_cons unsigned int

The CONS pointer of the region unfetched (including wrap).

q_unfetched_prod unsigned int

The PROD pointer of the region unfetched (including wrap).

rl_cmdq_state

A description of what the CMDQ can do now. Fields:

can_cerror bool

There is a CERROR that can be recognised.

can_cmd_sync bool

There is a CMD_SYNC that can be recognised.

can_fetch bool

The SMMU can fetch.

can_herror bool

There is an HERROR that can be recognised.

can_issue bool

There are fetched commands that are eligible to be issued.

can_update_cons bool

The programmer view of CMDQ_CONS can be moved to say some have been consumed.

cmd_queue unsigned int

The top bits indicate the CMDQ id. These bits are attached to all cmd_ids generated for this queue. The bits [63:44] contain the queue ID. [63:62] 0: s-ssd, 1: reserved, 2: ns-ssd. 3: rl-ssd [61] is ECMDQ [59:52] QCP index [51:44] ECMDQ index in QCP.

cmd_sync enum

Why a CMD_SYNC cannot be issued.

cmdqen bool

CMDQEN, when enabled then the CMDQ can fetch commands.

cmdqen_1_to_0_unacknowledged bool

CMDQEN has been set to 0, but we have yet to acknowledge it.

current-real-cerror enum

The current CERROR that would be visible to the programmer if there was an error.

fetch-disabled_due_to_programmer_error bool

The PROD/CONS pointer was inconsistent and the model decided to stop processing commands.

fetch-number_of_unfetched_commands unsigned int

The number of commands that have yet to be fetched from memory, or ~0u if CMDQEN == 0.

fetch-state enum

The state of the fetch state machine.

fetch-too_many_commands_outstanding_so_not_fetching_any_more bool

If there are too many issued commands, or pending to be issued then we won't fetch any more.

issue-number_of_issuable_commands unsigned int

The number of commands that have been fetched but not yet issued.

issue-number_of_unissuable_commands unsigned int

This might include commands behind an illegal command, or behind a CMD_SYNC. Under some circumstances this can include a command representing an external abort.

issue-state enum

The internal issued state machine state.

outstanding_fetches_since_reset unsigned int

The number of outstanding fetches.

synthetic_sync_can_be_completed bool

The synthetic sync that is outstanding can be completed.

waiting_for_synthetic_sync bool

There is a synthetic sync that is outstanding.

rl_eventq_adding_to_cmd_sync_prior_set

The event created by the specified transaction id must become visible to the programmer before the CMD_SYNC can complete. Fields:

trans_id unsigned int

The transaction id that caused the event.

rl_eventq_auto_retry_stalled_transaction

A transaction that was stalled but not yet reported on the Event Queue was retried. The retry can happen if the transaction didn't write because it was de-duplicated behind a reported stalling transaction and that stalling transaction was CMD_RESUMEd, or if the queue was unwritable and the queue is now writable. Fields:

trans_id unsigned int

Transaction id of the auto-retried transaction.

why enum

Why the transaction retried.

rl_eventq_cmd_sync_unhazardous

The CMD_SYNC has been unhazardous as the prior set is empty and as much of it as required is visible in the programmers' view. Fields:

prod_incl_wrap unsigned int

The prod index including wrap bit that was required to become visible.

prog_prod_incl_wrap unsigned int

The current programmer-visible prod including wrap bit.

rl_eventq_eventqen_and_eventqenack

Trace the values of EVENTQEN and EVENTQENACK as they change. Fields:

about enum

Some more information about the queue state.

cons_incl_wrap unsigned int

The CONS pointer (including wrap bits).

eventqen bool

EVENTQEN.

eventqenack bool

EVENTQENACK.

prod_incl_wrap unsigned int

The PROD pointer (including wrap bits).

rl_eventq_external_abort

A particular eventq record got an external abort. The event will subsequently appear in the ns/s_eventq_lost_event_records trace. Fields:

prod unsigned int

The prod pointer (excluding wrap bit) of the event record that aborted.

trans_id unsigned int

The transaction id that produced the event record that aborted.

rl_eventq_lost_event_records

Event records that are lost and never record appear on this trace. Fields:

dword0 unsigned int

Double word 0 of event.

dword1 unsigned int

Double word 1 of event.

dword2 unsigned int

Double word 2 of event.

dword3 unsigned int

Double word 3 of event.

event enum

Event code that was thrown away.

prod unsigned int

The PROD index of the event if appropriate or ~0u otherwise.

reason enum

The reason the record was lost.

trans_id unsigned int

The transaction id that this corresponds to.

rl_eventq_overflow_acknowledged

When events are lost then they appear on the eventq_lost_event_records_* trace. This will activate the overflow flag in the corresponding SMMU_s_EVENTQ_PROD register. Software can then acknowledge that flag using the overflow acknowledge flag in SMMU_s_EVENTQ_CONS. Fields:

bad bool

If true then the programmer has attempted to acknowledge an overflow condition that didn't exist.

count_of_events_lost_due_to_overflow unsigned int

This is the number of events that were lost because the event queue was full. This does not include any lost because of an EVENTQ_ABT_ERR.

rl_eventq_pending_event_records

Event records that are pending to be dealt with by the eventq state machine. A specific event can be traced multiple times if it becomes part of the prior set of transactions of a CMD_SYNC that has to complete before the CMD_SYNC is allowed to be completed. Fields:

dword0 unsigned int

Double word 0 of event.

dword1 unsigned int

Double word 1 of event.

dword2 unsigned int

Double word 2 of event.

dword3 unsigned int

Double word 3 of event.

event enum

Event code.

is_prior_set bool

The transaction forms part of the prior set of a CMD_SYNC that have to become visible (or thrown away if necessary) before the CMD_SYNC can complete.

trans_id unsigned int

The transaction id that this corresponds to.

rl_eventq_prod_cons_updated

This records when the programmer visible PROD/CONS pointer is updated. More can be visible in memory than are advertised in the programmer-visible PROD pointer. This also traces the latest PROD that could have been advertised. Fields:

mem_prod unsigned int

The PROD pointer value for all records that are visible in the memory.

num_events_available unsigned int

The number of events available to the programmer.

prog_cons unsigned int

The programmer-visible CONS pointer.

prog_prod unsigned int

The programmer-visible PROD pointer.

what enum

What changed?.

rl_eventq_raising_eventq_abt_err

EVENTQ_ABT_ERR can now be raised.

rl_eventq_removing_from_cmd_sync_prior_set

The event captured by a CMD_SYNC and created by the specified transaction id is now in memory. Fields:

prod unsigned int

The prod index (no wrap bit) that must become visible to the programmer to see this record. If not head_of_line then the record will not become visible and this field is meaningless.

reason enum

The reason the event was removed.

trans_id unsigned int

The transaction id that caused the event.

rl_eventq_stashing_unreported_stalled_transaction_for_auto_retry

A stall event record would have been lost, instead it is stashed away to auto-retry when it can. Fields:

dword0 unsigned int

Double word 0 of event.

dword1 unsigned int

Double word 1 of event.

dword2 unsigned int

Double word 2 of event.

dword3 unsigned int

Double word 3 of event.

event enum

Event code that was thrown away.

reason enum

The reason the record was lost.

trans_id unsigned int

The transaction id that this corresponds to.

rl_eventq_write_event

An event is committed to being written. Fields:

dword0 unsigned int

Double word 0 of event.

dword1 unsigned int

Double word 1 of event.

dword2 unsigned int

Double word 2 of event.

dword3 unsigned int

Double word 3 of event.

event enum

Event code that was thrown away.

prod unsigned int

The PROD pointer (without wrap) that this event will be written to.

stall_event bool

Describes an event for a stalled transaction.

trans_id unsigned int

Transaction ID that generated this event.

rl_eventq_write_event_visible_in_memory

An event is now visible in memory. Fields:

abort bool

The write aborted.

dword0 unsigned int

Double word 0 of event.

dword1 unsigned int

Double word 1 of event.

dword2 unsigned int

Double word 2 of event.

dword3 unsigned int

Double word 3 of event.

event enum

Event code that was thrown away.

prod unsigned int

The PROD pointer (without wrap) that this event will be written to.

stall_event bool

Describes an event for a stalled transaction.

trans_id unsigned int

Transaction ID that generated this event.

rl_gerror_inverted

A GERROR bit was attempted to be inverted to record a fault. Fields:

already_different_to_gerrorn bool

True if the bit was already different to the corresponding bit in GERRORN and so it was not inverted.

gerrorn unsigned int

GERRORN register value.

interrupt_potentially_sent bool

An interrupt is potentially sent, depending on IRQEN.
GERROR(N).MSI_GERROR_ABT_ERR does not send an interrupt.

new_gerror unsigned int

GERROR register value after any inversion.

which enum

Which bit was attempted to be inverted.

rl_gerrorn_acknowledge

A GERROR was acknowledged by SW. Fields:

which enum

Which GERROR was acknowledged.

rl_priq_auto_response

A PRIQ auto response is generated. Fields:

prgindex unsigned int

The PRG Index of the response.

resp enum

The response to send.

streamid unsigned int

The StreamID of the response.

substreamid unsigned int

The PASID/SubstreamID attached to the response, or ~0u if there is none.

trans_id unsigned int

The ID of the PPR that is being lost.

why enum

Why the PPR is being lost.

rl_priq_lost_ppr

The PRIQ is not reporting a PPR into the queue. It may or may not generate an auto-response. Fields:

is_stop_marker bool

The PPR being lost is a Stop Marker.

resp enum

What, if any, auto-response will be.

trans_id unsigned int

The ID of the PPR that is being lost.

why enum

Why the PPR is being lost.

rl_priq_overflow_acking

Indicates that an overflow condition was acknowledged by writing to:- SMMU_PRIQ_CONS.OVACKFLG. Fields:

new_ovackflg bool

The new value of the SMMU_PRIQ_CONS.OVACKFLG.

rl_priq_overflow_asserting

Indicates that we are toggling the SMMU_PRIQ_PROD.OVFLG because we lost a PRI request due to the PRIQ being full and an existing overflow condition does not already exist. Fields:

new_ovflg bool

The new value of the SMMU_PRIQ_PROD.OVFLG.

trans_id unsigned int

The transaction ID of the PPR that caused the overflow.

rl_priq_received

A PRIQ Request has been received and is queued waiting processing. This does not mean that it has been written to the PRIQ but has been placed in a pending queue awaiting a decision about what to do about it. Fields:

L bool

The Last in the Page Request Group (PRG).

R bool

Read permission is requested.

W bool

Read permission is requested.

prgindex unsigned int

The Page Request Group (PRG) index that is being used to identify this request to the EndPoint.

stop_marker bool

If LRW == 0b100 then this indicates this is a stop marker.

streamid unsigned int

The StreamID this request is associated with.

substreamid unsigned int

The SubstreamID (PASID) this request is associated with, of ~0u if no SubstreamID.

substreamid_P bool

If has a SubstreamID (!= ~0u) then Privileged Mode Requested.

substreamid_X bool

If has a SubstreamID (!= ~0u) then Execution Requested (R must also be 1).

trans_id unsigned int

The transaction ID that identifies this request.

untranslated_address unsigned int

The Untranslated Address that this is requesting.

rl_priq_state

The trace of various fields that indicate the state of the PRIQ. Fields:

cons_incl_wrap unsigned int

The value of SMMU_PRIQ.CONS.RD_and_RD_wrap.

number_of_pprs unsigned int

The number of PPRs as indicated by the CONS/PROD.

number_of_pprs_still_to_deal_with unsigned int

This is the number of PPRs that are currently waiting to either be written to the PRIQ, or auto-responded to.

number_of_priq_writes_in_flight unsigned int

The number of writes to the PRIQ that are currently in flight.

ovackflg bool

The OVACKFLG which if different to OVFLG is used to indicate that the PRIQ overflowed.

ovflg bool

The OVFLG which if different to OVACKFLG is used to indicate that the PRIQ overflowed.

priq_abt_err bool

There is an active SMMU_GERROR{N}.PRIQ_ABT_ERR.

priqen bool

The value of SMMU_CRO.PRIQEN. The *effective* value is 0 if SMMUEN == 0.

priqenack bool

The value of SMMU_CROACK.PRIQEN.

prod_incl_wrap unsigned int

The value of SMMU_PRIQ.CONS.WR_and_WR_wrap.

queue_disabled_due_to_prior_programming_error bool

The queue was disabled as the programmer got CONS/PROD into an inconsistent state. The model will disable the PRIQ until SW disables and re-enables the queue via SMMU_CR0.PRIQEN.

smmuen bool

The value of SMMU_CR0.SMMUEN. If this is 0 then the effective PRIQEN is 0.

smmuenack bool

The value of SMMU_CR0.SMMUEN. If this is 0 then the effective PRIQEN is 0.

table_size_in_elements unsigned int

The size of the table in the number of items it can hold.

rl_priq_write_aborted

A PRIQ write aborted. The PRIQ now goes into an error state and will start auto-responding to PRI requests. Fields:

prod_incl_wrap unsigned int

The PROD including the Wrap bit where we were trying to write.

trans_id unsigned int

The transaction ID of the PPR that aborted.

rl_priq_write_ok

A PRIQ write completed OK. Fields:

prod_incl_wrap unsigned int

The PROD including the Wrap bit where we wrote..

trans_id unsigned int

The transaction ID of the PPR.

rl_priq_write_start

A PRIQ request has been received and is going to be attempt to be written to the queue. Fields:

L bool

The Last in the Page Request Group (PRG).

R bool

Read permission is requested.

W bool

Read permission is requested.

prgindex unsigned int

The Page Request Group (PRG) index that is being used to identify this request to the EndPoint.

prod_incl_wrap unsigned int

PROD position including the wrap bit.

stop_marker bool

If LRW == 0b100 then this indicates this is a stop marker.

streamid unsigned int

The StreamID this request is associated with.

substreamid unsigned int

The SubstreamID (PASID) this request is associated with, of ~0u if no SubstreamID.

substreamid_P bool

If has a SubstreamID (!= ~0u) then Privileged Mode Requested.

substreamid_X bool

If has a SubstreamID (!= ~0u) then Execution Requested (R must also be 1).

trans_id unsigned int

The transaction ID that identifies this request.

untranslated_address unsigned int

The Untranslated Address that this is requesting.

s_cmd_sync_completed_irq

“Secure” CMD_SYNC completed its IRQ action. Fields:

cmd_id unsigned int

The ID of the command being issued.

cons_incl_wrap unsigned int

CONS pointer of CMD_SYNC.

is_synthetic bool

This is a synthetic sync. The cons_incl_wrap is 0xFFFF'ffff.

s_cmd_sync_completed_no_action

Secure CMD_SYNC completed, there was no CMD_SYNC action (SIG_NONE). Fields:

cmd_id unsigned int

The ID of the command being issued.

cons_incl_wrap unsigned int

CONS pointer of CMD_SYNC.

is_synthetic bool

This is a synthetic sync. The cons_incl_wrap is 0xFFFF'ffff.

s_cmd_sync_completed_sev

Secure CMD_SYNC completed its action of SIG_SEV. Fields:

cmd_id unsigned int

The ID of the command being issued.

cons_incl_wrap unsigned int

CONS pointer of CMD_SYNC.

is_synthetic bool

This is a synthetic sync. The cons_incl_wrap is 0xFFFF'ffff.

s_cmd_sync_completed_with_error

Secure CMD_SYNC completed, however, there was an error associated with the completion of the CMD_SYNC. Fields:

error enum

The error associated with the CMD_SYNC.

cmd_id unsigned int

The ID of the command being issued.

cons_incl_wrap unsigned int

CONS pointer of CMD_SYNC.

is_synthetic bool

This is a synthetic sync. The cons_incl_wrap is 0xFFFF'ffff.

s_cmd_sync_issued

Secure CMD_SYNC has now completed issuing and we now know if something is hazarding the CMD_SYNC from completing. Fields:

cmd_id unsigned int

The ID of the command being issued.

cons_incl_wrap unsigned int

CONS pointer of CMD_SYNC.

hazarded bool

The CMD_SYNC is currently hazarded from completing.

is_synthetic bool

This is a synthetic sync. The cons_incl_wrap is 0xFFFF'ffff.

s_cmd_sync_issuing

Secure CMD_SYNC is starting to issue, we are going to start accounting for which things have to be part of the prior set of things that have to complete before the CMD_SYNC can complete. Fields:

cmd_id unsigned int

The ID of the command being issued.

cons_incl_wrap unsigned int

CONS pointer of CMD_SYNC.

is_synthetic bool

This is a synthetic sync. The cons_incl_wrap is 0xFFFF'ffff.

s_cmd_sync_starting_completion_action_irq

Secure CMD_SYNC starting IRQ completion action. Fields:

cmd_id unsigned int

The ID of the command being issued.

cons_incl_wrap unsigned int

CONS pointer of CMD_SYNC.

is_synthetic bool

This is a synthetic sync. The cons_incl_wrap is 0xFFFF'ffff.

msiaddress unsigned int

The MSI address to use (or 0 if not sending an MSI).

msidata unsigned int

The MSI data to send.

s_cmd_sync_starting_completion_action_sev

Secure CMD_SYNC starting its completion action of SIG_SEV. Fields:

cmd_id unsigned int

The ID of the command being issued.

cons_incl_wrap unsigned int

CONS pointer of CMD_SYNC.

is_synthetic bool

This is a synthetic sync. The cons_incl_wrap is 0xFFFF'ffff.

s_cmdq_cmd_sync_error

The CMD_SYNC can fault if an ATC Invalidate timeout occurs. The timeout will be attached to the next CMD_SYNC on the queue on which the invalidate was launched. Fields:

cerror enum

The CERROR to attach.

cmd_queue unsigned int

The top bits indicate the CMDQ id. These bits are attached to all cmd_ids generated for this queue. The bits [63:44] contain the queue ID. [63:62] 0: s-ssd, 1: reserved, 2: ns-ssd. 3: rl-ssd [61] is ECMDQ [59:52] QCP index [51:44] ECMDQ index in QCP.

herror enum

The HERROR to attach, or HERROR_NONE if we are attaching a CERROR.

s_cmdq_cmd_sync_error_delivered

The CMD_SYNC can fault if an ATC Invalidate timeout occurs. This is the CMD_SYNC delivering that fault. Fields:

cerror enum

The CERROR to attach.

cmd_id unsigned int

Command ID of the CMD_SYNC.

s_cmdq_ecmdq_enack

Trace the changes to ENACK. Fields:

error enum

The error being exposed, if any.

cmd_queue unsigned int

The top bits indicate the CMDQ id. These bits are attached to all cmd_ids generated for this queue. The bits [63:44] contain the queue ID. [63:62] 0: s-ssd, 1: reserved, 2: ns-ssd. 3: rl-ssd [61] is ECMDQ [59:52] QCP index [51:44] ECMDQ index in QCP.

enack bool

The value of ENACK.

forced_error bool

The queue was enabled with ERR != ERRACK and so the implementation chooses to trigger the error reporting mechanism.

s_cmdq_issue

A command is actually being executed. Fields:

cmd_id unsigned int

Command ID of the command being executed.

cons unsigned int

CONS of the command.

what string

What is being executed.

s_cmdq_pointers_state

The command queue pointers. Fields:

cmd_queue unsigned int

The top bits indicate the CMDQ id. These bits are attached to all cmd_ids generated for this queue. The bits [63:44] contain the queue ID. [63:62] 0: s-ssd, 1: reserved, 2: ns-ssd. 3: rl-ssd [61] is ECMDQ [59:52] QCP index [51:44] ECMDQ index in QCP.

disable_fetch bool

The model has decided that as you programmed PROD/CONS inconsistently that it would not fetch as it has no confidence as to which are commands and which are not.

q_issuable_and_issued_cons unsigned int

The CONS pointer of the region that is issuable and/or issued (including wrap).

q_issuable_and_issued_prod unsigned int

The PROD pointer of the region that is issuable and/or issued (including wrap).

q_programmer_cons unsigned int

The CONS pointer as seen by the programmer (including wrap).

q_programmer_prod unsigned int

The PROD pointer as seen by the programmer (including wrap).

q_unfetched_cons unsigned int

The CONS pointer of the region unfetched (including wrap).

q_unfetched_prod unsigned int

The PROD pointer of the region unfetched (including wrap).

s_cmdq_state

A description of what the CMDQ can do now. Fields:

can_cerror bool

There is a CERROR that can be recognised.

can_cmd_sync bool

There is a CMD_SYNC that can be recognised.

can_fetch bool

The SMMU can fetch.

can_herror bool

There is an HERROR that can be recognised.

can_issue bool

There are fetched commands that are eligible to be issued.

can_update_cons bool

The programmer view of CMDQ_CONS can be moved to say some have been consumed.

cmd_queue unsigned int

The top bits indicate the CMDQ id. These bits are attached to all cmd_ids generated for this queue. The bits [63:44] contain the queue ID. [63:62] 0: s-ssd, 1: reserved, 2: ns-ssd. 3: rl-ssd [61] is ECMDQ [59:52] QCP index [51:44] ECMDQ index in QCP.

cmd_sync enum

Why a CMD_SYNC cannot be issued.

cmdqen bool

CMDQEN, when enabled then the CMDQ can fetch commands.

cmdqen_1_to_0_unacknowledged bool

CMDQEN has been set to 0, but we have yet to acknowledge it.

current-real-cerror enum

The current CERROR that would be visible to the programmer if there was an error.

fetch-disabled_due_to_programmer_error bool

The PROD/CONS pointer was inconsistent and the model decided to stop processing commands.

fetch-number_of_unfetched_commands unsigned int

The number of commands that have yet to be fetched from memory, or ~0u if CMDQEN == 0.

fetch-state enum

The state of the fetch state machine.

fetch-too_many_commands_outstanding_so_not_fetching_any_more bool

If there are too many issued commands, or pending to be issued then we won't fetch any more.

issue-number_of_issuable_commands unsigned int

The number of commands that have been fetched but not yet issued.

issue-number_of_unissuable_commands unsigned int

This might include commands behind an illegal command, or behind a CMD_SYNC. Under some circumstances this can include a command representing an external abort.

issue-state enum

The internal issued state machine state.

outstanding_fetches_since_reset unsigned int

The number of outstanding fetches.

synthetic_sync_can_be_completed bool

The synthetic sync that is outstanding can be completed.

waiting_for_synthetic_sync bool

There is a synthetic sync that is outstanding.

s_eventq_adding_to_cmd_sync_prior_set

The event created by the specified transaction id must become visible to the programmer before the CMD_SYNC can complete. Fields:

trans_id unsigned int

The transaction id that caused the event.

s_eventq_auto_retry_stalled_transaction

A transaction that was stalled but not yet reported on the Event Queue was retried. The retry can happen if the transaction didn't write because it was de-duplicated behind a reported stalling transaction and that stalling transaction was CMD_RESUMEd, or if the queue was unwritable and the queue is now writable. Fields:

trans_id unsigned int

Transaction id of the auto-retried transaction.

why enum

Why the transaction retried.

s_eventq_cmd_sync_unhazardous

The CMD_SYNC has been unhazardous as the prior set is empty and as much of it as required is visible in the programmers' view. Fields:

prod_incl_wrap unsigned int

The prod index including wrap bit that was required to become visible.

prog_prod_incl_wrap unsigned int

The current programmer-visible prod including wrap bit.

s_eventq_eventqen_and_eventqenack

Trace the values of EVENTQEN and EVENTQENACK as they change. Fields:

about enum

Some more information about the queue state.

cons_incl_wrap unsigned int

The CONS pointer (including wrap bits).

eventqen bool

EVENTQEN.

eventqenack bool

EVENTQENACK.

prod_incl_wrap unsigned int

The PROD pointer (including wrap bits).

s_eventq_external_abort

A particular eventq record got an external abort. The event will subsequently appear in the ns/s_eventq_lost_event_records trace. Fields:

prod unsigned int

The prod pointer (excluding wrap bit) of the event record that aborted.

trans_id unsigned int

The transaction id that produced the event record that aborted.

s_eventq_lost_event_records

Event records that are lost and never record appear on this trace. Fields:

dword0 unsigned int

Double word 0 of event.

dword1 unsigned int

Double word 1 of event.

dword2 unsigned int

Double word 2 of event.

dword3 unsigned int

Double word 3 of event.

event enum

Event code that was thrown away.

prod unsigned int

The PROD index of the event if appropriate or ~0u otherwise.

reason enum

The reason the record was lost.

trans_id unsigned int

The transaction id that this corresponds to.

s_eventq_overflow_acknowledged

When events are lost then they appear on the eventq_lost_event_records_* trace. This will activate the overflow flag in the corresponding SMMU_s_EVENTQ_PROD register. Software can then acknowledge that flag using the overflow acknowledge flag in SMMU_s_EVENTQ_CONS. Fields:

bad bool

If true then the programmer has attempted to acknowledge an overflow condition that didn't exist.

count_of_events_lost_due_to_overflow unsigned int

This is the number of events that were lost because the event queue was full. This does not include any lost because of an EVENTQ_ABT_ERR.

s_eventq_pending_event_records

Event records that are pending to be dealt with by the eventq state machine. A specific event can be traced multiple times if it becomes part of the prior set of transactions of a CMD_SYNC that has to complete before the CMD_SYNC is allowed to be completed. Fields:

dword0 unsigned int

Double word 0 of event.

dword1 unsigned int

Double word 1 of event.

dword2 unsigned int

Double word 2 of event.

dword3 unsigned int

Double word 3 of event.

event enum

Event code.

is_prior_set bool

The transaction forms part of the prior set of a CMD_SYNC that have to become visible (or thrown away if necessary) before the CMD_SYNC can complete.

trans_id unsigned int

The transaction id that this corresponds to.

s_eventq_prod_cons_updated

This records when the programmer visible PROD/CONS pointer is updated. More can be visible in memory than are advertised in the programmer-visible PROD pointer. This also traces the latest PROD that could have been advertised. Fields:

mem_prod unsigned int

The PROD pointer value for all records that are visible in the memory.

num_events_available unsigned int

The number of events available to the programmer.

prog_cons unsigned int

The programmer-visible CONS pointer.

prog_prod unsigned int

The programmer-visible PROD pointer.

what enum

What changed?.

s_eventq_raising_eventq_abt_err

EVENTQ_ABT_ERR can now be raised.

s_eventq_removing_from_cmd_sync_prior_set

The event captured by a CMD_SYNC and created by the specified transaction id is now in memory. Fields:

prod unsigned int

The prod index (no wrap bit) that must become visible to the programmer to see this record. If not head_of_line then the record will not become visible and this field is meaningless.

reason enum

The reason the event was removed.

trans_id unsigned int

The transaction id that caused the event.

s_eventq_stashing_unreported_stalled_transaction_for_auto_retry

A stall event record would have been lost, instead it is stashed away to auto-retry when it can. Fields:

dword0 unsigned int

Double word 0 of event.

dword1 unsigned int

Double word 1 of event.

dword2 unsigned int

Double word 2 of event.

dword3 unsigned int

Double word 3 of event.

event enum

Event code that was thrown away.

reason enum

The reason the record was lost.

trans_id unsigned int

The transaction id that this corresponds to.

s_eventq_write_event

An event is committed to being written. Fields:

dword0 unsigned int

Double word 0 of event.

dword1 unsigned int

Double word 1 of event.

dword2 unsigned int

Double word 2 of event.

dword3 unsigned int

Double word 3 of event.

event enum

Event code that was thrown away.

prod unsigned int

The PROD pointer (without wrap) that this event will be written to.

stall_event bool

Describes an event for a stalled transaction.

trans_id unsigned int

Transaction ID that generated this event.

s_eventq_write_event_visible_in_memory

An event is now visible in memory. Fields:

abort bool

The write aborted.

dword0 unsigned int

Double word 0 of event.

dword1 unsigned int

Double word 1 of event.

dword2 unsigned int

Double word 2 of event.

dword3 unsigned int

Double word 3 of event.

event enum

Event code that was thrown away.

prod unsigned int

The PROD pointer (without wrap) that this event will be written to.

stall_event bool

Describes an event for a stalled transaction.

trans_id unsigned int

Transaction ID that generated this event.

s_gbpa_abort_init

The pin is driven. This is the reset value of SMMU_S_GBPA.ABORT. This pin is sampled on the negedge of the reset pin and has to be driven beforehand for the value to take effect.

Fields:

value bool

The value of the signal.

s_gerror_inverted

A GERROR bit was attempted to be inverted to record a fault. Fields:

already_different_to_gerrorn bool

True if the bit was already different to the corresponding bit in GERRORN and so it was not inverted.

gerrorn unsigned int

GERRORN register value.

interrupt_potentially_sent bool

An interrupt is potentially sent, depending on IRQEN.
GERROR(N).MSI_GERROR_ABT_ERR does not send an interrupt.

new_gerror unsigned int

GERROR register value after any inversion.

which enum

Which bit was attempted to be inverted.

s_gerrorn_acknowledge

A GERROR was acknowledged by SW. Fields:

which enum

Which GERROR was acknowledged.

s_priq_auto_response

A PRIQ auto response is generated. Fields:

prgindex unsigned int

The PRG Index of the response.

resp enum

The response to send.

streamid unsigned int

The StreamID of the response.

substreamid unsigned int

The PASID/SubstreamID attached to the response, or ~0u if there is none.

trans_id unsigned int

The ID of the PPR that is being lost.

why enum

Why the PPR is being lost.

s_priq_lost_ppr

The PRIQ is not reporting a PPR into the queue. It may or may not generate an auto-response. Fields:

is_stop_marker bool

The PPR being lost is a Stop Marker.

resp enum

What, if any, auto-response will be.

trans_id unsigned int

The ID of the PPR that is being lost.

why enum

Why the PPR is being lost.

s_priq_received

A PRIQ Request has been received and is queued waiting processing. This does not mean that it has been written to the PRIQ but has been placed in a pending queue awaiting a decision about what to do about it. Fields:

L bool

The Last in the Page Request Group (PRG).

R bool

Read permission is requested.

W bool

Read permission is requested.

prgindex unsigned int

The Page Request Group (PRG) index that is being used to identify this request to the EndPoint.

stop_marker bool

If LRW == 0b100 then this indicates this is a stop marker.

streamid unsigned int

The StreamID this request is associated with.

substreamid unsigned int

The SubstreamID (PASID) this request is associated with, of ~0u if no SubstreamID.

substreamid_P bool

If has a SubstreamID (!= ~0u) then Privileged Mode Requested.

substreamid_X bool

If has a SubstreamID (!= ~0u) then Execution Requested (R must also be 1).

trans_id unsigned int

The transaction ID that identifies this request.

untranslated_address unsigned int

The Untranslated Address that this is requesting.

sec_override

Controls non-secure accesses to some registers. Fields:

value bool

The value of the signal.

sev

Send a SEV. Fields:

why enum

Why the SEV was generated.

smmu_atc_inv

The CMD_ATC_INV command is sent. Fields:

address unsigned int

Untranslated Address to invalidate.

cmd_id unsigned int

Command id.

cons unsigned int

CONS of the command.

global bool

Global flag.

log2_size_in_bytes unsigned int

Log 2 of the size in bytes of the region to invalidate.

ns_number_in_flight unsigned int

Number of ATC invalidates in flight issued from the Non-secure CMDQ.

ns_number_waiting_to_go_out unsigned int

The total number of ATC invalidates waiting to be pushed out from the Non-secure CMDQ.

rl_number_in_flight unsigned int

Number of ATC invalidates in flight issued from the Realm CMDQ.

rl_number_waiting_to_go_out unsigned int

The total number of ATC invalidates waiting to be pushed out from the Realm CMDQ.

s_number_in_flight unsigned int

Number of ATC invalidates in flight issued from the Secure CMDQ.

s_number_waiting_to_go_out unsigned int

The total number of ATC invalidates waiting to be pushed out from the Secure CMDQ.

ssd_of_queue enum

The SSD of the CMDQ.

ssd_of_streamid enum

The SSD of the StreamID.

streamid unsigned int

StreamID in the invalidate command.

substreamid unsigned int

SubstreamID or ~0u if SSValid == false.

smmu_atc_inv_completed

The CMDQ_ATC_INV command completed. Fields:

cmd_id unsigned int

Command id, if top-bit is set then was issued from the Non-secure CMDQ.

ns_number_in_flight unsigned int

Number of ATC invalidates in flight now this one has completed, that were issued by the Non-secure CMDQ.

ns_number_waiting_to_go_out unsigned int

The total number of ATC invalidates waiting to be pushed out from the Non-secure CMDQ.

response enum

The response.

rl_number_in_flight unsigned int

Number of ATC invalidates in flight now this one has completed, that were issued by the Realm CMDQ.

rl_number_waiting_to_go_out unsigned int

The total number of ATC invalidates waiting to be pushed out from the Realm CMDQ.

s_number_in_flight unsigned int

Number of ATC invalidates in flight now this one has completed, that were issued by the Secure CMDQ.

s_number_waiting_to_go_out unsigned int

The total number of ATC invalidates waiting to be pushed out from the Secure CMDQ.

ssd_of_cmdq enum

The SSD of the CMDQ.

ssd_of_streamid enum

The SSD of the StreamID.

smmu_atc_inv_end

An ATC invalidate request has been started. Fields:

PASID_global bool

If has a SubstreamID (PASID) then if it is 'global' or not.

address unsigned int

The Untranslated Address to use.

log2_size_in_bytes unsigned int

Log2 of the size of the region in bytes to invalidate.

response enum

The response to the ATC invalidate.

ssd enum

The SSD of the stream.

streamid unsigned int

The StreamID to use.

substreamid unsigned int

The SubstreamID to use (or ~0u if not used). This corresponds to the PASID.

smmu_atc_inv_start

An ATC invalidate request has been started. Fields:

PASID_global bool

If has a SubstreamID (PASID) then if it is 'global' or not.

address unsigned int

The Untranslated Address to use.

log2_size_in_bytes unsigned int

Log2 of the size of the region in bytes to invalidate.

ssd enum

The SSD of the stream.

streamid unsigned int

The StreamID to use.

substreamid unsigned int

The SubstreamID to use (or ~0u if not used). This corresponds to the PASID.

smmu_ats_initial

This is the initial ATS request. Fields:

XT bool

The XT bit.

ia unsigned int

Input address.

max_number_of_replies unsigned int

The maximum number of replies allowed to return.

no_write bool

The NW (no write flag) of the ATS request. If clear then the requester is going to do a write.

pasid_execute_requested bool

If there is a PASID (substreamid != ~0u) then this represents the requester asking for execution permissions.

pasid_privileged_mode_requested bool

If there is a PASID (substreamid != ~0u) then this represents the requester asking for the privileged mode's permissions.

protected_mode bool

Is the ATS Request protected-mode?.

source_cxl bool

Does the ATS request have Source.CXL set?.

ssd enum

The SSD of the ATS request.

ssd_ns bool

Incoming SSD is non-secure.

streamid unsigned int

StreamID of the ATS request.

substreamid unsigned int

SubstreamID (which is identical to the PASID) of the ATS request. If no PASID-prefix is sent then this is ~0u.

tbu unsigned int

Translation Buffer Unit number.

smmu_ats_reply_failure

This is an ATS reply indicating failure. Fields:

event enum

Equivalent event number that would have been generated for an equivalent ordinary transaction.

failure enum

What is the failure response code?.

state enum

The transaction state of the successfully ATS request.

smmu_ats_reply_success

This is an ATS reply, typically the SMMU will only return a single response, even if the requester indicated it could accept more replies. NOTE that the SMMU responds with 'success' in some cases when a fault is encountered and RW==0. Fields:

N bool

Non-snooped access. If one then the requester must clear the NoSnoop bit on transactions, unless otherwise enabled in a Function-specific manner.

P bool

Privileged mode. These permissions related to privileged mode.

RWX enum

Read/Write/Execute.

U bool

Untranslated access. If one, and RW !=0 then use UntranslatedAccesses for the allowed accesses by RW(X).

cxl_io bool

The CXL.io response.

inner enum

The inner cacheability attributes to use for TranslatedAccesses.

input_address unsigned int

Input address of the ATS request.

instcfg enum

The STE.INSTCFG field.

outer enum

The outer cacheability attributes to use for TranslatedAccesses.

pas enum

The PAS this mapping corresponds to. This holds the same information as the TE bit for realm streams.

privcfg enum

The STE.PRIVCFG field.

shareability unsigned int

The shareability to use for TranslatedAccesses.

size unsigned int

The size of the region covered by this translation.

state enum

The transaction state of the successfully ATS request.

translated_address unsigned int

If RW!=0 && U != 0, then the Translated Address that a TranslatedAccess can be made with.

smmu_axi_stream_msi

An SMMU generated MSI is directly sent through the axi_stream_msi_m port, typically connected to the GIC port axi_stream_msi_s. Fields:

TDEST unsigned int

Routing information for the data stream, typically identifying the GIC.

TID unsigned int

Data stream identifier for the SMMU.

axi_stream_msi_addr_to_match unsigned int

Current address to match for SMMU-originated MSIs to send out of the axi_stream_msi_m port.

data unsigned int

The MSI sent.

smmu_final_transaction

This is the transaction group request to remap has completed one set of remapping. For ordinary transactions, this represents a bundle of transactions with the same attributes but different addresses within a certain range around the address. If it stalls then it will report through this trace source, stall (stag_if_stalling != ~Ou) and when resume will issue another smmu_initial_transaction as it undergoes remapping again. This trace source can also represent part of the process for ATOS/ATS or finding STE.PPAR for PRI requests that need to be auto-responded to. Fields:

axmmuflow enum

The AxMMUFLOW for this transaction group. stallable – allow the transaction to be stalled if configured. translated-access – is a PCIe Translated Access non-stallable – do not stall transaction even if configured. transfault-flow – do not report for Translation Related Faults, but tell client device to use a page request mechanism instead (for example PRI).

begin_input_address_range unsigned int

The start of the input address range that is size_of_region_in_bytes.

begin_ipa_range unsigned int

The start of the IPA range that is of size_of_region_in_bytes.

begin_output_address_range unsigned int

The start of the output address range that is of size_of_region_in_bytes.

cmo_point enum

The point associated with the CMO, if applicable.

input_address unsigned int

The input address of the transaction group.

input_inner enum

Inner cacheability for the input attributes.

input_outer enum

Outer cacheability for the input attributes.

input_prot enum

The instruction/data, bus-ns/s and privileged/user marking.

input_sh enum

Shareability for the input attributes.

ipa_address unsigned int

The IPA of the transaction.

kind enum

The kind of access this transaction group that this represents.

mecid unsigned int

The MECID of the transaction.

mpam_pmg_and_partid unsigned int

Bits[23:16] are the MPAM PMG. Bits[15:0] are the physical PARTID.

mpam_sp enum

The MPAM_SP of the transaction.

operation enum

The kind of operation that this represents.

output_address unsigned int

The input address of the transaction group.

output_inner enum

Inner cacheability for the output attributes.

output_outer enum

Outer cacheability for the output attributes.

output_prot enum

The instruction/data, bus-ns/s and privileged/user marking.

output_sh enum

Shareability for the output attributes.

output_vmid unsigned int

The output VMID/GBPA.IMPDEF or ~0u if not valid.

size_of_region_in_bytes unsigned int

An imp def size of region for which this translation is valid for.

ssd enum

The SSD of the transaction.

ssd_ns enum

The security state of the transaction.

stag_if_stalling unsigned int

This is the STAG used by the transaction if it is going to stall. It is ~0u if it is not going to stall.

state enum

The final transaction state.

streamid unsigned int

The StreamID of the transaction. ~0ull if NoStreamID.

substreamid unsigned int

The SubstreamID of the transaction or 0xFFFFffff if no SubstreamID.

tbu unsigned int

Translation Buffer Unit number.

trans_id unsigned int

Transaction id. Top bit set if debug transaction.

smmu_gpt_read

A GPT read has completed. Fields:

unsigned int

.

abort enum

Non-zero if the access aborted/failed.

address_checking unsigned int

The address that we are checking.

address_gptdesc unsigned int

The address of the GPT descriptor.

adomain enum

The shareability of the access.

data unsigned int

The data fetched if it didn't abort.

inner_cache enum

The architectural attributes used for the access.

level unsigned int

The level of the GPT walk.

mpam_pmg_and_partid unsigned int

Bits[23:16] are the MPAM PMG. Bits[15:0] are the physical PARTID.

mpam_sp enum

The MPAM_SP of this GPT access.

outer_cache enum

The architectural attributes used for the access.

pas_checking enum

The PAS of the address that we are checking.

post_access_check bool

Is this GPT walk after the access it is checking.

table_base unsigned int

The base address of the table.

trans_id unsigned int

The trans_id of the transaction we are checking if appropriate.

what_checking enum

What kind of access are we checking.

smmu_gpt_tlb_invalidate

A GPT TLB invalidate has been initiated. Fields:

address unsigned int

The address as it appears in the operation.

encoded_size unsigned int

The size as it is encoded in the operation.

kind enum

The kind of operation this is.

pgs_in_bytes unsigned int

The PGS size in bytes.

size_in_bytes unsigned int

For range operations, the size as it appears in the operation.

source enum

Where the TLBI came from.

state enum

Is the operation well formed.

trans_id unsigned int

The transaction id of this invalidate.

smmu_gpt_tlb_invalidate_complete

The GPT TLB invalidate completed. Fields:

source enum

Where the TLBI came from.

trans_id unsigned int

The transaction id of this invalidate.

smmu_initial_transaction

This is the transaction group request to remap is going to start one set of remapping. For ordinary transactions, this represents a bundle of transactions with the same attributes but different addresses within a certain range around the address. A stalling transaction will report through this trace source when it unstalls. This trace source can also represent part of the process for ATOS/ATS or finding STE.PPAR for PRI requests that need to be auto-responded to. Fields:

XT_and_output_pas_checking enum

The XT bit for PCIe Transactions. This specifies the requested check on the output PAS that the device asked for.

axmmuflow enum

The AxMMUFLOW for this transaction group. storable – allow the transaction to be stalled if configured. translated-access – is a PCIe Translated Access non-storable – do not stall transaction even if configured. transfault-flow – do not report for Translation Related Faults, but tell client device to use a page request mechanism instead (for example PRI).

cmo_point enum

The point associated with the CMO, if applicable.

dcmdq_qcp_and_index unsigned int

If this is a DCMDQ fetch being translated then this field indicates the DCMDQ QCP index in bits [23:8] and the index in the page in bits [7:0]. Otherwise this is 0xFFFFffff.

input_address unsigned int

The input address of the transaction group.

input_inner enum

Inner cacheability for the input attributes.

input_outer enum

Outer cacheability for the input attributes.

input_prot enum

The instruction/data, bus-ns/s and privileged/user marking.

input_sh enum

Shareability for the input attributes.

kind enum

The kind of access this transaction group that this represents.

mecid unsigned int

The incoming MECID for NoStreamID transactions. ~0u for all other transactions.

mpam_partid unsigned int

The MPAM_PARTID for NoStreamID transactions. ~0u for all other transactions.

mpam_pmg unsigned int

The MPAM_PMG for NoStreamID transactions. ~0u for all other transactions.

mpam_sp enum

The MPAM_SP for NoStreamID transactions. ~0u for all other transactions.

operation enum

The kind of operation that this represents.

protected_mode enum

The PM bit.

ssd enum

The SSD of the transaction.

ssd_ns enum

The security state of the transaction.

streamid unsigned int

The StreamID of the transaction. ~0ull means NoStreamID.

substreamid unsigned int

The SubstreamID of the transaction or 0xFFFFffff if no SubstreamID.

tbu unsigned int

Translation Buffer Unit number.

trans_id unsigned int

Transaction id. Top bit set if debug transaction.

smmu_normalize_input_transaction

If the input transaction is normalized before being processed then this traceevent will fire. This is usually dependent on parameters of the implementation. Fields:

incoming_inner enum

The inner cacheability attributes.

incoming_is_instruction bool

The incoming transaction is marked as 'instruction'.

incoming_is_privileged bool

The incoming transaction is marked as 'privileged'.

incoming_outer enum

The outer cacheability attributes.

incoming_pas enum

The PAS of the incoming transaction.

incoming_shareability unsigned int

The incoming shareability.

normalized_inner enum

The normalized inner cacheability attributes.

normalized_is_instruction bool

The incoming transaction is marked as 'instruction'.

normalized_is_privileged bool

The incoming transaction is marked as 'privileged'.

normalized_outer enum

The normalized outer cacheability attributes.

normalized_pas enum

The PAS this mapping corresponds to.

normalized_shareability unsigned int

The normalized shareability.

ssd enum

The SSD of the transaction.

streamid unsigned int

The StreamID of the transaction. ~0ull means NoStreamID.

substreamid unsigned int

The SubstreamID of the transaction or 0xFFFFffff if no SubstreamID.

trans_id unsigned int

ID of the original transaction.

smmu_pmusnapshot_ack

Acknowledge the pmusnapshot_req to indicate the snapshot has occurred. This is part of a four-phase handshake. Fields:

pin_index unsigned int

The pin index of the signal, or ~0u if not relevant.

pmcg_index enum

The PMCG index.

value bool

The value of the signal.

smmu_pmusnapshot_ack_overridden

The value of pmusnapshot_ack was overridden (likely due to being in integration mode). Fields:

pin_index unsigned int

The pin index of the signal, or ~0u if not relevant.

pmcg_index enum

The PMCG index.

value bool

The value of the signal.

smmu_pmusnapshot_ack_override_end

The overriding of the pmusnapshot_ack signal has ended. Fields:

pin_index unsigned int

The pin index of the signal, or ~0u if not relevant.

pmcg_index enum

The PMCG index.

smmu_pmusnapshot_req

Take a snap shot of the PMU values as though SMMU_PMCG_CAPR.CAPTURE had been written. This is part of a four-phase handshake. Fields:

pin_index unsigned int

The pin index of the signal, or ~0u if not relevant.

pmcg_index enum

The PMCG index.

value bool

The value of the signal.

smmu_poison_tw_data

Poison data has been returned to a table walk transaction. Fields:

bitmap_of_poison unsigned int

The bitmap of which beats of the transaction where poisoned.

is_cas bool

True if this is a compare-and-swap operation.

number_of_64bit_beats unsigned int

Number of beats this transaction fetched.

paddress unsigned int

Physical address of the table walk transaction.

pas enum

The PAS of the bus transaction.

ras_group_id unsigned int

If non-~0u then is the RAS group that we wish to attach the error to. Otherwise will use the IMP DEF policy.

ras_record_index_in_group unsigned int

If non-~0u then is the RAS record index in the group that we wish to attach the error to. Otherwise will use the IMP DEF policy.

what enum

What table walk was being performed.

smmu_pri_resp

The CMD_PRI_RESP command is queued to be sent to the PCIe system. Fields:

auto_response_trans_id unsigned int

trans_id of PRI request we are auto-responding to, or ~0u if not valid.

cmd_id unsigned int

Command id, or ~0u if not valid.

cons unsigned int

CONS of the command. ~0u if an auto-response.

prgindex unsigned int

PRG Index.

response enum

If has a SubstreamID (PASID) then if it is 'global' or not.

ssd enum

The SSD of the StreamID this corresponds to.

streamid unsigned int

StreamID in the invalidate command.

substreamid unsigned int

SubstreamID or ~0u if SSValid == false.

smmu_pri_resp_nop

The CMD_PRI_RESP command was NOPped. Fields:

cmd_id unsigned int

Command id.

cons unsigned int

CONS of the command.

prgindex unsigned int

PRG Index.

response enum

If has a SubstreamID (PASID) then if it is 'global' or not.

ssd enum

The SSD of the StreamID this corresponds to.

streamid unsigned int

StreamID in the invalidate command.

substreamid unsigned int

SubstreamID or ~0u if SSValid == false.

why enum

Why the CMD_PRI_RESP was NOPped.

smmu_priq_resp_fake_return

A PRIQ Response is posted to the PCIe subsystem and so has no acknowledgement that it is received. However, in the model then we artificially know when the the PRIQ Response has been delivered to the PCIe subsystem, even if the ATC has not yet acted on it. Fields:

prgindex unsigned int

The PRG Index this request corresponds to.

ssd enum

SSD of response.

streamid unsigned int

The StreamID to use.

substreamid unsigned int

The SubstreamID to use (or ~0u if not used). This corresponds to the PASID.

what enum

What happened.

smmu_priq_resp_start

A PRIQ Response has been posted to the PCIe subsystem. As the response is posted then there is no way of knowing when it is received by the EndPoint. Fields:

prgindex unsigned int

The PRG Index this request corresponds to.

response enum

If has a SubstreamID (PASID) then if it is 'global' or not.

ssd enum

SSD of response.

streamid unsigned int

The StreamID to use.

substreamid unsigned int

The SubstreamID to use (or ~0u if not used). This corresponds to the PASID.

smmu_start_gpt_read

A GPT read is about to be issued. Fields:

address_checking unsigned int

The address that we are checking.

address_gptdesc unsigned int

The address of the GPT descriptor.

adomain enum

The shareability of the access.

inner_cache enum

The architectural attributes used for the access.

level unsigned int

The level of the GPT walk.

mpam_pmg_and_partid unsigned int

Bits[23:16] are the MPAM PMG. Bits[15:0] are the physical PARTID.

mpam_sp enum

The MPAM_SP of this GPT access.

outer_cache enum

The architectural attributes used for the access.

pas_checking enum

The PAS of the address that we are checking.

post_access_check bool

Is this GPT walk after the access it is checking.

read_size unsigned int

The size of the GPT read in bytes.

table_base unsigned int

The base address of the table.

trans_id unsigned int

The trans_id of the transaction we are checking if appropriate.

what_checking enum

What kind of access are we checking.

smmu_thread_wait_wake

Traces a thread's wait/wake status. Fields:

current_ticks unsigned int

The current tick count of simulated time.

event enum

What is happening to this thread.

thread_index unsigned int

The ID of this thread.

ticks unsigned int

If the event relates to a time then this is held in this field. Otherwise, 0.

stall_transaction

A transaction is about to stall. Fields:

stag unsigned int

STAG.

streamid unsigned int

StreamID.

streamid_ns bool

The StreamID was for the non-secure world.

trans_id unsigned int

Transaction id.

stall_transaction_inhibited_by_STALL_MAX

A transaction is about to stall but the maximum number of transactions have stalled and we can't report this one to the event queue (even if non-full). Fields:

streamid unsigned int

StreamID.

streamid_ns bool

The StreamID was for the non-secure world.

trans_id unsigned int

Transaction id.

stall_transaction_resuming

A stalled transaction is resuming. Fields:

stag unsigned int

STAG if appropriate, or if was inhibited by STALL_MAX then 0xFAFA.

stallresult enum

What the transaction resumed to do.

streamid unsigned int

StreamID.

streamid_ns bool

The StreamID was for the non-secure world.

trans_id unsigned int

Transaction id.

start_ptw_read

Page Table Walk (read). This is the start of the physical access that the SMMU is making. Fields:

adomain enum

The actual attributes of the access that was used, after IMP DEF mangling.

aprot enum

The actual attributes of the access that was used, after IMP DEF mangling. PAS[1] (bit[3]), Instruction/Data (bit[2]), PAS[0] (bit[1]), Privileged/User (bit[0]).

desckind enum

The kind of descriptor.

inner_cache enum

The actual attributes of the access that was used, after IMP DEF mangling.

input_address unsigned int

Input address this lookup is for (after TBI normalisation).

mecid unsigned int

The MECID of the TTD fetch.

mpam_pmg_and_partid unsigned int

Bits[23:16] are the MPAM PMG. Bits[15:0] are the physical PARTID.

ns enum

Non-Secure on the bus.

outer_cache enum

The actual attributes of the access that was used, after IMP DEF mangling.

pa_address unsigned int

Physical address.

pas enum

The PAS of the TTD fetch.

ssd enum

SSD of the transaction.

ssd_ns enum

Security state of stream.

streamid unsigned int

The StreamID this is for.

substreamid unsigned int

The SubstreamID this is for.

trans_id unsigned int

Transaction id.

ttb_grain_stage_and_level unsigned int

Which TTB used in bits[31:24]. Which Grain size used in bit[23:16]:- 0 – 4 KiB 1 – 16 KiB 2 – 64 KiB 3 – 64 KiB ARMv8.2 where descriptor bits [15:12] for bits [51:48] of output address 4 – 4 KiB FEAT_LPA2 where address is {bits[9:8], bits[49:12]} 5 – 16 KiB FEAT_LPA2 where address is {bits[9:8], bits[49:14], 2'b00} 6 – 64 KiB ARMv8.2 where descriptor bits [15:12] are ignored Stage in bits[15:8]. Level in bits[7:0], level -1 is 0xF.

ste_cc.STE_commentary

A commentary from the cache about what is going on. Fields:

output string

The stream output.

ste_cc.STE_entry_invalidated

An entry in the cache has been invalidated. Fields:

entry_id unsigned int

The entry id that is being invalidated.

reason enum

Why the entry is being invalidated.

ste_entry_allocated

An STE entry has been allocated. Fields:

entry_id unsigned int

The id of the cache entry. Top bit is set if is a debug entry.

ns enum

For the non-secure world.

ssd enum

The SSD of the entry.

ste string

A textual description of the STE.

streamid unsigned int

The StreamID of the entry this will match.

sup_btm

The pin is driven. This indicates the system supports BTM. This pin is sampled on the negedge of the reset pin and has to be driven beforehand for the value to take effect. Fields:

value bool

The value of the signal.

sup_cohacc

The pin is driven. This indicates the system supports COHACC. This pin is sampled on the negedge of the reset pin and has to be driven beforehand for the value to take effect. Fields:

value bool

The value of the signal.

sup_httu

The pin is driven. This indicates the system supports HTTU. This pin is sampled on the negedge of the reset pin and has to be driven beforehand for the value to take effect. Fields:

value bool

The value of the signal.

sup_sev

The pin is driven. This indicates the system supports SEV. This pin is sampled on the negedge of the reset pin and has to be driven beforehand for the value to take effect. Fields:

value bool

The value of the signal.

tbu0_reset_in

The reset signal of the TBU0. Fields:

value bool

The value of the signal.

tbu_fmurstdisable[0]

Port which, if asserted, prevents the reset from affecting the FMU registers. Fields:

value bool

The value of the signal.

tcu_fmurstdisable

Port which, if asserted, prevents the reset from affecting the FMU registers. Fields:

value bool

The value of the signal.

tcu_reset_in

The reset signal. Fields:

value bool

The value of the signal.

tlb_entry_allocated

A TLB entry has been allocated. Fields:

aset unsigned int

The ASET of this entry.

asid unsigned int

ASID if appropriate.

index unsigned int

Index of TLB entry.

input_end_incl_address unsigned int

The end inclusive address of the input range that this matches.

input_start_address unsigned int

The start address of the input range that this matches.

output_end_incl_address unsigned int

The end inclusive address of the output range.

output_start_address unsigned int

The start address of the output range.

scheme enum

The tagging scheme used.

ssd enum

The SSD of the tagging scheme.

ssd_ns enum

Security State.

tbi bool

Was the entry formed using Top Byte Ignore (TBI).

vmid unsigned int

VMID if appropriate.

tlb_info_tlb_entries_overlap

A TLB entry was inserted into the TLB and it overlaps an existing entry. This isn't a problem as it was inserted in such a way that it architecturally works. Fields:

end_address_of_new_entry unsigned int

End address of new entry.

end_address_of_old_entry unsigned int

End address of old entry.

how_inserted enum

How the entry was inserted.

index_of_new_entry unsigned int

Index of new entry.

index_of_old_entry unsigned int

Index of old entry.

start_address_of_new_entry unsigned int

Start address of new entry.

start_address_of_old_entry unsigned int

Start address of old entry.

verbose_commentary

This is a verbose commentary on the translation process the SMMU is performing. Fields:

output string

The stream output.

warning_MSI_EVENTQ_address_out_of_range_of_oas

The MSI address is configured outside of the range of the downstream address bus size.
Fields:

address unsigned int

The untruncated address of the MSI.

warning_MSI_GERROR_address_out_of_range_of_oas

The MSI address is configured outside of the range of the downstream address bus size.
Fields:

address unsigned int

The untruncated address of the MSI.

warning_MSI_HACDBS_PROCESSING_COMPLETE_address_out_of_range_of_oas

The MSI address is configured outside of the range of the downstream address bus size.
Fields:

address unsigned int

The untruncated address of the MSI.

warning_MSI_HDBSS_TABLE_FULL_address_out_of_range_of_oas

The MSI address is configured outside of the range of the downstream address bus size.
Fields:

address unsigned int

The untruncated address of the MSI.

warning_MSI_PRIQ_address_out_of_range_of_oas

The MSI address is configured outside of the range of the downstream address bus size.
Fields:

address unsigned int

The untruncated address of the MSI.

warning_MSI_R_EVENTQ_address_out_of_range_of_oas

The MSI address is configured outside of the range of the downstream address bus size.
Fields:

address unsigned int

The untruncated address of the MSI.

warning_MSI_R_GERROR_address_out_of_range_of_oas

The MSI address is configured outside of the range of the downstream address bus size.
Fields:

address unsigned int

The untruncated address of the MSI.

warning_MSI_R_HACDBS_PROCESSING_COMPLETE_address_out_of_range_of_oas

The MSI address is configured outside of the range of the downstream address bus size.
Fields:

address unsigned int

The untruncated address of the MSI.

warning_MSI_R_HDBSS_TABLE_FULL_address_out_of_range_of_oas

The MSI address is configured outside of the range of the downstream address bus size.
Fields:

address unsigned int

The untruncated address of the MSI.

warning_MSI_R_PRIQ_address_out_of_range_of_oas

The MSI address is configured outside of the range of the downstream address bus size.
Fields:

address unsigned int

The untruncated address of the MSI.

warning_MSI_S_EVENTQ_address_out_of_range_of_oas

The MSI address is configured outside of the range of the downstream address bus size.
Fields:

address unsigned int

The untruncated address of the MSI.

warning_MSI_S_GERROR_address_out_of_range_of_oas

The MSI address is configured outside of the range of the downstream address bus size.
Fields:

address unsigned int

The untruncated address of the MSI.

warning_MSI_S_HACDBS_PROCESSING_COMPLETE_address_out_of_range_of_oas

The MSI address is configured outside of the range of the downstream address bus size.
Fields:

address unsigned int

The untruncated address of the MSI.

warning_MSI_S_HDBSS_TABLE_FULL_address_out_of_range_of_oas

The MSI address is configured outside of the range of the downstream address bus size.
Fields:

address unsigned int

The untruncated address of the MSI.

warning_MSI_pmcg_address_out_of_range_of_oas

The MSI Address of the Performance Monitor Counter Group (PMCG) is out of range of the OAS and so will be silently truncated. Fields:

address unsigned int

The untruncated address of the MSI.

pmcg_index unsigned int

Index of the PMCG.

ssd_ns bool

True if this PMCG controlled by the non-secure world.

warning_discarding_interrupt_EVENTQ_as_irqen_low

Interrupt generation is turned off by SMMU_IRQ_CTRL.EVENTQ_IRQEN.

warning_discarding_interrupt_GERROR_as_irqen_low

Interrupt generation is turned off by SMMU_IRQ_CTRL.GERROR_IRQEN.

warning_discarding_interrupt_HACDBS_PROCESSING_COMPLETE_as_irqen_low

Interrupt generation is turned off by SMMU_IRQ_CTRL.HACDBS_IRQEN.

warning_discarding_interrupt_HDBSS_TABLE_FULL_as_irqen_low

Interrupt generation is turned off by SMMU_IRQ_CTRL.HDBSS_IRQEN.

warning_discarding_interrupt_PRIQ_as_irqen_low

Interrupt generation is turned off by SMMU_IRQ_CTRL.PRIQ_IRQEN.

warning_discarding_interrupt_R_EVENTQ_as_irqen_low

Interrupt generation is turned off by SMMU_R_IRQ_CTRL.EVENTQ_IRQEN.

warning_discarding_interrupt_R_GERROR_as_irqen_low

Interrupt generation is turned off by SMMU_R_IRQ_CTRL.GERROR_IRQEN.

warning_discarding_interrupt_R_HACDBS_PROCESSING_COMPLETE_as_irqen_low

Interrupt generation is turned off by SMMU_R_IRQ_CTRL.HACDBS_IRQEN.

warning_discarding_interrupt_R_HDBSS_TABLE_FULL_as_irqen_low

Interrupt generation is turned off by SMMU_R_IRQ_CTRL.HDBSS_IRQEN.

warning_discarding_interrupt_R_PRIQ_as_irqen_low

Interrupt generation is turned off by SMMU_R_IRQ_CTRL.PRIQ_IRQEN.

warning_discarding_interrupt_S_EVENTQ_as_irqen_low

Interrupt generation is turned off by SMMU_S_IRQ_CTRL.EVENTQ_IRQEN.

warning_discarding_interrupt_S_GERROR_as_irqen_low

Interrupt generation is turned off by SMMU_S_IRQ_CTRL.GERROR_IRQEN.

warning_discarding_interrupt_S_HACDBS_PROCESSING_COMPLETE_as_irqen_low

Interrupt generation is turned off by SMMU_IRQ_S_CTRL.HACDBS_IRQEN.

warning_discarding_interrupt_S_HDBSS_TABLE_FULL_as_irqen_low

Interrupt generation is turned off by SMMU_IRQ_S_CTRL.HDBSS_IRQEN.

warning_ns_gerrorn_bad_acknowledge

A GERROR was acknowledged by SW in the GERRORN register that did not have an active error. The result is **UNPREDICTABLE** if this will generate an interrupt or not. Fields:

which enum

Which GERROR was acknowledge when there was no active error.

warning_reg_after_doesnt_match_written_value

A write occurred that tried to set bits in a register, that for one reason or another, failed to get written. Fields:

desc string

The textual description of what happened.

warning_rl_gerrorn_bad_acknowledge

A GERROR was acknowledged by SW in the GERRORN register that did not have an active error. The result is **UNPREDICTABLE** if this will generate an interrupt or not. Fields:

which enum

Which GERROR was acknowledge when there was no active error.

warning_s_gerrorn_bad_acknowledge

A GERROR was acknowledged by SW in the GERRORN register that did not have an active error. The result is **UNPREDICTABLE** if this will generate an interrupt or not. Fields:

which enum

Which GERROR was acknowledge when there was no active error.

what_going_to_do_with_terminated_event

A terminating transaction has produced an event, this tells you what the model is going to do with the event. Fields:

CD.S bool

The CD.S field if available.

S2 bool

The event is related to Stage 2.

STE.S1STALLD bool

The STE.S1STALLD field if available.

STE.S2S bool

The STE.S2S field if available.

aborts bool

The transaction will abort.

axmmuflow enum

The AxMMUFLOW for this transaction group. stallable – allow the transaction to be stalled if configured. translated-access – is a PCIe Translated Access non-stallable – do not stall transaction even if configured. transfault-flow – do not report for Translation Related Faults, but tell client device to use a page request mechanism instead (for example PRI).

is_tr_fault bool

Is a Translation Related fault.

protected_mode bool

The transaction is protected-mode. As such, it cannot stall and will obey the report configuration bits.

reports bool

The transaction will attempt to report.

ssd enum

The SSD of the transaction.

ssd_ns bool

The transaction is classified as SSD non-secure.

supports_stall_model bool

The implementation supports the stall model.

trans_id unsigned int

The transaction id.

why_abort_decision enum

The reason why the transaction aborted/did not abort.

why_report_decision enum

The reason why the transaction reported/did not report.

2.127 MMU_L1

This section describes the trace sources.

ArchMsg.Error.error

These messages are about activity occurring on the SMMU that is considered an error. Messages will only come out here if parameter `all_error_messages_through_trace` is true. `DISPLAY %{output}`. Fields:

output string

The stream output.

ArchMsg.Error.fetch_from_memory_type_not_supporting_httu

A descriptor fetch from an HTTU-enabled translation regime to an unsupported memory type was made. Whilst the fetch itself may succeed, if an update to the descriptor was attempted then it would fail. Fields:

address unsigned int

The address of the descriptor fetch.

desc_inner enum

Inner cacheability of descriptor.

desc_outer enum

Outer cacheability of descriptor.

desc_prot enum

The instruction/data, bus-ns/s and privileged/user marking.

desc_sh enum

Shareability of descriptor.

ssd enum

The SSD of the transaction.

ssd_ns bool

Is the Security State Determination of the transaction non-secure.

stage unsigned int

The stage at which we had a problem.

streamid unsigned int

The StreamID of the transaction.

substreamid unsigned int

The SubstreamID of the transaction (or ~0u if no substreamid).

trans_id unsigned int

Transaction id.

ArchMsg.Error.ns_cmdq

There is an error associated with the command queue. Fields:

cmd_id unsigned int

Command ID of the command in error.

cons unsigned int

CONS of the command.

what string

What is wrong.

ArchMsg.Error.priq_streamid_truncated

The SMMU received a PCIe PRI request with a StreamID) that was larger than that which the SMMU has been configured for. The StreamID that appears in the PRIQ entry will be truncated. Fields:

actual_streamid unsigned int

The actual StreamID that this request has.

sidsize unsigned int

The bit width of the SMMU for StreamIDs, as indicated by SMMU_IDR1.SIDSIZE.

trans_id unsigned int

The transaction ID that identifies this request.

trunc_streamid unsigned int

The truncated StreamID that will appear in the PRIQ entry.

ArchMsg.Error.priq_substreamid_truncated

The SMMU received a PCIe PRI request with a PASID prefix (SubstreamID) that was larger than that which the SMMU has been configured for. The SubstreamID that appears in the PRIQ entry will be truncated. Fields:

actual_substreamid unsigned int

The actual SubstreamID that this request has.

ssidsize unsigned int

The bit width of the SMMU for SubstreamIDs, as indicated by SMMU_IDR1.SSIDSIZE.

trans_id unsigned int

The transaction ID that identifies this request.

trunc_substreamid unsigned int

The truncated SubstreamID that will appear in the PRIQ entry.

ArchMsg.Error.s_cmdq

There is an error associated with the command queue. Fields:

cmd_id unsigned int

Command ID of the command in error.

cons unsigned int

CONS of the command.

what string

What is wrong.

ArchMsg.Error.tlb_entries_overlap

A TLB entry was attempted to be inserted into the TLB and was determined that it overlaps an existing entry. This check is not perfect but will catch simple errors. Fields:

do_f_tlb_conflict bool

Chosen to perform an F_TLB_CONFLICT.

end_address_of_new_entry unsigned int

End address of new entry.

end_address_of_old_entry unsigned int

End address of old entry.

index_of_new_entry unsigned int

Index of new entry.

index_of_old_entry unsigned int

Index of old entry.

start_address_of_new_entry unsigned int

Start address of new entry.

start_address_of_old_entry unsigned int

Start address of old entry.

why enum

Why it is an error for these two entries to overlap.

ArchMsg.Error.tlb_entry_not_invalidated_due_to_ril

An entry in the cache was not invalidated even though in the right address range because of the RIL fields of the command the entry do not match. Fields:

cmd_num unsigned int

The NUM field of the RIL part of the command.

cmd_ril_tg enum

The RIL_TG field of the RIL part of the command.

cmd_ril_ttl unsigned int

The TTL field of the RIL part of the command, zero means any level. 0x80 means match level 0 (and is from a DVM message).

cmd_scale unsigned int

The SCALE field of the RIL part of the command.

entry_id unsigned int

The entry id that is being invalidated.

tlb_entry string

The TLB entry.

ArchMsg.Error.vatos_sel_vmid_out_of_range

The SMMU_(S_)VATOS_SEL.VMID field was programmed with a VMID that was too wide for this implementation (SMMU_IDR0.VMID16 == 0). DISPLAY %{{ssd_ns:(s-|ns-)}}VMID:%{vmid} is out of range. Fields:

ssd_ns bool

The security state of the VATOS interface.

vmid unsigned int

The VMID programmed.

ArchMsg.Info.info

These are information messages about what is happening in the SMMU. DISPLAY %{output}. Fields:

output string

The stream output.

ArchMsg.Warning.CMD_RESUME_no_transaction_resumed

A CMD_RESUME was issued that matched no transaction. Fields:

stag unsigned int

STAG in the CMD_RESUME.

streamid unsigned int

StreamID in the CMD_RESUME.

streamid_ns bool

The StreamID was for the non-secure world.

ArchMsg.Warning.atc_inv_strange

Something was odd about the CMD_ATC_INV. DISPLAY CMD_ATC_INV strange as: %{why}. Fields:

cmd_id unsigned int

Command id.

ssd_of_cmdq enum

The SSD of the CMDQ.

why string

Why the CMD_ATC_INV was strange.

ArchMsg.Warning.bad_axi_stream_msi_addr_to_match_s

The signal was not driven at reset by the external system, but was driven after we first started using the default value we had assumed. The external system, if it is going to drive the signal, must do so at first reset so that both sides of the connection can agree on the value of this signal. Fields:

default_value unsigned int

The default value of the signal we have been assuming.

value unsigned int

The value of the signal.

ArchMsg.Warning.bad_conf_reset_of_SMMU_S_GBPA_ABORT

The signal was not driven at reset by the external system, but was driven after we first started using the default value we had assumed. The external system, if it is going to drive the signal, must do so at first reset so that both sides of the connection can agree on the value of this signal. Fields:

default_value bool

The default value of the signal we have been assuming.

value bool

The value of the signal.

ArchMsg.Warning.bad_sec_override

The signal was not driven at reset by the external system, but was driven after we first started using the default value we had assumed. The external system, if it is going to drive the signal, must do so at first reset so that both sides of the connection can agree on the value of this signal. Fields:

default_value bool

The default value of the signal we have been assuming.

value bool

The value of the signal.

ArchMsg.Warning.bad_sup_btm

The signal was not driven at reset by the external system, but was driven after we first started using the default value we had assumed. The external system, if it is going to drive the signal, must do so at first reset so that both sides of the connection can agree on the value of this signal. Fields:

default_value bool

The default value of the signal we have been assuming.

value bool

The value of the signal.

ArchMsg.Warning.bad_sup_cohacc

The signal was not driven at reset by the external system, but was driven after we first started using the default value we had assumed. The external system, if it is going to drive the signal, must do so at first reset so that both sides of the connection can agree on the value of this signal. Fields:

default_value bool

The default value of the signal we have been assuming.

value bool

The value of the signal.

ArchMsg.Warning.bad_sup_httu

The signal was not driven at reset by the external system, but was driven after we first started using the default value we had assumed. The external system, if it is going to drive the signal, must do so at first reset so that both sides of the connection can agree on the value of this signal. Fields:

default_value bool

The default value of the signal we have been assuming.

value bool

The value of the signal.

ArchMsg.Warning.bad_sup_sev

The signal was not driven at reset by the external system, but was driven after we first started using the default value we had assumed. The external system, if it is going to drive the signal, must do so at first reset so that both sides of the connection can agree on the value of this signal. Fields:

default_value bool

The default value of the signal we have been assuming.

value bool

The value of the signal.

ArchMsg.Warning.bad_tbu0_reset_in

The signal was not driven at reset by the external system, but was driven after we first started using the default value we had assumed. The external system, if it is going to drive the signal, must do so at first reset so that both sides of the connection can agree on the value of this signal. Fields:

default_value bool

The default value of the signal we have been assuming.

value bool

The value of the signal.

ArchMsg.Warning.bad_tcu_reset_in

The signal was not driven at reset by the external system, but was driven after we first started using the default value we had assumed. The external system, if it is going to drive the signal, must do so at first reset so that both sides of the connection can agree on the value of this signal. Fields:

default_value bool

The default value of the signal we have been assuming.

value bool

The value of the signal.

ArchMsg.Warning.contig_bit_gives_too_large_region_for_TxSZ

If the contig bit was used then the size of the contig region would be larger than that indicated by TxSZ. Fields:

desc_kind enum

Descriptor kind.

el enum

Exception level.

input_address unsigned int

Input address.

log2_contig_region_size unsigned int

Log2(contig region size in bytes).

output_address unsigned int

Output address.

stage_and_level unsigned int

The stage in bits [7:4] and the level in [3:0]. Level -1 is 0xF.

vmid unsigned int

VMID if appropriate.

ArchMsg.Warning.contig_bit_has_inconsistent_input_and_output_address

If the contig bit was used then the some bits of the output address held in the descriptor and the input address must match. Fields:

desc_kind enum

Descriptor kind.

el enum

Exception level.

input_address unsigned int

Input address.

log2_contig_region_size unsigned int

Log2(contig region size in bytes).

match_mask unsigned int

Bits that must match.

output_address unsigned int

Output address.

stage_and_level unsigned int

The stage in bits [7:4] and the level in [3:0]. Level -1 is 0xF.

vmid unsigned int

VMID if appropriate.

ArchMsg.Warning.fetch_from_memory_type_not_supporting_httu

A descriptor fetch from an HTTU-enabled translation regime to an unsupported memory type was made. Whilst the fetch itself may succeed, if an update to the descriptor was attempted then it would fail. Fields:

address unsigned int

The address of the descriptor fetch.

desc_inner enum

Inner cacheability of descriptor.

desc_outer enum

Outer cacheability of descriptor.

desc_prot enum

The instruction/data, bus-ns/s and privileged/user marking.

desc_sh enum

Shareability of descriptor.

ssd enum

The SSD of the transaction.

ssd_ns bool

Is the Security State Determination of the transaction non-secure.

stage unsigned int

The stage at which we had a problem.

streamid unsigned int

The StreamID of the transaction.

substreamid unsigned int

The SubstreamID of the transaction (or ~0u if no substreamid).

trans_id unsigned int

Transaction id.

ArchMsg.Warning.msi_address_truncated

An MSI was generated, but the address was silently truncated due to the limited downstream address bus width. Fields:

pmcg_index_and_counter unsigned int

If this is a PMCG interrupt, then the top 16 bits are the PMCG index, the lower 16 bits are counter index.

ras_error_group_id_node_id_and_record_index_in_node unsigned int

If this is a RAS interrupt then this is the error group record id in bits[23:16], the node_id in [15:8] and the record index in the node in [7:0].

truncated_address unsigned int

The truncated address.

untruncated_address unsigned int

The untruncated address.

which enum

Which MSI this is.

ArchMsg.Warning.msi_lost

An MSI was attempted to be sent, but couldn't be sent. Fields:

id unsigned int

ID of this interrupt transaction.

kind enum

What kind of interrupt.

why enum

Why this interrupt was denied.

ArchMsg.Warning.pmcg_non_secure_world

A commentary on potential mis-programming of the PMCG when transitioning the PMCG between security states. Fields:

commentary string

The commentary.

ArchMsg.Warning.pmcg_programming_violates_security

A commentary on potential mis-programming of the PMCG when transitioning the PMCG between different security states. This is under the assumption that an agent from another security state could be writing to the PMCG. Fields:

commentary string

The commentary.

ArchMsg.Warning.priq_auto_response_failed_to_find_STE

The PRIQ was going to generate an auto-response, but failed to find an STE and so is returning a Failure message to the EndPoint which should disable the PRI interface of the EndPoint. Fields:

trans_id unsigned int

The ID of the PPR that is being lost.

why enum

Why the PPR is being lost.

ArchMsg.Warning.priq_overflow_bad_acking

Indicates that an overflow condition was acknowledged by writing to:- SMMU_PRIQ_CONS.OVACKFLG but an overflow condition did not exist. Fields:

new_ovackflg bool

The new value of the SMMU_PRIQ_CONS.OVACKFLG.

ArchMsg.Warning.priq_smmuen_forces_effective_priqen_low

If SMMUEN == 0, then the effective value of PRIQEN is 0. This warning is triggered when PRIQEN == 1 && SMMUEN == 0; which may not be what was intended. The PRIQ cannot be active if SMMUEN == 0.

ArchMsg.Warning.sev_lost

A SEV was lost because it isn't supported according to SMMU_IDRO.SEV. DISPLAY SEV was lost because: %{why}. Fields:

why enum

Why the SEV was generated.

ArchMsg.Warning.smmu_pcie_rc_is_in_reset_ignoring_atc_invalidate

A CMD_PRI_RESP or CMD_ATC_INV was sent with a streamid that mapped to a PCIe Root Complex that is in reset. The message will be discarded and act as though completed successfully. Fields:

port_index unsigned int

The port index (node index) that is in reset but we would have sent it to.

streamid unsigned int

The StreamID that failed to map.

ArchMsg.Warning.smmu_pcie_rc_not_found_for_streamid

A CMD_PRI_RESP or CMD_ATC_INV was sent with a streamid that did not map to a PCIe Root Complex. This might be that the SW used an incorrect StreamID or it might be that the model has not been connected correctly. ATC Invalidate messages complete as though successful and PRI Requests are ignored. Fields:

streamid unsigned int

The StreamID that failed to map.

ArchMsg.Warning.smmu_pmusnapshot

Something strange happened on the pmusnapshot_req/pmusnapshot_ack interface. Fields:

pin_index unsigned int

If the PMCG index corresponds to an array of signals, this is the index in the array, or 0 otherwise.

pmcg_index enum

The PMCG index.

warning string

The warning message.

ArchMsg.Warning.suspicious_overlapping_entries

Two DPT TLB entries are overlapping but they differ in ways that are potentially a SW error. Fields:

new_AC enum

The value of the 'AC' field that controls access to this region.

new_FWB bool

The region is FWB.

new_VMID unsigned int

The VMID, if any, associated with this region.

new_entry_index unsigned int

The entry index of the new DPT TLB entry.

new_entry_is_vmsa_formed bool

The new entry was inserted as a result of VMSA information.

new_input_end_incl_address unsigned int

The end inclusive address of the input range that this matches.

new_input_start_address unsigned int

The start address of the input range that this matches.

new_output_pas enum

The output PAS of this DPT region.

new_ssd enum

The SSD of the streams this region captures.

new_writeable bool

True if this region is writeable.

old_AC enum

The value of the 'AC' field that controls access to this region.

old_FWB bool

The region is FWB.

old_VMID unsigned int

The VMID, if any, associated with this region.

old_entry_index unsigned int

The entry index of the old DPT TLB entry.

old_entry_is_vmsa_formed bool

The old entry was inserted as a result of VMSA information.

old_input_end_incl_address unsigned int

The end inclusive address of the input range that this matches.

old_input_start_address unsigned int

The start address of the input range that this matches.

old_output_pas enum

The output PAS of this DPT region.

old_ssd enum

The SSD of the streams this region captures.

old_writeable bool

True if this region is writeable.

pas_differ bool

The output PAS of the two entries differ.

vmids_differ bool

The VMIDs are used by at least one of the AC schemes and are different.

vmsa_formed_writeable_while_DPT_entry_is_not bool

The VMSA-formed entry is writeable but the DPT Entry says it is not writeable.

ArchMsg.Warning.warning

These messages are about unusual (but not necessarily incorrect) activity occurring on the SMMU. DISPLAY %{output}. Fields:

output string

The stream output.

ArchMsg.Warning.warning_effective_EOPD_differs_from_what_would_be_cached_in_TLB

Effective value of EOPD differs from what would be cached in the TLB DISPLAY transaction (%{transaction_id}), sid (%{sid}), ssid (%{ssid}), ssd (%{ssd}), effective EOPD (%{effective_EOPD}), cached EOPD (%{cached_EOPD}). Fields:

cached_EOPD bool

The EOPD value that would be cached in the TLB.

effective_EOPD bool

The effective value of EOPD.

ssd enum

SSD.

streamid unsigned int

StreamID or ~0ull if NoStreamID.

substreamid unsigned int

SubstreamID or ~0u if SSValid == false.

transaction_id unsigned int

The transaction ID.

DPTTLB.tlb_commentary

A commentary from the cache about what is going on. Fields:

output string

The stream output.

DPTTLB.tlb_entry_invalidated

An entry in the cache has been invalidated. Fields:

entry_id unsigned int

The entry id that is being invalidated.

reason enum

Why the entry is being invalidated.

EVENTQ_config

Configuration of interrupt updated. Fields:

address unsigned int

Address of the MSI.

data unsigned int

Data payload of the MSI.

irqen bool

Appropriate bit of SMMU_s_IRQ_CTRL controlling this interrupt.

memattr enum

Memory type.

ns bool

PAS of bus attribute is non-secure.

pas enum

PAS of the MSI.

ra bool

read-allocate attribute.

sh enum

Shareability.

smmu_idr0_msi bool

Are MSIs supported for this security world.

tr bool

transient attribute.

wa bool

write-allocate attributes.

EVENTQ_irqen

The IRQEN has been changed. Fields:

new_value bool

The new value of IRQEN.

EVENTQ_irqen_ack

The IRQEN change has been acked. Fields:

new_value bool

The new value of IRQEN.

GERROR_config

Configuration of interrupt updated. Fields:

address unsigned int

Address of the MSI.

data unsigned int

Data payload of the MSI.

irqen bool

Appropriate bit of SMMU_s_IRQ_CTRL controlling this interrupt.

memattr enum

Memory type.

ns bool

PAS of bus attribute is non-secure.

pas enum

PAS of the MSI.

ra bool

read-allocate attribute.

sh enum

Shareability.

smmu_idr0_msi bool

Are MSIs supported for this security world.

tr bool

transient attribute.

wa bool

write-allocate attributes.

GERROR_irqen

The IRQEN has been changed. Fields:

new_value bool

The new value of IRQEN.

GERROR_irqen_ack

The IRQEN change has been acked. Fields:

new_value bool

The new value of IRQEN.

HACDBS_PROCESSING_COMPLETE_config

Configuration of interrupt updated. Fields:

address unsigned int

Address of the MSI.

data unsigned int

Data payload of the MSI.

irqen bool

Appropriate bit of SMMU_s_IRQ_CTRL controlling this interrupt.

memattr enum

Memory type.

ns bool

PAS of bus attribute is non-secure.

pas enum

PAS of the MSI.

ra bool

read-allocate attribute.

sh enum

Shareability.

smmu_idr0_msi bool

Are MSIs supported for this security world.

tr bool

transient attribute.

wa bool

write-allocate attributes.

HACDBS_PROCESSING_COMPLETE_irqen

The IRQEN has been changed. Fields:

new_value bool

The new value of IRQEN.

HACDBS_PROCESSING_COMPLETE_irqen_ack

The IRQEN change has been acked. Fields:

new_value bool

The new value of IRQEN.

HDBSS_TABLE_FULL_config

Configuration of interrupt updated. Fields:

address unsigned int

Address of the MSI.

data unsigned int

Data payload of the MSI.

irqen bool

Appropriate bit of SMMU_s_IRQ_CTRL controlling this interrupt.

memattr enum

Memory type.

ns bool

PAS of bus attribute is non-secure.

pas enum

PAS of the MSI.

ra bool

read-allocate attribute.

sh enum

Shareability.

smmu_idr0_msi bool

Are MSIs supported for this security world.

tr bool

transient attribute.

wa bool

write-allocate attributes.

HDBSS_TABLE_FULL_irqen

The IRQEN has been changed. Fields:

new_value bool

The new value of IRQEN.

HDBSS_TABLE_FULL_irqen_ack

The IRQEN change has been acked. Fields:

new_value bool

The new value of IRQEN.

MMU_L1_integration_mode_end_ras_level_interrupt_restored

RAS level sensitive interrupt restored due to integration mode ending. Fields:

is_tcu bool

Is TCU integration mode.

ras_interrupt enum

RAS interrupt being restored.

tbu_index unsigned int

TBU index. ~0u if it's TCU.

value bool

Level of the interrupt signal.

MMU_L1_integration_mode_pmcg_interrupt_lost

PMCG interrupt lost due to being in integration mode. Fields:

is_tcu enum

Is TCU integration mode.

pmcg_interrupt enum

PMCG interrupt being dropped.

tbu_index unsigned int

TBU index. ~0u if it's TCU.

MMU_L1_integration_mode_pmusnapshot_ack_lost

pmusnapshot_ack lost due to being in integration mode. Fields:

is_tcu enum

Is TCU integration mode.

tbu_index unsigned int

TBU index. ~0u if it's TCU.

MMU_L1_integration_mode_ras_interrupt_lost

RAS interrupt lost due to being in integration mode. Fields:

is_tcu enum

Is TCU integration mode.

ras_interrupt enum

RAS interrupt being dropped.

tbu_index unsigned int

TBU index. ~0u if it's TCU.

MMU_L1_integration_mode_ras_level_interrupt_lost

RAS level sensitive interrupt lost due to being in integration mode. Fields:

is_tcu bool

Is TCU integration mode.

ras_interrupt enum

RAS interrupt being dropped.

tbu_index unsigned int

TBU index. ~0u if it's TCU.

value bool

Level of the interrupt signal.

MMU_L1_integration_mode_start_ras_level_interrupt_cleared

RAS level sensitive interrupt cleared due to integration mode starting. Fields:

is_tcu bool

Is TCU integration mode.

ras_interrupt enum

RAS interrupt being cleared.

tbu_index unsigned int

TBU index. ~0u if it's TCU.

MMU_L1_integration_mode_tcu_evento_lost

Evento lost due to being in integration mode.

MMU_L1_integration_mode_tcu_interrupt_lost

Interrupt lost due to being in integration mode. Fields:

interrupt enum

Interrupt that is being dropped.

MMU_L1_integration_register_change

An integration register write occurred and it's modified, which drives a signal. Fields:

is_tcu enum

Is TCU integration mode.

new_value bool

New value of the signal.

old_value bool

Old value of the signal.

register enum

The register being modified.

register_value unsigned int

Value written to the register.

signal string

Signal affected by the register write.

tbu_index unsigned int

TBU index. ~0u if it's TCU.

PRIQ_config

Configuration of interrupt updated. Fields:

address unsigned int

Address of the MSI.

data unsigned int

Data payload of the MSI.

irqen bool

Appropriate bit of SMMU_s_IRQ_CTRL controlling this interrupt.

memattr enum

Memory type.

ns bool

PAS of bus attribute is non-secure.

pas enum

PAS of the MSI.

ra bool

read-allocate attribute.

sh enum

Shareability.

smmu_idr0_msi bool

Are MSIs supported for this security world.

tr bool

transient attribute.

wa bool

write-allocate attributes.

PRIQ_irqen

The IRQEN has been changed. Fields:

new_value bool

The new value of IRQEN.

PRIQ_irqen_ack

The IRQEN change has been acked. Fields:

new_value bool

The new value of IRQEN.

SMMU_CR0ACK_SMMUEN_hazarded_by_priq

The SMMU_r_CR0ACK.SMMUEN cannot acknowledge the change to SMMUEN because there are outstanding PRIQ writes.

SMMU_CR0ACK_SMMUEN_update

The acknowledge to SMMU_CR0.SMMUEN was performed. Fields:

new_value bool

The new value of the register.

SMMU_CR0_SMMUEN_old_set_complete

A set of transactions associated with the old value of SMMU_CR0.SMMUEN completed. Fields:

last bool

This is the last set completing.

old_value bool

The old value of the set completing.

SMMU_CR0_SMMUEN_write

A write to SMMU_CR0.SMMUEN occurred. Fields:

new_value bool

The new value of the register.

old_value bool

The old value of the register.

SMMU_GBPA_old_set_complete

A set of transactions associated with the old value of SMMU_GBPA completed. Fields:

last bool

This is the last set completing.

old_value unsigned int

The old value of the set completing.

SMMU_GBPA_update

The Update flag to SMMU_GBPA was lowered. Fields:

new_value unsigned int

The new value of the register.

SMMU_GBPA_write

A write to SMMU_GBPA occurred. Fields:

new_value unsigned int

The new value of the register.

old_value unsigned int

The old value of the register.

SMMU_S_CR0ACK_SMMUEN_update

The acknowledge to SMMU_S_CR0.SMMUEN was performed. Fields:

new_value bool

The new value of the register.

SMMU_S_CR0_SMMUEN_old_set_complete

A set of transactions associated with the old value of SMMU_S_CR0.SMMUEN completed. Fields:

last bool

This is the last set completing.

old_value bool

The old value of the set completing.

SMMU_S_CR0_SMMUEN_write

A write to SMMU_S_CR0.SMMUEN occurred. Fields:

new_value bool

The new value of the register.

old_value bool

The old value of the register.

SMMU_S_GBPA_old_set_complete

A set of transactions associated with the old value of SMMU_S_GBPA completed. Fields:

last bool

This is the last set completing.

old_value unsigned int

The old value of the set completing.

SMMU_S_GBPA_update

The Update flag to SMMU_S_GBPA was lowered. Fields:

new_value unsigned int

The new value of the register.

SMMU_S_GBPA_write

A write to SMMU_S_GBPA occurred. Fields:

new_value unsigned int

The new value of the register.

old_value unsigned int

The old value of the register.

S_EVENTQ_config

Configuration of interrupt updated. Fields:

address unsigned int

Address of the MSI.

data unsigned int

Data payload of the MSI.

irgen bool

Appropriate bit of SMMU_s_IRQ_CTRL controlling this interrupt.

memattr enum

Memory type.

ns bool

PAS of bus attribute is non-secure.

pas enum

PAS of the MSI.

ra bool

read-allocate attribute.

sh enum

Shareability.

smmu_idr0_msi bool

Are MSIs supported for this security world.

tr bool

transient attribute.

wa bool

write-allocate attributes.

S_EVENTQ_irqen

The IRQEN has been changed. Fields:

new_value bool

The new value of IRQEN.

S_EVENTQ_irqen_ack

The IRQEN change has been acked. Fields:

new_value bool

The new value of IRQEN.

S_GERROR_config

Configuration of interrupt updated. Fields:

address unsigned int

Address of the MSI.

data unsigned int

Data payload of the MSI.

irqen bool

Appropriate bit of SMMU_s_IRQ_CTRL controlling this interrupt.

memattr enum

Memory type.

ns bool

PAS of bus attribute is non-secure.

pas enum

PAS of the MSI.

ra bool

read-allocate attribute.

sh enum

Shareability.

smmu_idr0_msi bool

Are MSIs supported for this security world.

tr bool

transient attribute.

wa bool

write-allocate attributes.

S_GERROR_irqen

The IRQEN has been changed. Fields:

new_value bool

The new value of IRQEN.

S_ERROR_irqen_ack

The IRQEN change has been acked. Fields:

new_value bool

The new value of IRQEN.

S_HACDBS_PROCESSING_COMPLETE_config

Configuration of interrupt updated. Fields:

address unsigned int

Address of the MSI.

data unsigned int

Data payload of the MSI.

irqen bool

Appropriate bit of SMMU_s_IRQ_CTRL controlling this interrupt.

memattr enum

Memory type.

ns bool

PAS of bus attribute is non-secure.

pas enum

PAS of the MSI.

ra bool

read-allocate attribute.

sh enum

Shareability.

smmu_idr0_msi bool

Are MSIs supported for this security world.

tr bool

transient attribute.

wa bool

write-allocate attributes.

S_HACDBS_PROCESSING_COMPLETE_irqen

The IRQEN has been changed. Fields:

new_value bool

The new value of IRQEN.

S_HACDBS_PROCESSING_COMPLETE_irqen_ack

The IRQEN change has been acked. Fields:

new_value bool

The new value of IRQEN.

S_HDBSS_TABLE_FULL_config

Configuration of interrupt updated. Fields:

address unsigned int

Address of the MSI.

data unsigned int

Data payload of the MSI.

irqen bool

Appropriate bit of SMMU_s_IRQ_CTRL controlling this interrupt.

memattr enum

Memory type.

ns bool

PAS of bus attribute is non-secure.

pas enum

PAS of the MSI.

ra bool

read-allocate attribute.

sh enum

Shareability.

smmu_idr0_msi bool

Are MSIs supported for this security world.

tr bool

transient attribute.

wa bool

write-allocate attributes.

S_HDBSS_TABLE_FULL_irqen

The IRQEN has been changed. Fields:

new_value bool

The new value of IRQEN.

S_HDBSS_TABLE_FULL_irqen_ack

The IRQEN change has been acked. Fields:

new_value bool

The new value of IRQEN.

TLB.tlb_commentary

A commentary from the cache about what is going on. Fields:

output string

The stream output.

TLB.tlb_entry_invalidated

An entry in the cache has been invalidated. Fields:

entry_id unsigned int

The entry id that is being invalidated.

reason enum

Why the entry is being invalidated.

atc_inv_nop

The CMD_ATC_INV command is ignored as a **NOP**. This may be emitted multiple times if the CMD_ATC_INV is being ignored for multiple reasons. Fields:

address unsigned int

Untranslated Address to invalidate.

cmd_id unsigned int

Command id.

cons unsigned int

CONS of the command.

global bool

Global flag.

log2_size_in_bytes unsigned int

Log 2 of the size in bytes of the region to invalidate.

ssd_of_queue enum

The SSD of the CMDQ.

ssd_of_streamid enum

The SSD of the StreamID.

streamid unsigned int

StreamID in the invalidate command.

substreamid unsigned int

SubstreamID or ~0u if SSValid == false.

why enum

Why was NOPped.

atos_complete_fault

The ATOS operation completed with a fault. Fields:

effective_st1translate bool

Because of the settings and/or the ATOS type then the effective st1translate can be different.

fault_faddr unsigned int

The fault FADDR.

fault_faultcode enum

The fault code.

fault_reason enum

The fault reason.

ssd_ns bool

This is a non-secure ATOS operation.

st1translate bool

The StreamID/SubstreamID combination should have been translated by stage 1.

st2translate bool

The translation suffered a stage 2 translation.

state enum

The final transaction state of the ATOS operation.

vatos_vmid unsigned int

The VMID of the VATOS operation, or ~0u if not a VATOS operation.

atos_complete_fault_inv_req

The ATOS operation completed, faulted and generated an INV_REQ response. Fields:

ssd_ns bool

This is a non-secure ATOS operation.

vatos_vmid unsigned int

The VMID of the VATOS operation, or ~0u if not a VATOS operation.

why enum

Why the ATOS operation generated an INV_REQ.

atos_complete_success

The ATOS operation completed successfully. Fields:

base_addr unsigned int

The actual base address of region.

effective_st1translate bool

Because of the settings and/or the ATOS type then the effective st1translate can be different.

par_addr unsigned int

The PAR.ADDR field.

par_mair unsigned int

The memory attributes encoded as a MAIR.

par_ns bool

The PAR.NS field, for an SSD-ns request then this will always be 0.

par_sh enum

Shareability.

par_size bool

The PAR.Size field.

size_in_bytes unsigned int

The actual size in bytes of the region.

ssd_ns bool

This is a non-secure ATOS operation.

st1translate bool

The StreamID/SubstreamID combination should have been translated by stage 1.

st2translate bool

The translation suffered a stage 2 translation.

state enum

The final transaction state of the ATOS operation.

vatos_vmid unsigned int

The VMID of the VATOS operation, or ~0u if not a VATOS operation.

atos_run_set

The SMMU_s_GATOS_CTRL.RUN field was set to start the ATOS operation. Fields:

ssd_ns bool

This is a non-secure ATOS operation.

vatos bool

This is a VATOS operation.

atos_starting

The ATOS operation is starting. Fields:

addr unsigned int

The input address to the ATOS operation.

httui bool

Inhibit HTTU update.

ind bool

Instruction Data.

pnu bool

Privileged not User.

rnw bool

Read not Write.

ssd_ns bool

This is a non-secure ATOS operation.

ssec bool

If this is a secure ATOS operation then this is if it is secure or not.

streamid unsigned int

The StreamID requested.

substreamid unsigned int

The SubstreamID requested, or ~0u if no SubstreamID.

type enum

The requested ATOS type.

vatos_vmid unsigned int

The VMID of the VATOS operation, or ~0u if not a VATOS operation.

axi_stream_msi_addr_to_match_s

Address to use to send SMMU originated MSIs directly to the GIC. This pin is sampled on the negedge of the reset pin and has to be driven beforehand for the value to take effect. Fields:

value unsigned int

The value of the signal.

cd_cc.CD_commentary

A commentary from the cache about what is going on. Fields:

output string

The stream output.

cd_cc.CD_entry_invalidated

An entry in the cache has been invalidated. Fields:

entry_id unsigned int

The entry id that is being invalidated.

reason enum

Why the entry is being invalidated.

cd_entry_allocated

An CD entry has been allocated. Fields:

AssuredTranslation bool

The CD (and any L1CD) was fetched from an AssuredOnly stage 2 page and so might be capable of creating AssuredTranslations.

cd string

A textual description of the CD.

entry_id unsigned int

The id of the cache entry. Top bit is set if is a debug entry.

ns enum

For the non-secure world.

ssd enum

The SSD of the entry.

streamid unsigned int

The StreamID of the entry this will match.

substreamid unsigned int

The SubstreamID of the entry this will match. This may be zero for transactions without a SubstreamID.

conf_reset_of_SMMU_S_GBPA_ABORT

The pin is driven. This is the reset value of SMMU_S_GBPA.ABORT. This pin is sampled on the negedge of the reset pin and has to be driven beforehand for the value to take effect. Fields:

value bool

The value of the signal.

dpttlb_entry_allocated

A DPT TLB entry has been allocated. Fields:

AC enum

The value of the 'AC' field that controls access to this region.

FWB bool

The region is FWB.

VMID unsigned int

The VMID, if any, associated with this region.

index unsigned int

Index of the TLB entry.

input_end_incl_address unsigned int

The end inclusive address of the input range that this matches.

input_start_address unsigned int

The start address of the input range that this matches.

output_pas enum

The output PAS of this DPT region.

ssd enum

The SSD of the streams this region captures.

trans_id unsigned int

The trans_id of the transaction that caused this allocation.

vmsa_formed bool

The entry was formed from VMSA information rather than from a DPT walk.

writeable bool

True if this region is writeable.

dpttlb_invalidate_intersects_but_does_not_cover_entry_range

ENCODED_SIZE < size of the region covered by the DPT entry so invalidation is not architecturally guaranteed. No invalidation is performed. Fields:

dpttlb_entry_id unsigned int

ID of the TLB entry.

entry_end_incl_address unsigned int

Last address covered by the TLB entry.

entry_start_address unsigned int

First address covered by the TLB entry.

invalidate_end_incl_address unsigned int

Last address covered by the invalidation range.

invalidate_start_address unsigned int

First address covered by the invalidation range.

ssd enum

The security state of the TLB entry.

dpttlb_overlapping_entries

Two DPT TLB entries are overlapping. Fields:

new_AC enum

The value of the 'AC' field that controls access to this region.

new_FWB bool

The region is FWB.

new_VMID unsigned int

The VMID, if any, associated with this region.

new_entry_index unsigned int

The entry index of the new DPT TLB entry.

new_entry_is_vmsa_formed bool

The new entry was inserted as a result of VMSA information.

new_input_end_incl_address unsigned int

The end inclusive address of the input range that this matches.

new_input_start_address unsigned int

The start address of the input range that this matches.

new_output_pas enum

The output PAS of this DPT region.

new_ssd enum

The SSD of the streams this region captures.

new_writeable bool

True if this region is writeable.

old_AC enum

The value of the 'AC' field that controls access to this region.

old_FWB bool

The region is FWB.

old_VMID unsigned int

The VMID, if any, associated with this region.

old_entry_index unsigned int

The entry index of the old DPT TLB entry.

old_entry_is_vmsa_formed bool

The old entry was inserted as a result of VMSA information.

old_input_end_incl_address unsigned int

The end inclusive address of the input range that this matches.

old_input_start_address unsigned int

The start address of the input range that this matches.

old_output_pas enum

The output PAS of this DPT region.

old_ssd enum

The SSD of the streams this region captures.

old_writeable bool

True if this region is writeable.

dvm_tlbinvalidate_complete

The DVM TLB Invalidate message completed. Fields:

id unsigned int

The unique id of this DVM message.

ok bool

The DVM message was OK.

dvm_tlbinvalidate_received

A DVM message for a TLB Invalidate has been received. Fields:

address unsigned int

The VA or IPA to use if match_address.

asid unsigned int

The ASID to match if match_asid.

by_ipa bool

The operation is for an IPA operation if match_address.

id unsigned int

The unique id of this DVM message.

ignored enum

The DVM message was ignored.

last_level bool

The operation is for last level if supported.

match_address bool

Match the address field.

match_asid bool

Match the asid field.

match_vmid bool

Match the vmid field.

num unsigned int

If a range operation, the NUM field. If a single-address operation this is 0.

prot enum

The protection level for which this TLB Invalidate will operate on.

security_world enum

The security world that this will apply to.

smmu_scale unsigned int

If a range operation, then the SCALE field with the meaning in the SMMU architecture which is different to the PE architecture. If a single-address operation this is 0.

stage1_only bool

The operation is for stage 1 only if supported.

tg enum

If a single-address or address-range operation, then the Translation Granule hint. Address-range operations always supply a Translation Granule.

translation_table_level enum

The leaf level of the translation table.

vmid unsigned int

The VMID to match if match_vmid.

found_tlb_entry_has_different_aset

Architecturally, a particular ASID either should be ASET0 or ASET1. However, we have managed to find a TLB entry that has a different ASET than that which we were searching for. This indicates a programming error. You should examine all contexts with this particular ASID/VMID and ensure they are consistent. Fields:

aset unsigned int

The ASET of this entry.

asid unsigned int

ASID if appropriate.

index unsigned int

Index of TLB entry.

input_start_address unsigned int

The start address of the input range that this matches.

scheme enum

The tagging scheme used.

ssd enum

The SSD of the tagging scheme.

ssd_ns enum

Security State.

vmid unsigned int

VMID if appropriate.

httu_update_abandoned_update

The HTTU update of a descriptor in memory was potentially possible, but it was behind an update that failed to apply cleanly. Fields:

AF enum

Whether an AF update should/could occur.

DBM enum

Whether a DBM update should/could occur.

address unsigned int

Address of HTTU update.

httu_inner enum

Inner cacheability of descriptor to update.

httu_outer enum

Outer cacheability of descriptor to update.

httu_prot enum

The instruction/data, bus-ns/s and privileged/user marking.

httu_sh enum

Shareability of descriptor to update.

mpam_pmg_and_partid unsigned int

Bits[23:16] are the MPAM PMG. Bits[15:0] are the physical PARTID.

mpam_sp enum

The MPAM_SP of the transaction.

original_descriptor unsigned int

The original descriptor that the model observed.

stage_and_level unsigned int

The stage (bits[7:4]) and level (bits[3:0]). A level of 0xF means -1.

trans_id unsigned int

The transaction id that this HTTU update is associated with.

what enum

What this descriptor represents.

httu_update_end_update

The attempted update of a descriptor in memory has occurred. Fields:

is_big_endian bool

The descriptor is big-endian in memory.

original_descriptor unsigned int

The original descriptor value.

result enum

The result of the attempt to update.

trans_id unsigned int

The transaction id that this HTTU update is associated with.

try_to_change_to_descriptor unsigned int

The new descriptor value that tried to replace the original.

value_that_was_in_memory unsigned int

The value that the compare-and-swap operation returned as the value that was in memory.

httu_update_not_done

A discretionary HTTU update could occur and the implementation choose not to do it. A mandatory_do_if_linked_performed means that this is a stage 2 leaf descriptor that maps a subsequent stage 1 leaf descriptor whose update is discretionary and if that subsequent discretionary update is going to occur then this update becomes mandatory. Fields:

AF enum

Whether an AF update should/could occur.

DBM enum

Whether a DBM update should/could occur.

address unsigned int

Address of HTTU update.

httu_inner enum

Inner cacheability of descriptor to update.

httu_outer enum

Outer cacheability of descriptor to update.

httu_prot enum

The instruction/data, bus-ns/s and privileged/user marking.

httu_sh enum

Shareability of descriptor to update.

mpam_pmg_and_partid unsigned int

Bits[23:16] are the MPAM PMG. Bits[15:0] are the physical PARTID.

mpam_sp enum

The MPAM_SP of the transaction.

original_descriptor unsigned int

The original descriptor that the model observed.

stage_and_level unsigned int

The stage (bits[7:4]) and level (bits[3:0]). A level of 0xF means -1.

trans_id unsigned int

The transaction id that this HTTU update is associated with.

what enum

What this descriptor represents.

httu_update_start_update

An HTTU update could occur and the implementation chose to try it. A mandatory_do_if_linked_performed means that this is a stage 2 leaf descriptor that maps a subsequent stage 1 leaf descriptor whose update is discretionary and if that subsequent discretionary update is going to occur then this update becomes mandatory. Fields:

AF enum

Whether an AF update should/could occur.

DBM enum

Whether a DBM update should/could occur.

address unsigned int

Address of HTTU update.

httu_inner enum

Inner cacheability of descriptor to update.

httu_outer enum

Outer cacheability of descriptor to update.

httu_prot enum

The instruction/data, bus-ns/s and privileged/user marking.

httu_sh enum

Shareability of descriptor to update.

is_big_endian bool

The descriptor will be written to memory as big-endian.

mecid unsigned int

The masked MECID used for the update transaction, or ~0u if not appropriate.

mpam_pmg_and_partid unsigned int

Bits[23:16] are the MPAM PMG. Bits[15:0] are the physical PARTID.

mpam_sp enum

The MPAM_SP of the transaction.

original_descriptor unsigned int

The original descriptor that the model observed.

stage_and_level unsigned int

The stage (bits[7:4]) and level (bits[3:0]). A level of 0xF means -1.

trans_id unsigned int

The transaction id that this HTTU update is associated with.

try_to_change_to_descriptor unsigned int

The new descriptor value that is going to try replace the original.

what enum

What this descriptor represents.

will_do_AF bool

What the implementation chose to do for the AF update.

will_do_DBM bool

What the implementation chose to do for the DBM update.

interrupt_returned

An interrupt/MSI returned from downstream. Fields:

InD_NS_PnU enum

Instruction/Data, Non-secure/Secure/Root/Realm, Privileged/User. Bit[1] = PAS[0], and bit[3] = PAS[1].

address unsigned int

Address of the MSI.

data unsigned int

Data payload of the MSI.

dcmdqcp_and_index unsigned int

If this not a DCMDQ MSI CMD_SYNC, then this is 0xFFFFFFFF. If it is then the DCMDQ QCP index is in bits[31:16] and the index in the QCP is in [7:0].

gpc_denied_msi bool

True if the MSI was denied as it failed its GPC checks. Thus the field 'ok' will be false.

id unsigned int

ID of this interrupt transaction.

inner_cache enum

The actual attributes of the access.

kind enum

What kind of interrupt.

mecid unsigned int

The MECID of the MSI, if applicable.

mpam_ns bool

For a non-RME system, this is true if the MPAM space is Non-secure. For an RME system, this is MPAM_SP[0].

mpam_pmg_and_partid unsigned int

Bits[23:16] are the MPAM PMG. Bits[15:0] are the physical PARTID.

mpam_sp enum

The MPAM_SP security state of the MPAM partition. 0 – secure, 1 – non-secure, 2 – root, 3 – realm.

msi_valid bool

MSI will attempt to be sent.

ok bool

Did the access return OK or an abort?.

outer_cache enum

The actual attributes of the access.

pas enum

The PAS of the MSI, if applicable.

pmcg_index_and_counter unsigned int

If this is a PMCG interrupt, then the top 16 bits are the PMCG index, the lower 16 bits are counter index.

ras_error_group_id_node_id_and_record_index_in_node unsigned int

If this is a RAS interrupt then this is the error group record id in bits[23:16], the node_id in [15:8] and the record index in the node in [7:0].

sh enum

Shareability.

interrupt_sent

An interrupt is raised. If it sends an MSI then this is *after* any device-dependent transform on the architectural attributes and so may differ from what is programmed. Fields:

InD_NS_PnU enum

Instruction/Data, Non-secure/Secure/Root/Realm, Privileged/User. Bit[1] = PAS[0], and bit[3] = PAS[1].

address unsigned int

Address of the MSI.

data unsigned int

Data payload of the MSI.

dcmdq_device_id unsigned int

If this is a DCMDQ then this is the DeviceID of the MSI write. Otherwise, 0xFFFF'ffff.

dcmdq_qcp_and_index unsigned int

If this not a DCMDQ MSI CMD_SYNC, then this is 0xFFFF'ffff. If it is then the DCMDQ QCP index is in bits[31:16] and the index in the QCP is in [7:0].

id unsigned int

ID of this interrupt transaction.

inner_cache enum

The actual attributes of the access.

kind enum

What kind of interrupt.

mecid unsigned int

The MECID of the MSI, if applicable.

mpam_ns bool

For a non-RME system, this is true if the MPAM space is Non-secure. For an RME system, this is MPAM_SP[0].

mpam_pmg_and_partid unsigned int

Bits[23:16] are the MPAM PMG. Bits[15:0] are the physical PARTID.

mpam_sp enum

The MPAM_SP security state of the MPAM partition. 0 – secure, 1 – non-secure, 2 – root, 3 – realm.

msi_valid bool

MSI will attempt to be sent.

outer_cache enum

The actual attributes of the access.

pas enum

The PAS of the MSI, if applicable.

pmcg_index_and_counter unsigned int

If this is a PMCG interrupt, then the top 16 bits are the PMCG index, the lower 16 bits are counter index.

qSID unsigned int

If this is a DCMDQ then this is the qSID of the MSI write. Otherwise, 0xFFFF'ffff.

ras_error_group_id_node_id_and_record_index_in_node unsigned int

If this is a RAS interrupt then this is the error group record id in bits[23:16], the node_id in [15:8] and the record index in the node in [7:0].

sh enum

Shareability.

l1cd_cc.L1CD_commentary

A commentary from the cache about what is going on. Fields:

output string

The stream output.

l1cd_cc.L1CD_entry_invalidated

An entry in the cache has been invalidated. Fields:

entry_id unsigned int

The entry id that is being invalidated.

reason enum

Why the entry is being invalidated.

l1cd_entry_allocated

An L1 CD entry has been allocated. Fields:

AssuredTranslation bool

The L1CD was fetched from an AssuredOnly stage 2 page and so might be capable of creating AssuredTranslations.

entry_id unsigned int

The id of the cache entry. Top bit is set if is a debug entry.

normalised_substreamid unsigned int

The first SubstreamID of the range of SubstreamIDs that this L1CD entry will match.

ns enum

For the non-secure world.

pa_12 unsigned int

The PA of the L2 CD table. This is L2Ptr << 12.

ssd enum

The SSD of the entry.

streamid unsigned int

The StreamID this CD is for.

valid bool

Is the entry valid.

l1ste_cc.L1STE_commentary

A commentary from the cache about what is going on. Fields:

output string

The stream output.

l1ste_cc.L1STE_entry_invalidated

An entry in the cache has been invalidated. Fields:

entry_id unsigned int

The entry id that is being invalidated.

reason enum

Why the entry is being invalidated.

l1ste_entry_allocated

An L1 STE entry has been allocated. Fields:

entry_id unsigned int

The id of the cache entry. Top bit is set if is a debug entry.

first_streamid_of_range unsigned int

The first StreamID of the range of StreamIDs that this L1STE entry will match.

ns enum

For the non-secure world.

num_entries_in_l2 unsigned int

The number of entries in the L2 table. This is $2^{(\text{Span}-1)}$ or 0 if invalid.

pa_l2 unsigned int

The PA of the L2 ST table. This is $\text{L2Ptr} \ll 6$ and aligned to the size of the table.

ssd enum

The SSD of the entry.

level_interrupt_sent

A level interrupt changed state. Fields:

kind enum

What kind of interrupt.

ras_error_group_id_node_id_and_record_index_in_node unsigned int

If this is a level RAS interrupt then this is the error group record id in bits[23:16], the node_id in [15:8] and the record index in the node in [7:0].

set_high bool

Level interrupt state.

ns_cmd_sync_completed_irq

“Non-secure” CMD_SYNC completed its IRQ action. Fields:

cmd_id unsigned int

The ID of the command being issued.

cons_incl_wrap unsigned int

CONS pointer of CMD_SYNC.

is_synthetic bool

This is a synthetic sync. The cons_incl_wrap is 0xFFFF'ffff.

ns_cmd_sync_completed_no_action

Non-secure CMD_SYNC completed, there was no CMD_SYNC action (SIG_NONE). Fields:

cmd_id unsigned int

The ID of the command being issued.

cons_incl_wrap unsigned int

CONS pointer of CMD_SYNC.

is_synthetic bool

This is a synthetic sync. The cons_incl_wrap is 0xFFFF'ffff.

ns_cmd_sync_completed_sev

Non-secure CMD_SYNC completed its action of SIG_SEV. Fields:

cmd_id unsigned int

The ID of the command being issued.

cons_incl_wrap unsigned int

CONS pointer of CMD_SYNC.

is_synthetic bool

This is a synthetic sync. The cons_incl_wrap is 0xFFFF'ffff.

ns_cmd_sync_completed_with_error

Non-secure CMD_SYNC completed, however, there was an error associated with the completion of the CMD_SYNC. Fields:

error enum

The error associated with the CMD_SYNC.

cmd_id unsigned int

The ID of the command being issued.

cons_incl_wrap unsigned int

CONS pointer of CMD_SYNC.

is_synthetic bool

This is a synthetic sync. The cons_incl_wrap is 0xFFFF'ffff.

ns_cmd_sync_issued

Non-secure CMD_SYNC has now completed issuing and we now know if something is hazarding the CMD_SYNC from completing. Fields:

cmd_id unsigned int

The ID of the command being issued.

cons_incl_wrap unsigned int

CONS pointer of CMD_SYNC.

hazarded bool

The CMD_SYNC is currently hazarded from completing.

is_synthetic bool

This is a synthetic sync. The cons_incl_wrap is 0xFFFF'ffff.

ns_cmd_sync_issuing

Non-secure CMD_SYNC is starting to issue, we are going to start accounting for which things have to be part of the prior set of things that have to complete before the CMD_SYNC can complete. Fields:

cmd_id unsigned int

The ID of the command being issued.

cons_incl_wrap unsigned int

CONS pointer of CMD_SYNC.

is_synthetic bool

This is a synthetic sync. The cons_incl_wrap is 0xFFFF'ffff.

ns_cmd_sync_starting_completion_action_irq

Non-secure CMD_SYNC starting IRQ completion action. Fields:

cmd_id unsigned int

The ID of the command being issued.

cons_incl_wrap unsigned int

CONS pointer of CMD_SYNC.

is_synthetic bool

This is a synthetic sync. The cons_incl_wrap is 0xFFFF'ffff.

msiaddress unsigned int

The MSI address to use (or 0 if not sending an MSI).

msidata unsigned int

The MSI data to send.

ns_cmd_sync_starting_completion_action_sev

Non-secure CMD_SYNC starting its completion action of SIG_SEV. Fields:

cmd_id unsigned int

The ID of the command being issued.

cons_incl_wrap unsigned int

CONS pointer of CMD_SYNC.

is_synthetic bool

This is a synthetic sync. The cons_incl_wrap is 0xFFFF'ffff.

ns_cmdq_cmd_sync_error

The CMD_SYNC can fault if an ATC Invalidate timeout occurs. The timeout will be attached to the next CMD_SYNC on the queue on which the invalidate was launched. Fields:

cerror enum

The CERROR to attach.

cmd_queue unsigned int

The top bits indicate the CMDQ id. These bits are attached to all cmd_ids generated for this queue. The bits [63:44] contain the queue ID. [63:62] 0: s-ssd, 1: reserved, 2: ns-ssd, 3: rl-ssd [61] is ECMDQ [59:52] QCP index [51:44] ECMDQ index in QCP.

herror enum

The HERROR to attach, or HERROR_NONE if we are attaching a CERROR.

ns_cmdq_cmd_sync_error_delivered

The CMD_SYNC can fault if an ATC Invalidate timeout occurs. This is the CMD_SYNC delivering that fault. Fields:

cerror enum

The CERROR to attach.

cmd_id unsigned int

Command ID of the CMD_SYNC.

ns_cmdq_ecmdq_enack

Trace the changes to ENACK. Fields:

cerror enum

The error being exposed, if any.

cmd_queue unsigned int

The top bits indicate the CMDQ id. These bits are attached to all cmd_ids generated for this queue. The bits [63:44] contain the queue ID. [63:62] 0: s-ssd, 1: reserved, 2: ns-ssd. 3: rl-ssd [61] is ECMDQ [59:52] QCP index [51:44] ECMDQ index in QCP.

enack bool

The value of ENACK.

forced_error bool

The queue was enabled with ERR != ERRACK and so the implementation chooses to trigger the error reporting mechanism.

ns_cmdq_issue

A command is actually being executed. Fields:

cmd_id unsigned int

Command ID of the command being executed.

cons unsigned int

CONS of the command.

what string

What is being executed.

ns_cmdq_pointers_state

The command queue pointers. Fields:

cmd_queue unsigned int

The top bits indicate the CMDQ id. These bits are attached to all cmd_ids generated for this queue. The bits [63:44] contain the queue ID. [63:62] 0: s-ssd, 1: reserved, 2: ns-ssd. 3: rl-ssd [61] is ECMDQ [59:52] QCP index [51:44] ECMDQ index in QCP.

disable_fetch bool

The model has decided that as you programmed PROD/CONS inconsistently that it would not fetch as it has no confidence as to which are commands and which are not.

q_issuable_and_issued_cons unsigned int

The CONS pointer of the region that is issuable and/or issued (including wrap).

q_issuable_and_issued_prod unsigned int

The PROD pointer of the region that is issuable and/or issued (including wrap).

q_programmer_cons unsigned int

The CONS pointer as seen by the programmer (including wrap).

q_programmer_prod unsigned int

The PROD pointer as seen by the programmer (including wrap).

q_unfetched_cons unsigned int

The CONS pointer of the region unfetched (including wrap).

q_unfetched_prod unsigned int

The PROD pointer of the region unfetched (including wrap).

ns_cmdq_state

A description of what the CMDQ can do now. Fields:

can_cerror bool

There is a CERROR that can be recognised.

can_cmd_sync bool

There is a CMD_SYNC that can be recognised.

can_fetch bool

The SMMU can fetch.

can_herror bool

There is an HERROR that can be recognised.

can_issue bool

There are fetched commands that are eligible to be issued.

can_update_cons bool

The programmer view of CMDQ_CONS can be moved to say some have been consumed.

cmd_queue unsigned int

The top bits indicate the CMDQ id. These bits are attached to all cmd_ids generated for this queue. The bits [63:44] contain the queue ID. [63:62] 0: s-ssd, 1: reserved, 2: ns-ssd, 3: rl-ssd [61] is ECMDQ [59:52] QCP index [51:44] ECMDQ index in QCP.

cmd_sync enum

Why a CMD_SYNC cannot be issued.

cmdqen bool

CMDQEN, when enabled then the CMDQ can fetch commands.

cmdqen_1_to_0_unacknowledged bool

CMDQEN has been set to 0, but we have yet to acknowledge it.

current-real-cerror enum

The current CERROR that would be visible to the programmer if there was an error.

fetch-disabled_due_to_programmer_error bool

The PROD/CONS pointer was inconsistent and the model decided to stop processing commands.

fetch-number_of_unfetched_commands unsigned int

The number of commands that have yet to be fetched from memory, or ~0u if CMDQEN == 0.

fetch-state enum

The state of the fetch state machine.

fetch-too_many_commands_outstanding_so_not_fetching_any_more bool

If there are too many issued commands, or pending to be issued then we won't fetch any more.

issue-number_of_issuable_commands unsigned int

The number of commands that have been fetched but not yet issued.

issue-number_of_unissuable_commands unsigned int

This might include commands behind an illegal command, or behind a CMD_SYNC. Under some circumstances this can include a command representing an external abort.

issue-state enum

The internal issued state machine state.

outstanding_fetches_since_reset unsigned int

The number of outstanding fetches.

synthetic_sync_can_be_completed bool

The synthetic sync that is outstanding can be completed.

waiting_for_synthetic_sync bool

There is a synthetic sync that is outstanding.

ns_eventq_adding_to_cmd_sync_prior_set

The event created by the specified transaction id must become visible to the programmer before the CMD_SYNC can complete. Fields:

trans_id unsigned int

The transaction id that caused the event.

ns_eventq_auto_retry_stalled_transaction

A transaction that was stalled but not yet reported on the Event Queue was retried. The retry can happen if the transaction didn't write because it was de-duplicated behind a reported stalling transaction and that stalling transaction was CMD_RESUMEd, or if the queue was unwritable and the queue is now writable. Fields:

trans_id unsigned int

Transaction id of the auto-retried transaction.

why enum

Why the transaction retried.

ns_eventq_cmd_sync_unhazarded

The CMD_SYNC has been unhazarded as the prior set is empty and as much of it as required is visible in the programmers' view. Fields:

prod_incl_wrap unsigned int

The prod index including wrap bit that was required to become visible.

prog_prod_incl_wrap unsigned int

The current programmer-visible prod including wrap bit.

ns_eventq_eventqen_and_eventqenack

Trace the values of EVENTQEN and EVENTQENACK as they change. Fields:

about enum

Some more information about the queue state.

cons_incl_wrap unsigned int

The CONS pointer (including wrap bits).

eventqen bool

EVENTQEN.

eventqenack bool

EVENTQENACK.

prod_incl_wrap unsigned int

The PROD pointer (including wrap bits).

ns_eventq_external_abort

A particular eventq record got an external abort. The event will subsequently appear in the ns/s_eventq_lost_event_records trace. Fields:

prod unsigned int

The prod pointer (excluding wrap bit) of the event record that aborted.

trans_id unsigned int

The transaction id that produced the event record that aborted.

ns_eventq_lost_event_records

Event records that are lost and never record appear on this trace. Fields:

dword0 unsigned int

Double word 0 of event.

dword1 unsigned int

Double word 1 of event.

dword2 unsigned int

Double word 2 of event.

dword3 unsigned int

Double word 3 of event.

event enum

Event code that was thrown away.

prod unsigned int

The PROD index of the event if appropriate or ~0u otherwise.

reason enum

The reason the record was lost.

trans_id unsigned int

The transaction id that this corresponds to.

ns_eventq_overflow_acknowledged

When events are lost then they appear on the eventq_lost_event_records_* trace. This will activate the overflow flag in the corresponding SMMU_s_EVENTQ_PROD register. Software can then acknowledge that flag using the overflow acknowledge flag in SMMU_s_EVENTQ_CONS. Fields:

bad bool

If true then the programmer has attempted to acknowledge an overflow condition that didn't exist.

count_of_events_lost_due_to_overflow unsigned int

This is the number of events that were lost because the event queue was full. This does not include any lost because of an EVENTQ_ABT_ERR.

ns_eventq_pending_event_records

Event records that are pending to be dealt with by the eventq state machine. A specific event can be traced multiple times if it becomes part of the prior set of transactions of a CMD_SYNC that has to complete before the CMD_SYNC is allowed to be completed. Fields:

dword0 unsigned int

Double word 0 of event.

dword1 unsigned int

Double word 1 of event.

dword2 unsigned int

Double word 2 of event.

dword3 unsigned int

Double word 3 of event.

event enum

Event code.

is_prior_set bool

The transaction forms part of the prior set of a CMD_SYNC that have to become visible (or thrown away if necessary) before the CMD_SYNC can complete.

trans_id unsigned int

The transaction id that this corresponds to.

ns_eventq_prod_cons_updated

This records when the programmer visible PROD/CONS pointer is updated. More can be visible in memory than are advertised in the programmer-visible PROD pointer. This also traces the latest PROD that could have been advertised. Fields:

mem_prod unsigned int

The PROD pointer value for all records that are visible in the memory.

num_events_available unsigned int

The number of events available to the programmer.

prog_cons unsigned int

The programmer-visible CONS pointer.

prog_prod unsigned int

The programmer-visible PROD pointer.

what enum

What changed?.

ns_eventq_raising_eventq_abt_err

EVENTQ_ABT_ERR can now be raised.

ns_eventq_removing_from_cmd_sync_prior_set

The event captured by a CMD_SYNC and created by the specified transaction id is now in memory. Fields:

prod unsigned int

The prod index (no wrap bit) that must become visible to the programmer to see this record. If not head_of_line then the record will not become visible and this field is meaningless.

reason enum

The reason the event was removed.

trans_id unsigned int

The transaction id that caused the event.

ns_eventq_stashing_unreported_stalled_transaction_for_auto_retry

A stall event record would have been lost, instead it is stashed away to auto-retry when it can. Fields:

dword0 unsigned int

Double word 0 of event.

dword1 unsigned int

Double word 1 of event.

dword2 unsigned int

Double word 2 of event.

dword3 unsigned int

Double word 3 of event.

event enum

Event code that was thrown away.

reason enum

The reason the record was lost.

trans_id unsigned int

The transaction id that this corresponds to.

ns_eventq_write_event

An event is committed to being written. Fields:

dword0 unsigned int

Double word 0 of event.

dword1 unsigned int

Double word 1 of event.

dword2 unsigned int

Double word 2 of event.

dword3 unsigned int

Double word 3 of event.

event enum

Event code that was thrown away.

prod unsigned int

The PROD pointer (without wrap) that this event will be written to.

stall_event bool

Describes an event for a stalled transaction.

trans_id unsigned int

Transaction ID that generated this event.

ns_eventq_write_event_visible_in_memory

An event is now visible in memory. Fields:

abort bool

The write aborted.

dword0 unsigned int

Double word 0 of event.

dword1 unsigned int

Double word 1 of event.

dword2 unsigned int

Double word 2 of event.

dword3 unsigned int

Double word 3 of event.

event enum

Event code that was thrown away.

prod unsigned int

The PROD pointer (without wrap) that this event will be written to.

stall_event bool

Describes an event for a stalled transaction.

trans_id unsigned int

Transaction ID that generated this event.

ns_gerror_inverted

A GERROR bit was attempted to be inverted to record a fault. Fields:

already_different_to_gerrorn bool

True if the bit was already different to the corresponding bit in GERRORN and so it was not inverted.

gerrorn unsigned int

GERRORN register value.

interrupt_potentially_sent bool

An interrupt is potentially sent, depending on IRQEN.
GERROR(N).MSI_GERROR_ABT_ERR does not send an interrupt.

new_gerror unsigned int

GERROR register value after any inversion.

which enum

Which bit was attempted to be inverted.

ns_gerrorn_acknowledge

A GERROR was acknowledged by SW. Fields:

which enum

Which GERROR was acknowledged.

pmcg_irq_config

The interrupt configuration of the Performance Monitor Counter Group (PMCG) changed.
Fields:

address unsigned int

Address of the MSI.

data unsigned int

Data payload of the MSI.

irqen bool

SMMU_PMCG_CTRL.IRQEN.

memattr enum

Memory type.

mpam_ns bool

The NS state of the MPAM PARTID and MPAM PMG.

mpam_pmg_and_partid unsigned int

Bits[23:16] are the MPAM PMG. Bits[15:0] are the physical PARTID.

msi_supported bool

Are MSIs supported for this security world.

ns bool

Non-secure bus attribute.

number_of_interrupts_in_flight unsigned int

The number of interrupts that have been committed to be produced or in flight.

pmcg_index unsigned int

Index of the PMCG.

sh enum

Shareability.

smmu_pmcg_gmpam_Update bool

The SMMU_PMC_GMPAM.Update flag. Only when this is zero are writes predictable.

ssd_ns bool

True if this PMCG controlled by the non-secure world.

pmcg_irqen

A trace of SMMU_PMC_GIRQ_CTRL.IRQEN. Fields:

new_value bool

The new value of IRQEN.

pmcg_index unsigned int

Index of the PMCG.

ssd_ns bool

True if this PMCG controlled by the non-secure world.

pmcg_irqen_ack

A trace of SMMU_PMC_GIRQLACK.IRQEN. Fields:

new_value bool

The new value of IRQEN.

pmcg_index unsigned int

Index of the PMCG.

ssd_ns bool

True if this PMCG controlled by the non-secure world.

pmcg_merging_interrupts

An interrupt was wanted to be generated, but one was already pending so the two were merged together. Fields:

pmcg_index unsigned int

Index of the PMCG.

ssd_ns bool

True if this PMCG controlled by the non-secure world.

pmcg_trigger

A PMCG counter has been triggered. Fields:

NoStreamID bool

True if the transaction is a NoStreamID transaction.

counter_index unsigned int

The index of the counter within the PMCG.

event_id unsigned int

The event id that has been triggered.

ns_event bool

Is the event associated with non-secure state.

pmcg_index unsigned int

Index of the PMCG.

prior_counter_value unsigned int

The Counter value *before* the event has incremented it.

ssd enum

The security state associated with the event.

streamid unsigned int

The StreamID associated with the event, if there is one.

tbu_index unsigned int

The TBU index of the transaction, or ~0u if not applicable.

pmu_active_counter

Traces what active counters are in a PMCG and what StreamIDs it might filter on. Those counters that trace StreamIDs for multiple security states, or those that are not filtered by StreamID, will appear multiple times, once for each security state. All active counters for a PMCG are traced one after another. Fields:

NoStreamID bool

True if NoStreamID transactions will be traced.

begin_streamid unsigned int

The start StreamID to filter on.

counter_index unsigned int

The counter index within the PMCG.

end_incl_streamid unsigned int

The end inclusive StreamID to filter on.

evcnt unsigned int

The current count.

event_id unsigned int

The event ID to filter.

ns bool

Are the StreamIDs non-secure?.

pmcg_index unsigned int

The index of the PMCG that this counter belongs to.

ssd enum

SSD of the StreamID.

tbu_index_to_match unsigned int

The TBU index that must match, or ~0u if no matching applicable.

pmu_all_counters_in_pmcg_became_inactive

The PMCG was tracing some events and now is not tracing any. Fields:

pmcg_index unsigned int

The index of the PMCG.

pmu_capture

For some reason, a capture event occurred. Fields:

pmcg_index unsigned int

The index of the PMCG that the capture occurred on.

why enum

Why did the capture occur?.

pmu_counter_configured_to_use_unsupported_event

An enabled counter was configured to use a unsupported event. Fields:

counter_index unsigned int

The counter index within the PMCG.

event_id unsigned int

The unsupported event id.

pmcg_index unsigned int

The index of the PMCG that this counter belongs to.

pmu_counter_overflowed

A counter in a particular PMCG overflowed. Fields:

already_overflowed bool

True if the overflow flag was already set.

capture bool

True if it captured the other counter values.

counter_index unsigned int

The counter index within the PMCG.

interrupt bool

True if going to attempt to generate an interrupt.

interrupt_action enum

The interrupt action that is going to occur.

pmcg_index unsigned int

The index of the PMCG that this counter belongs to.

ssd_ns bool

The PMCG is controlled by the Non-secure security state.

priq_auto_response

A PRIQ auto response is generated. Fields:

prgindex unsigned int

The PRG Index of the response.

resp enum

The response to send.

streamid unsigned int

The StreamID of the response.

substreamid unsigned int

The PASID/SubstreamID attached to the response, or ~0u if there is none.

trans_id unsigned int

The ID of the PPR that is being lost.

why enum

Why the PPR is being lost.

priq_lost_ppr

The PRIQ is not reporting a PPR into the queue. It may or may not generate an auto-response. Fields:

is_stop_marker bool

The PPR being lost is a Stop Marker.

resp enum

What, if any, auto-response will be.

trans_id unsigned int

The ID of the PPR that is being lost.

why enum

Why the PPR is being lost.

priq_overflow_acking

Indicates that an overflow condition was acknowledged by writing to:-
SMMU_PRIQ_CONS.OVACKFLG. Fields:

new_ovackflg bool

The new value of the SMMU_PRIQ_CONS.OVACKFLG.

priq_overflow_asserting

Indicates that we are toggling the SMMU_PRIQ_PROD.OVFLG because we lost a PRI request due to the PRIQ being full and an existing overflow condition does not already exist. Fields:

new_ovflg bool

The new value of the SMMU_PRIQ_PROD.OVFLG.

trans_id unsigned int

The transaction ID of the PPR that caused the overflow.

priq_received

A PRIQ Request has been received and is queued waiting processing. This does not mean that it has been written to the PRIQ but has been placed in a pending queue awaiting a decision about what to do about it. Fields:

L bool

The Last in the Page Request Group (PRG).

R bool

Read permission is requested.

W bool

Read permission is requested.

prgindex unsigned int

The Page Request Group (PRG) index that is being used to identify this request to the EndPoint.

stop_marker bool

If LRW == 0b100 then this indicates this is a stop marker.

streamid unsigned int

The StreamID this request is associated with.

substreamid unsigned int

The SubstreamID (PASID) this request is associated with, of ~0u if no SubstreamID.

substreamid_P bool

If has a SubstreamID (!= ~0u) then Privileged Mode Requested.

substreamid_X bool

If has a SubstreamID (!= ~0u) then Execution Requested (R must also be 1).

trans_id unsigned int

The transaction ID that identifies this request.

untranslated_address unsigned int

The Untranslated Address that this is requesting.

priq_state

The trace of various fields that indicate the state of the PRIQ. Fields:

cons_incl_wrap unsigned int

The value of SMMU_PRIQ.CONS.RD_and_RD_wrap.

number_of_pprs unsigned int

The number of PPRs as indicated by the CONS/PROD.

number_of_pprs_still_to_deal_with unsigned int

This is the number of PPRs that are currently waiting to either be written to the PRIQ, or auto-responded to.

number_of_priq_writes_in_flight unsigned int

The number of writes to the PRIQ that are currently in flight.

ovackflg bool

The OVACKFLG which if different to OVFLG is used to indicate that the PRIQ overflowed.

ovflg bool

The OVFLG which if different to OVACKFLG is used to indicate that the PRIQ overflowed.

priq_abt_err bool

There is an active SMMU_GERROR{N}.PRIQ_ABT_ERR.

priqen bool

The value of SMMU_CRO.PRIQEN. The *effective* value is 0 if SMMUEN == 0.

priqenack bool

The value of SMMU_CROACK.PRIQEN.

prod_incl_wrap unsigned int

The value of SMMU_PRIQ.CONS.WR_and_WR_wrap.

queue_disabled_due_to_prior_programming_error bool

The queue was disabled as the programmer got CONS/PROD into an inconsistent state. The model will disable the PRIQ until SW disables and re-enables the queue via SMMU_CRO.PRIQEN.

smmuen bool

The value of SMMU_CRO.SMMUEN. If this is 0 then the effective PRIQEN is 0.

smmuenack bool

The value of SMMU_CRO.SMMUEN. If this is 0 then the effective PRIQEN is 0.

table_size_in_elements unsigned int

The size of the table in the number of items it can hold.

priq_write_aborted

A PRIQ write aborted. The PRIQ now goes into an error state and will start auto-responding to PRI requests. Fields:

prod_incl_wrap unsigned int

The PROD including the Wrap bit where we were trying to write.

trans_id unsigned int

The transaction ID of the PPR that aborted.

priq_write_ok

A PRIQ write completed OK. Fields:

prod_incl_wrap unsigned int

The PROD including the Wrap bit where we wrote..

trans_id unsigned int

The transaction ID of the PPR.

priq_write_start

A PRIQ request has been received and is going to be attempt to be written to the queue. Fields:

L bool

The Last in the Page Request Group (PRG).

R bool

Read permission is requested.

W bool

Read permission is requested.

prgindex unsigned int

The Page Request Group (PRG) index that is being used to identify this request to the EndPoint.

prod_incl_wrap unsigned int

PROD position including the wrap bit.

stop_marker bool

If LRW == 0b100 then this indicates this is a stop marker.

streamid unsigned int

The StreamID this request is associated with.

substreamid unsigned int

The SubstreamID (PASID) this request is associated with, of ~0u if no SubstreamID.

substreamid_P bool

If has a SubstreamID (!= ~0u) then Privileged Mode Requested.

substreamid_X bool

If has a SubstreamID (!= ~0u) then Execution Requested (R must also be 1).

trans_id unsigned int

The transaction ID that identifies this request.

untranslated_address unsigned int

The Untranslated Address that this is requesting.

ptw_read

Page Table Walk (read). This is the result of the physical access that the SMMU is making.
Fields:

abort enum

Non-zero if the access aborted/failed.

adomain enum

The actual attributes of the access that was used, after IMP DEF mangling.

aprot enum

The actual attributes of the access that was used, after IMP DEF mangling. PAS[1] (bit[3]), Instruction/Data (bit[2]), PAS[0] (bit[1]), Privileged/User (bit[0]).

data unsigned int

The data fetch if it didn't abort.

desckind enum

The kind of descriptor.

inner_cache enum

The actual attributes of the access that was used, after IMP DEF mangling.

input_address unsigned int

Input address this lookup is for (after TBI normalisation).

mecid unsigned int

The MECID of the TTD fetch.

ns enum

Non-Secure on the bus.

outer_cache enum

The actual attributes of the access that was used, after IMP DEF mangling.

pa_address unsigned int

Physical address.

pas enum

The PAS of the TTD fetch.

ssd enum

SSD of the transaction.

ssd_ns enum

Security state of stream.

streamid unsigned int

The StreamID this is for.

substreamid unsigned int

The SubstreamID this is for.

trans_id unsigned int

Transaction id.

ttb_grain_stage_and_level unsigned int

Which TTB used in bits[31:24]. Which Grain size used in bit[23:16]:- 0 – 4 KiB 1 – 16 KiB 2 – 64 KiB 3 – 64 KiB ARMv8.2 where descriptor bits [15:12] for bits [51:48] of output address 4 – 4 KiB FEAT_LPA2 where address is {bits[9:8], bits[49:12]} 5 – 16 KiB FEAT_LPA2 where address is {bits[9:8], bits[49:14], 2'b00} 6 – 64 KiB ARMv8.2 where descriptor bits [15:12] are ignored Stage in bits[15:8]. Level in bits[7:0], level -1 is 0xF.

ptw_read_st1_invalid_long_descriptor

Page Table Walk (read). The descriptor that the PTW fetched is invalid. Fields:

desckind enum

The kind of descriptor.

input_address unsigned int

Input address this lookup is for (after TBI normalisation).

mecid unsigned int

The MECID of the TTD fetch.

ns enum

Non-Secure on the bus.

pa_address unsigned int

Physical address.

pas enum

The PAS of the TTD fetch.

ssd enum

SSD of the transaction.

ssd_ns enum

Security state of stream.

streamid unsigned int

The StreamID this is for.

substreamid unsigned int

The SubstreamID this is for.

trans_id unsigned int

Transaction id.

ttb_grain_stage_and_level unsigned int

Which TTB used in bits[31:24]. Which Grain size used in bit[23:16]:- 0 – 4 KiB 1 – 16 KiB 2 – 64 KiB 3 – 64 KiB ARMv8.2 where descriptor bits [15:12] for bits [51:48] of output address 4 – 4 KiB FEAT_LPA2 where address is {bits[9:8], bits[49:12]} 5 – 16

KiB FEAT_LPA2 where address is {bits[9:8], bits[49:14], 2'b00} 6 – 64 KiB ARMv8.2 where descriptor bits [15:12] are ignored Stage in bits[15:8]. Level in bits[7:0], level -1 is 0xF.

ptw_read_st1_leaf_long_descriptor

Page Table Walk (read). The descriptor that the PTW fetched is a block or page and this is the decode. Fields:

AF bool

Access Flag.

AMEC enum

Use Primary or Alternative MECID.

AP21 enum

The access permissions.

AttrIndx210 unsigned int

The attribute index into the MAIR0/1. If AIE is implemented then this is the full index AttrIndx[3:0].

DBM bool

Dirty Bit Modifier. This only has meaning if running in AArch64 mode.

NS bool

The encoding is for non-secure if this is a secure fetch.

PIE_Dirty enum

The Dirty bit if S2PIE is in use.

PIE_PIIIndex unsigned int

The S2PIIndex if S2PIE is in use, or 0xFFFF if not.

PXN bool

Privileged eXecute Never.

Protected enum

Is the descriptor producing an AssuredTranslation.

SH10 enum

The shareability.

XN bool

eXecute Never.

contiguous bool

Contiguous hint.

desckind enum

The kind of descriptor.

hwu_pbha unsigned int

Top four bits are appropriate CD.HWU, *bottom bits[62:59] of descriptor. Page Based Hardware Attributes: only valid on a bit-per-bit basis enabled by CD.HWU0/CD.HWU1**.

input_address unsigned int

Input address this lookup is for (after TBI normalisation).

mecid unsigned int

The MECID of the TTD fetch.

nG bool

not Global.

nT bool

SMMUv3.2: If true then do not cache this entry in such a way that it can cause a TLB conflict abort (F_TLB_CONFLICT). The entry must produce a consistent result.

ns enum

Non-Secure on the bus.

output_address unsigned int

Output address.

pa_address unsigned int

Physical address.

pas enum

The PAS of the TTD fetch.

ssd enum

SSD of the transaction.

ssd_ns enum

Security state of stream.

streamid unsigned int

The StreamID this is for.

substreamid unsigned int

The SubstreamID this is for.

trans_id unsigned int

Transaction id.

ttb_grain_stage_and_level unsigned int

Which TTB used in bits[31:24]. Which Grain size used in bit[23:16]:- 0 – 4 KiB 1 – 16 KiB 2 – 64 KiB 3 – 64 KiB ARMv8.2 where descriptor bits [15:12] for bits [51:48] of output address 4 – 4 KiB FEAT_LPA2 where address is {bits[9:8], bits[49:12]} 5 – 16 KiB FEAT_LPA2 where address is {bits[9:8], bits[49:14], 2'b00} 6 – 64 KiB ARMv8.2 where descriptor bits [15:12] are ignored Stage in bits[15:8]. Level in bits[7:0], level -1 is 0xF.

ptw_read_st1_table_long_descriptor

Page Table Walk (read). The descriptor that the PTW fetched is a Table, this decodes the fields. Fields:

AF enum

.

APTable enum

Remove permissions independently of subsequent descriptors.

NSTable bool

The next level table descriptor is forced to non-secure.

PXNTable bool

Force PXN independently of subsequent descriptors.

Protected enum

Is the descriptor capable of producing an AssuredTranslation.

TableAddress unsigned int

Address of the next table.

XNTable bool

Force XN independently of subsequent descriptors.

desckind enum

The kind of descriptor.

input_address unsigned int

Input address this lookup is for (after TBI normalisation).

mecid unsigned int

The MECID of the TTD fetch.

ns enum

Non-Secure on the bus.

pa_address unsigned int

Physical address.

pas enum

The PAS of the TTD fetch.

ssd enum

SSD of the transaction.

ssd_ns enum

Security state of stream.

streamid unsigned int

The StreamID this is for.

substreamid unsigned int

The SubstreamID this is for.

trans_id unsigned int

Transaction id.

ttb_grain_stage_and_level unsigned int

Which TTB used in bits[31:24]. Which Grain size used in bit[23:16]:- 0 – 4 KiB 1 – 16 KiB 2 – 64 KiB 3 – 64 KiB ARMv8.2 where descriptor bits [15:12] for bits [51:48] of output address 4 – 4 KiB FEAT_LPA2 where address is {bits[9:8], bits[49:12]} 5 – 16

KiB FEAT_LPA2 where address is {bits[9:8], bits[49:14], 2'b00} 6 – 64 KiB ARMv8.2 where descriptor bits [15:12] are ignored Stage in bits[15:8]. Level in bits[7:0], level -1 is 0xF.

ptw_read_st2_invalid_descriptor

Page Table Walk (read). The descriptor that the PTW fetched is invalid. Fields:

desckind enum

The kind of descriptor.

input_address unsigned int

Input address this lookup is for (after TBI normalisation).

mecid unsigned int

The MECID of the TTD fetch.

ns enum

Non-Secure on the bus.

pa_address unsigned int

Physical address.

pas enum

The PAS of the TTD fetch.

ssd enum

SSD of the transaction.

ssd_ns enum

Security state of stream.

streamid unsigned int

The StreamID this is for.

substreamid unsigned int

The SubstreamID this is for.

trans_id unsigned int

Transaction id.

tth_grain_stage_and_level unsigned int

Which TTB used in bits[31:24]. Which Grain size used in bit[23:16]:- 0 – 4 KiB 1 – 16 KiB 2 – 64 KiB 3 – 64 KiB ARMv8.2 where descriptor bits [15:12] for bits [51:48] of output address 4 – 4 KiB FEAT_LPA2 where address is {bits[9:8], bits[49:12]} 5 – 16 KiB FEAT_LPA2 where address is {bits[9:8], bits[49:14], 2'b00} 6 – 64 KiB ARMv8.2 where descriptor bits [15:12] are ignored Stage in bits[15:8]. Level in bits[7:0], level -1 is 0xF.

ptw_read_st2_leaf_descriptor

Page Table Walk (read). The descriptor that the PTW fetched is a block or page and this is the decode. Fields:

AF bool

Access Flag.

AMEC enum

Use Primary or Alternative MECID.

AssuredOnly enum

The descriptor is marked as AssuredOnly.

DBM bool

Dirty Bit Modifier. This only has meaning if running in AArch64 mode.

HAP21 enum

The access permissions.

MemAttr3_0 enum

The memory attributes.

NS enum

Whether this descriptor forces NS.

PIE_Dirty enum

The Dirty bit if S2PIE is in use.

PIE_PIIIndex unsigned int

The S2PIIndex if S2PIE is in use, or 0xFFFF if not.

POE_POIndex unsigned int

The S2POIndex if S2POE is in use, or 0xFFFF if not.

SH10 enum

The shareability.

XN enum

eXecute Never.

contiguous bool

Contiguous hint.

desckind enum

The kind of descriptor.

input_address unsigned int

Input address this lookup is for (after TBI normalisation).

mecid unsigned int

The MECID of the TTD fetch.

nT bool

SMMUv3.2: If true then do not cache this entry in such a way that it can cause a TLB conflict abort (F_TLB_CONFLICT). The entry must produce a consistent result.

ns enum

Non-Secure on the bus.

output_address unsigned int

Output address.

pa_address unsigned int

Physical address.

pas enum

The PAS of the TTD fetch.

s2hwu_pbha unsigned int

Top four bits are STE.S2HWU, bottom bits[62:59] of descriptor. *Page Based Hardware Attributes: only valid on a bit-per-bit basis enabled by STE.S2HWU.*

ssd enum

SSD of the transaction.

ssd_ns enum

Security state of stream.

streamid unsigned int

The StreamID this is for.

substreamid unsigned int

The SubstreamID this is for.

trans_id unsigned int

Transaction id.

ttb_grain_stage_and_level unsigned int

Which TTB used in bits[31:24]. Which Grain size used in bit[23:16]:- 0 – 4 KiB 1 – 16 KiB 2 – 64 KiB 3 – 64 KiB ARMv8.2 where descriptor bits [15:12] for bits [51:48] of output address 4 – 4 KiB FEAT_LPA2 where address is {bits[9:8], bits[49:12]} 5 – 16 KiB FEAT_LPA2 where address is {bits[9:8], bits[49:14], 2'b00} 6 – 64 KiB ARMv8.2 where descriptor bits [15:12] are ignored Stage in bits[15:8]. Level in bits[7:0], level -1 is 0xF.

ptw_read_st2_table_descriptor

Page Table Walk (read). The descriptor that the PTW fetched is a Table, this decodes the fields. Fields:

AF enum

.

APTable enum

Remove permissions independently of subsequent descriptors.

PXNTable bool

Force PXN independently of subsequent descriptors.

TableAddress unsigned int

Address of the next table.

XNTable bool

Force XN independently of subsequent descriptors.

desckind enum

The kind of descriptor.

input_address unsigned int

Input address this lookup is for (after TBI normalisation).

mecid unsigned int

The MECID of the TTD fetch.

ns enum

Non-Secure on the bus.

pa_address unsigned int

Physical address.

pas enum

The PAS of the TTD fetch.

ssd enum

SSD of the transaction.

ssd_ns enum

Security state of stream.

streamid unsigned int

The StreamID this is for.

substreamid unsigned int

The SubstreamID this is for.

trans_id unsigned int

Transaction id.

ttb_grain_stage_and_level unsigned int

Which TTB used in bits[31:24]. Which Grain size used in bit[23:16]:- 0 – 4 KiB 1 – 16 KiB 2 – 64 KiB 3 – 64 KiB ARMv8.2 where descriptor bits [15:12] for bits [51:48] of output address 4 – 4 KiB FEAT_LPA2 where address is {bits[9:8], bits[49:12]} 5 – 16 KiB FEAT_LPA2 where address is {bits[9:8], bits[49:14], 2'b00} 6 – 64 KiB ARMv8.2 where descriptor bits [15:12] are ignored Stage in bits[15:8]. Level in bits[7:0], level -1 is 0xF.

raw_register_end_read

The raw register read transaction. This is the transaction as directed to the register port.
Fields:

address unsigned int

The address of the transaction.

id unsigned int

Id of this transaction.

ok bool

Was the access OK? The bus response will always be OK, but is the register access reasonable.

pas enum

The PAS of the transaction.

read_data unsigned int

The data read.

raw_register_end_write

The raw register write transaction. This is the transaction as directed to the register port.
Fields:

address unsigned int

The address of the transaction.

id unsigned int

Id of this transaction.

ok bool

Was the access OK? The bus response will always be OK, but is the register access reasonable.

pas enum

The PAS of the transaction.

raw_register_start_read

The raw register read transaction. This is the transaction as directed to the register port.
Fields:

address unsigned int

Address of transaction.

id unsigned int

Id of this transaction. Top bit is set if is a debug transaction.

ns enum

Is the transaction non-secure?.

number_of_beats unsigned int

Number of beats.

pas enum

The PAS of the transaction.

size_of_beat_in_bytes unsigned int

Size of each beat in bytes.

raw_register_start_write

The raw register write transaction. This is the transaction as directed to the register port.
Fields:

address unsigned int

Address of transaction.

id unsigned int

Id of this transaction. Top bit is set if is a debug transaction.

ns enum

Is the transaction non-secure?.

number_of_beats unsigned int

Number of beats.

pas enum

The PAS of the transaction.

size_of_beat_in_bytes unsigned int

Size of each beat in bytes.

write_data unsigned int

The data to write.

register_disallowed_read_string

A text representation of the read of a register that was disallowed. Fields:

out string

The text description of the register value read.

register_disallowed_write_string

A text representation of the write of a register write that was disallowed. Fields:

in string

The text description of the register value written.

register_read_reserved

A text representation of an access to a register address that is reserved. Fields:

in string

The text description of the register value.

register_read_string

A text representation of the read of a register. Fields:

out string

The text description of the register value read.

register_write_reserved

A text representation of an access to a register address that is reserved or a write.to a **RES0** field in a register. Fields:

in string

The text description of the register value.

register_write_string

A text representation of the write of a register. Fields:

in string

The text description of the register value written.

rl_priq_auto_response

A PRIQ auto response is generated. Fields:

prgindex unsigned int

The PRG Index of the response.

resp enum

The response to send.

streamid unsigned int

The StreamID of the response.

substreamid unsigned int

The PASID/SubstreamID attached to the response, or ~0u if there is none.

trans_id unsigned int

The ID of the PPR that is being lost.

why enum

Why the PPR is being lost.

rl_priq_lost_ppr

The PRIQ is not reporting a PPR into the queue. It may or may not generate an auto-response. Fields:

is_stop_marker bool

The PPR being lost is a Stop Marker.

resp enum

What, if any, auto-response will be.

trans_id unsigned int

The ID of the PPR that is being lost.

why enum

Why the PPR is being lost.

rl_priq_received

A PRIQ Request has been received and is queued waiting processing. This does not mean that it has been written to the PRIQ but has been placed in a pending queue awaiting a decision about what to do about it. Fields:

L bool

The Last in the Page Request Group (PRG).

R bool

Read permission is requested.

W bool

Read permission is requested.

prgindex unsigned int

The Page Request Group (PRG) index that is being used to identify this request to the EndPoint.

stop_marker bool

If LRW == 0b100 then this indicates this is a stop marker.

streamid unsigned int

The StreamID this request is associated with.

substreamid unsigned int

The SubstreamID (PASID) this request is associated with, of ~0u if no SubstreamID.

substreamid_P bool

If has a SubstreamID (!= ~0u) then Privileged Mode Requested.

substreamid_X bool

If has a SubstreamID (!= ~0u) then Execution Requested (R must also be 1).

trans_id unsigned int

The transaction ID that identifies this request.

untranslated_address unsigned int

The Untranslated Address that this is requesting.

s_cmd_sync_completed_irq

“Secure” CMD_SYNC completed its IRQ action. Fields:

cmd_id unsigned int

The ID of the command being issued.

cons_incl_wrap unsigned int

CONS pointer of CMD_SYNC.

is_synthetic bool

This is a synthetic sync. The cons_incl_wrap is 0xFFFF'ffff.

s_cmd_sync_completed_no_action

Secure CMD_SYNC completed, there was no CMD_SYNC action (SIG_NONE). Fields:

cmd_id unsigned int

The ID of the command being issued.

cons_incl_wrap unsigned int

CONS pointer of CMD_SYNC.

is_synthetic bool

This is a synthetic sync. The cons_incl_wrap is 0xFFFF'ffff.

s_cmd_sync_completed_sev

Secure CMD_SYNC completed its action of SIG_SEV. Fields:

cmd_id unsigned int

The ID of the command being issued.

cons_incl_wrap unsigned int

CONS pointer of CMD_SYNC.

is_synthetic bool

This is a synthetic sync. The cons_incl_wrap is 0xFFFF'ffff.

s_cmd_sync_completed_with_error

Secure CMD_SYNC completed, however, there was an error associated with the completion of the CMD_SYNC. Fields:

error enum

The error associated with the CMD_SYNC.

cmd_id unsigned int

The ID of the command being issued.

cons_incl_wrap unsigned int

CONS pointer of CMD_SYNC.

is_synthetic bool

This is a synthetic sync. The cons_incl_wrap is 0xFFFF'ffff.

s_cmd_sync_issued

Secure CMD_SYNC has now completed issuing and we now know if something is hazarding the CMD_SYNC from completing. Fields:

cmd_id unsigned int

The ID of the command being issued.

cons_incl_wrap unsigned int

CONS pointer of CMD_SYNC.

hazarded bool

The CMD_SYNC is currently hazarded from completing.

is_synthetic bool

This is a synthetic sync. The cons_incl_wrap is 0xFFFF'ffff.

s_cmd_sync_issuing

Secure CMD_SYNC is starting to issue, we are going to start accounting for which things have to be part of the prior set of things that have to complete before the CMD_SYNC can complete. Fields:

cmd_id unsigned int

The ID of the command being issued.

cons_incl_wrap unsigned int

CONS pointer of CMD_SYNC.

is_synthetic bool

This is a synthetic sync. The cons_incl_wrap is 0xFFFF'ffff.

s_cmd_sync_starting_completion_action_irq

Secure CMD_SYNC starting IRQ completion action. Fields:

cmd_id unsigned int

The ID of the command being issued.

cons_incl_wrap unsigned int

CONS pointer of CMD_SYNC.

is_synthetic bool

This is a synthetic sync. The cons_incl_wrap is 0xFFFF'ffff.

msiaddress unsigned int

The MSI address to use (or 0 if not sending an MSI).

msidata unsigned int

The MSI data to send.

s_cmd_sync_starting_completion_action_sev

Secure CMD_SYNC starting its completion action of SIG_SEV. Fields:

cmd_id unsigned int

The ID of the command being issued.

cons_incl_wrap unsigned int

CONS pointer of CMD_SYNC.

is_synthetic bool

This is a synthetic sync. The cons_incl_wrap is 0xFFFF'ffff.

s_cmdq_cmd_sync_error

The CMD_SYNC can fault if an ATC Invalidate timeout occurs. The timeout will be attached to the next CMD_SYNC on the queue on which the invalidate was launched. Fields:

cerror enum

The CERROR to attach.

cmd_queue unsigned int

The top bits indicate the CMDQ id. These bits are attached to all cmd_ids generated for this queue. The bits [63:44] contain the queue ID. [63:62] 0: s-ssd, 1: reserved, 2: ns-ssd. 3: rl-ssd [61] is ECMDQ [59:52] QCP index [51:44] ECMDQ index in QCP.

herror enum

The HERROR to attach, or HERROR_NONE if we are attaching a CERROR.

s_cmdq_cmd_sync_error_delivered

The CMD_SYNC can fault if an ATC Invalidate timeout occurs. This is the CMD_SYNC delivering that fault. Fields:

cerror enum

The CERROR to attach.

cmd_id unsigned int

Command ID of the CMD_SYNC.

s_cmdq_ecmdq_enack

Trace the changes to ENACK. Fields:

error enum

The error being exposed, if any.

cmd_queue unsigned int

The top bits indicate the CMDQ id. These bits are attached to all cmd_ids generated for this queue. The bits [63:44] contain the queue ID. [63:62] 0: s-ssd, 1: reserved, 2: ns-ssd. 3: rl-ssd [61] is ECMDQ [59:52] QCP index [51:44] ECMDQ index in QCP.

enack bool

The value of ENACK.

forced_error bool

The queue was enabled with ERR != ERRACK and so the implementation chooses to trigger the error reporting mechanism.

s_cmdq_issue

A command is actually being executed. Fields:

cmd_id unsigned int

Command ID of the command being executed.

cons unsigned int

CONS of the command.

what string

What is being executed.

s_cmdq_pointers_state

The command queue pointers. Fields:

cmd_queue unsigned int

The top bits indicate the CMDQ id. These bits are attached to all cmd_ids generated for this queue. The bits [63:44] contain the queue ID. [63:62] 0: s-ssd, 1: reserved, 2: ns-ssd. 3: rl-ssd [61] is ECMDQ [59:52] QCP index [51:44] ECMDQ index in QCP.

disable_fetch bool

The model has decided that as you programmed PROD/CONS inconsistently that it would not fetch as it has no confidence as to which are commands and which are not.

q_issuable_and_issued_cons unsigned int

The CONS pointer of the region that is issuable and/or issued (including wrap).

q_issuable_and_issued_prod unsigned int

The PROD pointer of the region that is issuable and/or issued (including wrap).

q_programmer_cons unsigned int

The CONS pointer as seen by the programmer (including wrap).

q_programmer_prod unsigned int

The PROD pointer as seen by the programmer (including wrap).

q_unfetched_cons unsigned int

The CONS pointer of the region unfetched (including wrap).

q_unfetched_prod unsigned int

The PROD pointer of the region unfetched (including wrap).

s_cmdq_state

A description of what the CMDQ can do now. Fields:

can_cerror bool

There is a CERROR that can be recognised.

can_cmd_sync bool

There is a CMD_SYNC that can be recognised.

can_fetch bool

The SMMU can fetch.

can_herror bool

There is an HERROR that can be recognised.

can_issue bool

There are fetched commands that are eligible to be issued.

can_update_cons bool

The programmer view of CMDQ_CONS can be moved to say some have been consumed.

cmd_queue unsigned int

The top bits indicate the CMDQ id. These bits are attached to all cmd_ids generated for this queue. The bits [63:44] contain the queue ID. [63:62] 0: s-ssd, 1: reserved, 2: ns-ssd. 3: rl-ssd [61] is ECMDQ [59:52] QCP index [51:44] ECMDQ index in QCP.

cmd_sync enum

Why a CMD_SYNC cannot be issued.

cmdqen bool

CMDQEN, when enabled then the CMDQ can fetch commands.

cmdqen_1_to_0_unacknowledged bool

CMDQEN has been set to 0, but we have yet to acknowledge it.

current-real-cerror enum

The current CERROR that would be visible to the programmer if there was an error.

fetch-disabled_due_to_programmer_error bool

The PROD/CONS pointer was inconsistent and the model decided to stop processing commands.

fetch-number_of_unfetched_commands unsigned int

The number of commands that have yet to be fetched from memory, or ~0u if CMDQEN == 0.

fetch-state enum

The state of the fetch state machine.

fetch-too_many_commands_outstanding_so_not_fetching_any_more bool

If there are too many issued commands, or pending to be issued then we won't fetch any more.

issue-number_of_issuable_commands unsigned int

The number of commands that have been fetched but not yet issued.

issue-number_of_unissuable_commands unsigned int

This might include commands behind an illegal command, or behind a CMD_SYNC. Under some circumstances this can include a command representing an external abort.

issue-state enum

The internal issued state machine state.

outstanding_fetches_since_reset unsigned int

The number of outstanding fetches.

synthetic_sync_can_be_completed bool

The synthetic sync that is outstanding can be completed.

waiting_for_synthetic_sync bool

There is a synthetic sync that is outstanding.

s_eventq_adding_to_cmd_sync_prior_set

The event created by the specified transaction id must become visible to the programmer before the CMD_SYNC can complete. Fields:

trans_id unsigned int

The transaction id that caused the event.

s_eventq_auto_retry_stalled_transaction

A transaction that was stalled but not yet reported on the Event Queue was retried. The retry can happen if the transaction didn't write because it was de-duplicated behind a reported stalling transaction and that stalling transaction was CMD_RESUMEd, or if the queue was unwritable and the queue is now writable. Fields:

trans_id unsigned int

Transaction id of the auto-retried transaction.

why enum

Why the transaction retried.

s_eventq_cmd_sync_unhazardded

The CMD_SYNC has been unhazardded as the prior set is empty and as much of it as required is visible in the programmers' view. Fields:

prod_incl_wrap unsigned int

The prod index including wrap bit that was required to become visible.

prog_prod_incl_wrap unsigned int

The current programmer-visible prod including wrap bit.

s_eventq_eventqen_and_eventqenack

Trace the values of EVENTQEN and EVENTQENACK as they change. Fields:

about enum

Some more information about the queue state.

cons_incl_wrap unsigned int

The CONS pointer (including wrap bits).

eventqen bool

EVENTQEN.

eventqenack bool

EVENTQENACK.

prod_incl_wrap unsigned int

The PROD pointer (including wrap bits).

s_eventq_external_abort

A particular eventq record got an external abort. The event will subsequently appear in the ns/s_eventq_lost_event_records trace. Fields:

prod unsigned int

The prod pointer (excluding wrap bit) of the event record that aborted.

trans_id unsigned int

The transaction id that produced the event record that aborted.

s_eventq_lost_event_records

Event records that are lost and never record appear on this trace. Fields:

dword0 unsigned int

Double word 0 of event.

dword1 unsigned int

Double word 1 of event.

dword2 unsigned int

Double word 2 of event.

dword3 unsigned int

Double word 3 of event.

event enum

Event code that was thrown away.

prod unsigned int

The PROD index of the event if appropriate or ~0u otherwise.

reason enum

The reason the record was lost.

trans_id unsigned int

The transaction id that this corresponds to.

s_eventq_overflow_acknowledged

When events are lost then they appear on the eventq_lost_event_records_* trace. This will activate the overflow flag in the corresponding SMMU_s_EVENTQ_PROD register. Software can then acknowledge that flag using the overflow acknowledge flag in SMMU_s_EVENTQ_CONS. Fields:

bad bool

If true then the programmer has attempted to acknowledge an overflow condition that didn't exist.

count_of_events_lost_due_to_overflow unsigned int

This is the number of events that were lost because the event queue was full. This does not include any lost because of an EVENTQ_ABT_ERR.

s_eventq_pending_event_records

Event records that are pending to be dealt with by the eventq state machine. A specific event can be traced multiple times if it becomes part of the prior set of transactions of a CMD_SYNC that has to complete before the CMD_SYNC is allowed to be completed. Fields:

dword0 unsigned int

Double word 0 of event.

dword1 unsigned int

Double word 1 of event.

dword2 unsigned int

Double word 2 of event.

dword3 unsigned int

Double word 3 of event.

event enum

Event code.

is_prior_set bool

The transaction forms part of the prior set of a CMD_SYNC that have to become visible (or thrown away if necessary) before the CMD_SYNC can complete.

trans_id unsigned int

The transaction id that this corresponds to.

s_eventq_prod_cons_updated

This records when the programmer visible PROD/CONS pointer is updated. More can be visible in memory than are advertised in the programmer-visible PROD pointer. This also traces the latest PROD that could have been advertised. Fields:

mem_prod unsigned int

The PROD pointer value for all records that are visible in the memory.

num_events_available unsigned int

The number of events available to the programmer.

prog_cons unsigned int

The programmer-visible CONS pointer.

prog_prod unsigned int

The programmer-visible PROD pointer.

what enum

What changed?.

s_eventq_raising_eventq_abt_err

EVENTQ_ABT_ERR can now be raised.

s_eventq_removing_from_cmd_sync_prior_set

The event captured by a CMD_SYNC and created by the specified transaction id is now in memory. Fields:

prod unsigned int

The prod index (no wrap bit) that must become visible to the programmer to see this record. If not head_of_line then the record will not become visible and this field is meaningless.

reason enum

The reason the event was removed.

trans_id unsigned int

The transaction id that caused the event.

s_eventq_stashing_unreported_stalled_transaction_for_auto_retry

A stall event record would have been lost, instead it is stashed away to auto-retry when it can. Fields:

dword0 unsigned int

Double word 0 of event.

dword1 unsigned int

Double word 1 of event.

dword2 unsigned int

Double word 2 of event.

dword3 unsigned int

Double word 3 of event.

event enum

Event code that was thrown away.

reason enum

The reason the record was lost.

trans_id unsigned int

The transaction id that this corresponds to.

s_eventq_write_event

An event is committed to being written. Fields:

dword0 unsigned int

Double word 0 of event.

dword1 unsigned int

Double word 1 of event.

dword2 unsigned int

Double word 2 of event.

dword3 unsigned int

Double word 3 of event.

event enum

Event code that was thrown away.

prod unsigned int

The PROD pointer (without wrap) that this event will be written to.

stall_event bool

Describes an event for a stalled transaction.

trans_id unsigned int

Transaction ID that generated this event.

s_eventq_write_event_visible_in_memory

An event is now visible in memory. Fields:

abort bool

The write aborted.

dword0 unsigned int

Double word 0 of event.

dword1 unsigned int

Double word 1 of event.

dword2 unsigned int

Double word 2 of event.

dword3 unsigned int

Double word 3 of event.

event enum

Event code that was thrown away.

prod unsigned int

The PROD pointer (without wrap) that this event will be written to.

stall_event bool

Describes an event for a stalled transaction.

trans_id unsigned int

Transaction ID that generated this event.

s_gerror_inverted

A GERROR bit was attempted to be inverted to record a fault. Fields:

already_different_to_gerrorn bool

True if the bit was already different to the corresponding bit in GERRORN and so it was not inverted.

gerrorn unsigned int

GERRORN register value.

interrupt_potentially_sent bool

An interrupt is potentially sent, depending on IRQEN.
GERROR(N).MSI_GERROR_ABT_ERR does not send an interrupt.

new_gerror unsigned int

GERROR register value after any inversion.

which enum

Which bit was attempted to be inverted.

s_gerrorn_acknowledge

A GERROR was acknowledged by SW. Fields:

which enum

Which GERROR was acknowledged.

s_priq_auto_response

A PRIQ auto response is generated. Fields:

prgindex unsigned int

The PRG Index of the response.

resp enum

The response to send.

streamid unsigned int

The StreamID of the response.

substreamid unsigned int

The PASID/SubstreamID attached to the response, or ~0u if there is none.

trans_id unsigned int

The ID of the PPR that is being lost.

why enum

Why the PPR is being lost.

s_priq_lost_ppr

The PRIQ is not reporting a PPR into the queue. It may or may not generate an auto-response. Fields:

is_stop_marker bool

The PPR being lost is a Stop Marker.

resp enum

What, if any, auto-response will be.

trans_id unsigned int

The ID of the PPR that is being lost.

why enum

Why the PPR is being lost.

s_priq_received

A PRIQ Request has been received and is queued waiting processing. This does not mean that it has been written to the PRIQ but has been placed in a pending queue awaiting a decision about what to do about it. Fields:

L bool

The Last in the Page Request Group (PRG).

R bool

Read permission is requested.

W bool

Read permission is requested.

prgindex unsigned int

The Page Request Group (PRG) index that is being used to identify this request to the EndPoint.

stop_marker bool

If LRW == 0b100 then this indicates this is a stop marker.

streamid unsigned int

The StreamID this request is associated with.

substreamid unsigned int

The SubstreamID (PASID) this request is associated with, of ~0u if no SubstreamID.

substreamid_P bool

If has a SubstreamID (!= ~0u) then Privileged Mode Requested.

substreamid_X bool

If has a SubstreamID (!= ~0u) then Execution Requested (R must also be 1).

trans_id unsigned int

The transaction ID that identifies this request.

untranslated_address unsigned int

The Untranslated Address that this is requesting.

sec_override

Controls non-secure accesses to some registers. Fields:

value bool

The value of the signal.

sev

Send a SEV. Fields:

why enum

Why the SEV was generated.

smmu_atc_inv

The CMD_ATC_INV command is sent. Fields:

address unsigned int

Untranslated Address to invalidate.

cmd_id unsigned int

Command id.

cons unsigned int

CONS of the command.

global bool

Global flag.

log2_size_in_bytes unsigned int

Log 2 of the size in bytes of the region to invalidate.

ns_number_in_flight unsigned int

Number of ATC invalidates in flight issued from the Non-secure CMDQ.

ns_number_waiting_to_go_out unsigned int

The total number of ATC invalidates waiting to be pushed out from the Non-secure CMDQ.

rl_number_in_flight unsigned int

Number of ATC invalidates in flight issued from the Realm CMDQ.

rl_number_waiting_to_go_out unsigned int

The total number of ATC invalidates waiting to be pushed out from the Realm CMDQ.

s_number_in_flight unsigned int

Number of ATC invalidates in flight issued from the Secure CMDQ.

s_number_waiting_to_go_out unsigned int

The total number of ATC invalidates waiting to be pushed out from the Secure CMDQ.

ssd_of_queue enum

The SSD of the CMDQ.

ssd_of_streamid enum

The SSD of the StreamID.

streamid unsigned int

StreamID in the invalidate command.

substreamid unsigned int

SubstreamID or ~0u if SSValid == false.

smmu_atc_inv_completed

The CMD_ATC_INV command completed. Fields:

cmd_id unsigned int

Command id, if top-bit is set then was issued from the Non-secure CMDQ.

ns_number_in_flight unsigned int

Number of ATC invalidates in flight now this one has completed, that were issued by the Non-secure CMDQ.

ns_number_waiting_to_go_out unsigned int

The total number of ATC invalidates waiting to be pushed out from the Non-secure CMDQ.

response enum

The response.

rl_number_in_flight unsigned int

Number of ATC invalidates in flight now this one has completed, that were issued by the Realm CMDQ.

rl_number_waiting_to_go_out unsigned int

The total number of ATC invalidates waiting to be pushed out from the Realm CMDQ.

s_number_in_flight unsigned int

Number of ATC invalidates in flight now this one has completed, that were issued by the Secure CMDQ.

s_number_waiting_to_go_out unsigned int

The total number of ATC invalidates waiting to be pushed out from the Secure CMDQ.

ssd_of_cmdq enum

The SSD of the CMDQ.

ssd_of_streamid enum

The SSD of the StreamID.

smmu_atc_inv_end

An ATC invalidate request has been started. Fields:

PASID_global bool

If has a SubstreamID (PASID) then if it is 'global' or not.

address unsigned int

The Untranslated Address to use.

log2_size_in_bytes unsigned int

Log2 of the size of the region in bytes to invalidate.

response enum

The response to the ATC invalidate.

ssd enum

The SSD of the stream.

streamid unsigned int

The StreamID to use.

substreamid unsigned int

The SubstreamID to use (or ~0u if not used). This corresponds to the PASID.

smmu_atc_inv_start

An ATC invalidate request has been started. Fields:

PASID_global bool

If has a SubstreamID (PASID) then if it is 'global' or not.

address unsigned int

The Untranslated Address to use.

log2_size_in_bytes unsigned int

Log2 of the size of the region in bytes to invalidate.

ssd enum

The SSD of the stream.

streamid unsigned int

The StreamID to use.

substreamid unsigned int

The SubstreamID to use (or ~0u if not used). This corresponds to the PASID.

smmu_ats_initial

This is the initial ATS request. Fields:

XT bool

The XT bit.

ia unsigned int

Input address.

max_number_of_replies unsigned int

The maximum number of replies allowed to return.

no_write bool

The NW (no write flag) of the ATS request. If clear then the requester is going to do a write.

pasid_execute_requested bool

If there is a PASID (substreamid != ~0u) then this represents the requester asking for execution permissions.

pasid_privileged_mode_requested bool

If there is a PASID (streamid != ~0u) then this represents the requester asking for the privileged mode's permissions.

protected_mode bool

Is the ATS Request protected-mode?.

source_cxl bool

Does the ATS request have Source.CXL set?.

ssd enum

The SSD of the ATS request.

ssd_ns bool

Incoming SSD is non-secure.

streamid unsigned int

StreamID of the ATS request.

substreamid unsigned int

SubstreamID (which is identical to the PASID) of the ATS request. If no PASID-prefix is sent then this is ~0u.

tbu unsigned int

Translation Buffer Unit number.

smmu_ats_reply_failure

This is an ATS reply indicating failure. Fields:

event enum

Equivalent event number that would have been generated for an equivalent ordinary transaction.

failure enum

What is the failure response code?.

state enum

The transaction state of the successfully ATS request.

smmu_ats_reply_success

This is an ATS reply, typically the SMMU will only return a single response, even if the requester indicated it could accept more replies. NOTE that the SMMU responds with 'success' in some cases when a fault is encountered and RW==0. Fields:

N bool

Non-snooped access. If one then the requester must clear the NoSnoop bit on transactions, unless otherwise enabled in a Function-specific manner.

P bool

Privileged mode. These permissions related to privileged mode.

RWX enum

Read/Write/Execute.

U bool

Untranslated access. If one, and RW !=0 then use UntranslatedAccesses for the allowed accesses by RW(X).

cxl_io bool

The CXL.io response.

inner enum

The inner cacheability attributes to use for TranslatedAccesses.

input_address unsigned int

Input address of the ATS request.

instcfg enum

The STE.INSTCFG field.

outer enum

The outer cacheability attributes to use for TranslatedAccesses.

pas enum

The PAS this mapping corresponds to. This holds the same information as the TE bit for realm streams.

privcfg enum

The STE.PRIVCFG field.

shareability unsigned int

The shareability to use for TranslatedAccesses.

size unsigned int

The size of the region covered by this translation.

state enum

The transaction state of the successfully ATS request.

translated_address unsigned int

If RW!=0 && U != 0, then the Translated Address that a TranslatedAccess can be made with.

smmu_axi_stream_msi

An SMMU generated MSI is directly sent through the axi_stream_msi_m port, typically connected to the GIC port axi_stream_msi_s. Fields:

TDEST unsigned int

Routing information for the data stream, typically identifying the GIC.

TID unsigned int

Data stream identifier for the SMMU.

axi_stream_msi_addr_to_match unsigned int

Current address to match for SMMU-originated MSIs to send out of the axi_stream_msi_m port.

data unsigned int

The MSI sent.

smmu_final_transaction

This is the transaction group request to remap has completed one set of remapping. For ordinary transactions, this represents a bundle of transactions with the same attributes but different addresses within a certain range around the address. If it stalls then it will report through this trace source, stall (stag_if_stalling != ~Ou) and when resume will issue another smmu_initial_transaction as it undergoes remapping again. This trace source can also represent part of the process for ATOS/ATS or finding STE.PPAR for PRI requests that need to be auto-responded to. Fields:

axmmuflow enum

The AxMMUFLOW for this transaction group. stallable – allow the transaction to be stalled if configured. translated-access – is a PCIe Translated Access non-stallable – do not stall transaction even if configured. transfault-flow – do not report for Translation Related Faults, but tell client device to use a page request mechanism instead (for example PRI).

begin_input_address_range unsigned int

The start of the input address range that is size_of_region_in_bytes.

begin_ipa_range unsigned int

The start of the IPA range that is of size_of_region_in_bytes.

begin_output_address_range unsigned int

The start of the output address range that is of size_of_region_in_bytes.

cmo_point enum

The point associated with the CMO, if applicable.

input_address unsigned int

The input address of the transaction group.

input_inner enum

Inner cacheability for the input attributes.

input_outer enum

Outer cacheability for the input attributes.

input_prot enum

The instruction/data, bus-ns/s and privileged/user marking.

input_sh enum

Shareability for the input attributes.

ipa_address unsigned int

The IPA of the transaction.

kind enum

The kind of access this transaction group that this represents.

mecid unsigned int

The MECID of the transaction.

mpam_pmg_and_partid unsigned int

Bits[23:16] are the MPAM PMG. Bits[15:0] are the physical PARTID.

mpam_sp enum

The MPAM_SP of the transaction.

operation enum

The kind of operation that this represents.

output_address unsigned int

The input address of the transaction group.

output_inner enum

Inner cacheability for the output attributes.

output_outer enum

Outer cacheability for the output attributes.

output_prot enum

The instruction/data, bus-ns/s and privileged/user marking.

output_sh enum

Shareability for the output attributes.

output_vmid unsigned int

The output VMID/GBPA.IMPDEF or ~0u if not valid.

size_of_region_in_bytes unsigned int

An imp def size of region for which this translation is valid for.

ssd enum

The SSD of the transaction.

ssd_ns enum

The security state of the transaction.

stag_if_stalling unsigned int

This is the STAG used by the transaction if it is going to stall. It is ~0u if it is not going to stall.

state enum

The final transaction state.

streamid unsigned int

The StreamID of the transaction. ~0ull if NoStreamID.

substreamid unsigned int

The SubstreamID of the transaction or 0xFFFFffff if no SubstreamID.

tbu unsigned int

Translation Buffer Unit number.

trans_id unsigned int

Transaction id. Top bit set if debug transaction.

smmu_gpt_tlb_invalidate

A GPT TLB invalidate has been initiated. Fields:

address unsigned int

The address as it appears in the operation.

encoded_size unsigned int

The size as it is encoded in the operation.

kind enum

The kind of operation this is.

pgs_in_bytes unsigned int

The PGS size in bytes.

size_in_bytes unsigned int

For range operations, the size as it appears in the operation.

source enum

Where the TLBI came from.

state enum

Is the operation well formed.

trans_id unsigned int

The transaction id of this invalidate.

smmu_gpt_tlb_invalidate_complete

The GPT TLB invalidate completed. Fields:

source enum

Where the TLBI came from.

trans_id unsigned int

The transaction id of this invalidate.

smmu_initial_transaction

This is the transaction group request to remap is going to start one set of remapping. For ordinary transactions, this represents a bundle of transactions with the same attributes but different addresses within a certain range around the address. A stalling transaction will report through this trace source when it unstalls. This trace source can also represent part of the process for ATOS/ATS or finding STE.PPAR for PRI requests that need to be auto-responded to. Fields:

XT_and_output_pas_checking enum

The XT bit for PCIe Transactions. This specifies the requested check on the output PAS that the device asked for.

axmmuflow enum

The AxMMUFLOW for this transaction group. stallable – allow the transaction to be stalled if configured. translated-access – is a PCIe Translated Access non-stallable – do not stall transaction even if configured. transfault-flow – do not report for Translation Related Faults, but tell client device to use a page request mechanism instead (for example PRI).

cmo_point enum

The point associated with the CMO, if applicable.

dcmdq_qcp_and_index unsigned int

If this is a DCMDQ fetch being translated then this field indicates the DCMDQ QCP index in bits [23:8] and the index in the page in bits [7:0]. Otherwise this is 0xFFFFffff.

input_address unsigned int

The input address of the transaction group.

input_inner enum

Inner cacheability for the input attributes.

input_outer enum

Outer cacheability for the input attributes.

input_prot enum

The instruction/data, bus-ns/s and privileged/user marking.

input_sh enum

Shareability for the input attributes.

kind enum

The kind of access this transaction group that this represents.

mecid unsigned int

The incoming MECID for NoStreamID transactions. ~0u for all other transactions.

mpam_partid unsigned int

The MPAM_PARTID for NoStreamID transactions. ~0u for all other transactions.

mpam_pmg unsigned int

The MPAM_PMG for NoStreamID transactions. ~0u for all other transactions.

mpam_sp enum

The MPAM_SP for NoStreamID transactions. ~0u for all other transactions.

operation enum

The kind of operation that this represents.

protected_mode enum

The PM bit.

ssd enum

The SSD of the transaction.

ssd_ns enum

The security state of the transaction.

streamid unsigned int

The StreamID of the transaction. ~0ull means NoStreamID.

substreamid unsigned int

The SubstreamID of the transaction or 0xFFFFffff if no SubstreamID.

tbu unsigned int

Translation Buffer Unit number.

trans_id unsigned int

Transaction id. Top bit set if debug transaction.

smmu_normalize_input_transaction

If the input transaction is normalized before being processed then this traceevent will fire. This is usually dependent on parameters of the implementation. Fields:

incoming_inner enum

The inner cacheability attributes.

incoming_is_instruction bool

The incoming transaction is marked as 'instruction'.

incoming_is_privileged bool

The incoming transaction is marked as 'privileged'.

incoming_outer enum

The outer cacheability attributes.

incoming_pas enum

The PAS of the incoming transaction.

incoming_shareability unsigned int

The incoming shareability.

normalized_inner enum

The normalized inner cacheability attributes.

normalized_is_instruction bool

The incoming transaction is marked as 'instruction'.

normalized_is_privileged bool

The incoming transaction is marked as 'privileged'.

normalized_outer enum

The normalized outer cacheability attributes.

normalized_pas enum

The PAS this mapping corresponds to.

normalized_shareability unsigned int

The normalized shareability.

ssd enum

The SSD of the transaction.

streamid unsigned int

The StreamID of the transaction. ~0ull means NoStreamID.

substreamid unsigned int

The SubstreamID of the transaction or 0xFFFFffff if no SubstreamID.

trans_id unsigned int

ID of the original transaction.

smmu_pmusnapshot_ack

Acknowledge the pmusnapshot_req to indicate the snapshot has occurred. This is part of a four-phase handshake. Fields:

pin_index unsigned int

The pin index of the signal, or ~0u if not relevant.

pmcg_index enum

The PMCG index.

value bool

The value of the signal.

smmu_pmusnapshot_ack_overridden

The value of pmusnapshot_ack was overridden (likely due to being in integration mode). Fields:

pin_index unsigned int

The pin index of the signal, or ~0u if not relevant.

pmcg_index enum

The PMCG index.

value bool

The value of the signal.

smmu_pmusnapshot_ack_override_end

The overriding of the pmusnapshot_ack signal has ended. Fields:

pin_index unsigned int

The pin index of the signal, or ~0u if not relevant.

pmcg_index enum

The PMCG index.

smmu_pmusnapshot_req

Take a snap shot of the PMU values as though SMMU_PMCG_CAPR.CAPTURE had been written. This is part of a four-phase handshake. Fields:

pin_index unsigned int

The pin index of the signal, or ~0u if not relevant.

pmcg_index enum

The PMCG index.

value bool

The value of the signal.

smmu_poison_tw_data

Poison data has been returned to a table walk transaction. Fields:

bitmap_of_poison unsigned int

The bitmap of which beats of the transaction where poisoned.

is_cas bool

True if this is a compare-and-swap operation.

number_of_64bit_beats unsigned int

Number of beats this transaction fetched.

paddress unsigned int

Physical address of the table walk transaction.

pas enum

The PAS of the bus transaction.

ras_group_id unsigned int

If non-~0u then is the RAS group that we wish to attach the error to. Otherwise will use the IMP DEF policy.

ras_record_index_in_group unsigned int

If non-~0u then is the RAS record index in the group that we wish to attach the error to. Otherwise will use the IMP DEF policy.

what enum

What table walk was being performed.

smmu_pri_resp

The CMD_PRI_RESP command is queued to be sent to the PCIe system. Fields:

auto_response_trans_id unsigned int

trans_id of PRI request we are auto-responding to, or ~0u if not valid.

cmd_id unsigned int

Command id, or ~0u if not valid.

cons unsigned int

CONS of the command. ~0u if an auto-response.

prgindex unsigned int

PRG Index.

response enum

If has a SubstreamID (PASID) then if it is 'global' or not.

ssd enum

The SSD of the StreamID this corresponds to.

streamid unsigned int

StreamID in the invalidate command.

substreamid unsigned int

SubstreamID or ~0u if SSValid == false.

smmu_pri_resp_nop

The CMD_PRI_RESP command was NOPped. Fields:

cmd_id unsigned int

Command id.

cons unsigned int

CONS of the command.

prgindex unsigned int

PRG Index.

response enum

If has a SubstreamID (PASID) then if it is 'global' or not.

ssd enum

The SSD of the StreamID this corresponds to.

streamid unsigned int

StreamID in the invalidate command.

substreamid unsigned int

SubstreamID or ~0u if SSValid == false.

why enum

Why the CMD_PRI_RESP was NOPped.

smmu_priq_resp_fake_return

A PRIQ Response is posted to the PCIe subsystem and so has no acknowledgement that it is received. However, in the model then we artificially know when the the PRIQ Response has been delivered to the PCIe subsystem, even if the ATC has not yet acted on it. Fields:

prgindex unsigned int

The PRG Index this request corresponds to.

ssd enum

SSD of response.

streamid unsigned int

The StreamID to use.

substreamid unsigned int

The SubstreamID to use (or ~0u if not used). This corresponds to the PASID.

what enum

What happened.

smmu_priq_resp_start

A PRIQ Response has been posted to the PCIe subsystem. As the response is posted then there is no way of knowing when it is received by the EndPoint. Fields:

prgindex unsigned int

The PRG Index this request corresponds to.

response enum

If has a SubstreamID (PASID) then if it is 'global' or not.

ssd enum

SSD of response.

streamid unsigned int

The StreamID to use.

substreamid unsigned int

The SubstreamID to use (or ~0u if not used). This corresponds to the PASID.

smmu_thread_wait_wake

Traces a thread's wait/wake status. Fields:

current_ticks unsigned int

The current tick count of simulated time.

event enum

What is happening to this thread.

thread_index unsigned int

The ID of this thread.

ticks unsigned int

If the event relates to a time then this is held in this field. Otherwise, 0.

stall_transaction

A transaction is about to stall. Fields:

stag unsigned int

STAG.

streamid unsigned int

StreamID.

streamid_ns bool

The StreamID was for the non-secure world.

trans_id unsigned int

Transaction id.

stall_transaction_inhibited_by_STALL_MAX

A transaction is about to stall but the maximum number of transactions have stalled and we can't report this one to the event queue (even if non-full). Fields:

streamid unsigned int

StreamID.

streamid_ns bool

The StreamID was for the non-secure world.

trans_id unsigned int

Transaction id.

stall_transaction_resuming

A stalled transaction is resuming. Fields:

stag unsigned int

STAG if appropriate, or if was inhibited by STALL_MAX then 0xFAFA.

stallresult enum

What the transaction resumed to do.

streamid unsigned int

StreamID.

streamid_ns bool

The StreamID was for the non-secure world.

trans_id unsigned int

Transaction id.

start_ptw_read

Page Table Walk (read). This is the start of the physical access that the SMMU is making. Fields:

adomain enum

The actual attributes of the access that was used, after IMP DEF mangling.

aprot enum

The actual attributes of the access that was used, after IMP DEF mangling. PAS[1] (bit[3]), Instruction/Data (bit[2]), PAS[0] (bit[1]), Privileged/User (bit[0]).

desckind enum

The kind of descriptor.

inner_cache enum

The actual attributes of the access that was used, after IMP DEF mangling.

input_address unsigned int

Input address this lookup is for (after TBI normalisation).

mecid unsigned int

The MECID of the TTD fetch.

mpam_pmg_and_partid unsigned int

Bits[23:16] are the MPAM PMG. Bits[15:0] are the physical PARTID.

ns enum

Non-Secure on the bus.

outer_cache enum

The actual attributes of the access that was used, after IMP DEF mangling.

pa_address unsigned int

Physical address.

pas enum

The PAS of the TTD fetch.

ssd enum

SSD of the transaction.

ssd_ns enum

Security state of stream.

streamid unsigned int

The StreamID this is for.

substreamid unsigned int

The SubstreamID this is for.

trans_id unsigned int

Transaction id.

ttb_grain_stage_and_level unsigned int

Which TTB used in bits[31:24]. Which Grain size used in bit[23:16]:- 0 – 4 KiB 1 – 16 KiB 2 – 64 KiB 3 – 64 KiB ARMv8.2 where descriptor bits [15:12] for bits [51:48] of output address 4 – 4 KiB FEAT_LPA2 where address is {bits[9:8], bits[49:12]} 5 – 16 KiB FEAT_LPA2 where address is {bits[9:8], bits[49:14], 2'b00} 6 – 64 KiB ARMv8.2 where descriptor bits [15:12] are ignored Stage in bits[15:8]. Level in bits[7:0], level -1 is 0xF.

ste_cc.STE_commentary

A commentary from the cache about what is going on. Fields:

output string

The stream output.

ste_cc.STE_entry_invalidated

An entry in the cache has been invalidated. Fields:

entry_id unsigned int

The entry id that is being invalidated.

reason enum

Why the entry is being invalidated.

ste_entry_allocated

An STE entry has been allocated. Fields:

entry_id unsigned int

The id of the cache entry. Top bit is set if is a debug entry.

ns enum

For the non-secure world.

ssd enum

The SSD of the entry.

ste string

A textual description of the STE.

streamid unsigned int

The StreamID of the entry this will match.

sup_btm

The pin is driven. This indicates the system supports BTM. This pin is sampled on the negedge of the reset pin and has to be driven beforehand for the value to take effect. Fields:

value bool

The value of the signal.

sup_cohacc

The pin is driven. This indicates the system supports COHACC. This pin is sampled on the negedge of the reset pin and has to be driven beforehand for the value to take effect. Fields:

value bool

The value of the signal.

sup_httu

The pin is driven. This indicates the system supports HTTU. This pin is sampled on the negedge of the reset pin and has to be driven beforehand for the value to take effect. Fields:

value bool

The value of the signal.

sup_sev

The pin is driven. This indicates the system supports SEV. This pin is sampled on the negedge of the reset pin and has to be driven beforehand for the value to take effect. Fields:

value bool

The value of the signal.

tbu0_reset_in

The reset signal of the TBU0. Fields:

value bool

The value of the signal.

tcu_reset_in

The reset signal. Fields:

value bool

The value of the signal.

tlb_entry_allocated

A TLB entry has been allocated. Fields:

aset unsigned int

The ASET of this entry.

asid unsigned int

ASID if appropriate.

index unsigned int

Index of TLB entry.

input_end_incl_address unsigned int

The end inclusive address of the input range that this matches.

input_start_address unsigned int

The start address of the input range that this matches.

output_end_incl_address unsigned int

The end inclusive address of the output range.

output_start_address unsigned int

The start address of the output range.

scheme enum

The tagging scheme used.

ssd enum

The SSD of the tagging scheme.

ssd_ns enum

Security State.

tbi bool

Was the entry formed using Top Byte Ignore (TBI).

vmid unsigned int

VMID if appropriate.

tlb_info_tlb_entries_overlap

A TLB entry was inserted into the TLB and it overlaps an existing entry. This isn't a problem as it was inserted in such a way that it architecturally works. Fields:

end_address_of_new_entry unsigned int

End address of new entry.

end_address_of_old_entry unsigned int

End address of old entry.

how_inserted enum

How the entry was inserted.

index_of_new_entry unsigned int

Index of new entry.

index_of_old_entry unsigned int

Index of old entry.

start_address_of_new_entry unsigned int

Start address of new entry.

start_address_of_old_entry unsigned int

Start address of old entry.

verbose_commentary

This is a verbose commentary on the translation process the SMMU is performing. Fields:

output string

The stream output.

warning_MSI_EVENTQ_address_out_of_range_of_oas

The MSI address is configured outside of the range of the downstream address bus size.
Fields:

address unsigned int

The untruncated address of the MSI.

warning_MSI_GERROR_address_out_of_range_of_oas

The MSI address is configured outside of the range of the downstream address bus size.
Fields:

address unsigned int

The untruncated address of the MSI.

warning_MSI_HACDBS_PROCESSING_COMPLETE_address_out_of_range_of_oas

The MSI address is configured outside of the range of the downstream address bus size.
Fields:

address unsigned int

The untruncated address of the MSI.

warning_MSI_HDBSS_TABLE_FULL_address_out_of_range_of_oas

The MSI address is configured outside of the range of the downstream address bus size.
Fields:

address unsigned int

The untruncated address of the MSI.

warning_MSI_PRIQ_address_out_of_range_of_oas

The MSI address is configured outside of the range of the downstream address bus size.
Fields:

address unsigned int

The untruncated address of the MSI.

warning_MSI_S_EVENTQ_address_out_of_range_of_oas

The MSI address is configured outside of the range of the downstream address bus size.
Fields:

address unsigned int

The untruncated address of the MSI.

warning_MSI_S_GERROR_address_out_of_range_of_oas

The MSI address is configured outside of the range of the downstream address bus size.
Fields:

address unsigned int

The untruncated address of the MSI.

warning_MSI_S_HACDBS_PROCESSING_COMPLETE_address_out_of_range_of_oas

The MSI address is configured outside of the range of the downstream address bus size.
Fields:

address unsigned int

The untruncated address of the MSI.

warning_MSI_S_HDBSS_TABLE_FULL_address_out_of_range_of_oas

The MSI address is configured outside of the range of the downstream address bus size.
Fields:

address unsigned int

The untruncated address of the MSI.

warning_MSI_pmcg_address_out_of_range_of_oas

The MSI Address of the Performance Monitor Counter Group (PMCG) is out of range of the OAS and so will be silently truncated. Fields:

address unsigned int

The untruncated address of the MSI.

pmcg_index unsigned int

Index of the PMCG.

ssd_ns bool

True if this PMCG controlled by the non-secure world.

warning_discarding_interrupt_EVENTQ_as_irgen_low

Interrupt generation is turned off by SMMU_IRQ_CTRL.EVENTQ_IRQEN.

warning_discarding_interrupt_GERROR_as_irqen_low

Interrupt generation is turned off by SMMU_IRQ_CTRL.GERROR_IRQEN.

warning_discarding_interrupt_HACDBS_PROCESSING_COMPLETE_as_irqen_low

Interrupt generation is turned off by SMMU_IRQ_CTRL.HACDBS_IRQEN.

warning_discarding_interrupt_HDBSS_TABLE_FULL_as_irqen_low

Interrupt generation is turned off by SMMU_IRQ_CTRL.HDBSS_IRQEN.

warning_discarding_interrupt_PRIQ_as_irqen_low

Interrupt generation is turned off by SMMU_IRQ_CTRL.PRIQ_IRQEN.

warning_discarding_interrupt_S_EVENTQ_as_irqen_low

Interrupt generation is turned off by SMMU_S_IRQ_CTRL.EVENTQ_IRQEN.

warning_discarding_interrupt_S_GERROR_as_irqen_low

Interrupt generation is turned off by SMMU_S_IRQ_CTRL.GERROR_IRQEN.

warning_discarding_interrupt_S_HACDBS_PROCESSING_COMPLETE_as_irqen_low

Interrupt generation is turned off by SMMU_IRQ_S_CTRL.HACDBS_IRQEN.

warning_discarding_interrupt_S_HDBSS_TABLE_FULL_as_irqen_low

Interrupt generation is turned off by SMMU_IRQ_S_CTRL.HDBSS_IRQEN.

warning_ns_gerrorn_bad_acknowledge

A GERROR was acknowledged by SW in the GERRORN register that did not have an active error. The result is **UNPREDICTABLE** if this will generate an interrupt or not. Fields:

which enum

Which GERROR was acknowledge when there was no active error.

warning_reg_after_doesnt_match_written_value

A write occurred that tried to set bits in a register, that for one reason or another, failed to get written. Fields:

desc string

The textual description of what happened.

warning_s_gerrorn_bad_acknowledge

A GERROR was acknowledged by SW in the GERRORN register that did not have an active error. The result is **UNPREDICTABLE** if this will generate an interrupt or not. Fields:

which enum

Which GERROR was acknowledge when there was no active error.

what_going_to_do_with_terminated_event

A terminating transaction has produced an event, this tells you what the model is going to do with the event. Fields:

CD.S bool

The CD.S field if available.

S2 bool

The event is related to Stage 2.

STE.S1STALLD bool

The STE.S1STALLD field if available.

STE.S2S bool

The STE.S2S field if available.

aborts bool

The transaction will abort.

axmmuflow enum

The AxMMUFLOW for this transaction group. stallable – allow the transaction to be stalled if configured. translated-access – is a PCIe Translated Access non-stallable – do not stall transaction even if configured. transfault-flow – do not report for Translation Related Faults, but tell client device to use a page request mechanism instead (for example PRI).

is_tr_fault bool

Is a Translation Related fault.

protected_mode bool

The transaction is protected-mode. As such, it cannot stall and will obey the report configuration bits.

reports bool

The transaction will attempt to report.

ssd enum

The SSD of the transaction.

ssd_ns bool

The transaction is classified as SSD non-secure.

supports_stall_model bool

The implementation supports the stall model.

trans_id unsigned int

The transaction id.

why_abort_decision enum

The reason why the transaction aborted/did not abort.

why_report_decision enum

The reason why the transaction reported/did not report.

2.128 MMU_S3

This section describes the trace sources.

ArchMsg.Error.error

These messages are about activity occurring on the SMMU that is considered an error. Messages will only come out here if parameter `all_error_messages_through_trace` is true. `DISPLAY %{output}`. Fields:

output string

The stream output.

ArchMsg.Error.fetch_from_memory_type_not_supporting_httu

A descriptor fetch from an HTTU-enabled translation regime to an unsupported memory type was made. Whilst the fetch itself may succeed, if an update to the descriptor was attempted then it would fail. Fields:

address unsigned int

The address of the descriptor fetch.

desc_inner enum

Inner cacheability of descriptor.

desc_outer enum

Outer cacheability of descriptor.

desc_prot enum

The instruction/data, bus-ns/s and privileged/user marking.

desc_sh enum

Shareability of descriptor.

ssd enum

The SSD of the transaction.

ssd_ns bool

Is the Security State Determination of the transaction non-secure.

stage unsigned int

The stage at which we had a problem.

streamid unsigned int

The StreamID of the transaction.

substreamid unsigned int

The SubstreamID of the transaction (or ~0u if no substreamid).

trans_id unsigned int

Transaction id.

ArchMsg.Error.ns_cmdq

There is an error associated with the command queue. Fields:

cmd_id unsigned int

Command ID of the command in error.

cons unsigned int

CONS of the command.

what string

What is wrong.

ArchMsg.Error.priq_streamid_truncated

The SMMU received a PCIe PRI request with a StreamID) that was larger than that which the SMMU has been configured for. The StreamID that appears in the PRIQ entry will be truncated. Fields:

actual_streamid unsigned int

The actual StreamID that this request has.

sidsize unsigned int

The bit width of the SMMU for StreamIDs, as indicated by SMMU_IDR1.SIDSIZE.

trans_id unsigned int

The transaction ID that identifies this request.

trunc_streamid unsigned int

The truncated StreamID that will appear in the PRIQ entry.

ArchMsg.Error.priq_substreamid_truncated

The SMMU received a PCIe PRI request with a PASID prefix (SubstreamID) that was larger than that which the SMMU has been configured for. The SubstreamID that appears in the PRIQ entry will be truncated. Fields:

actual_substreamid unsigned int

The actual SubstreamID that this request has.

ssidsize unsigned int

The bit width of the SMMU for SubstreamIDs, as indicated by SMMU_IDR1.SSIDSIZE.

trans_id unsigned int

The transaction ID that identifies this request.

trunc_substreamid unsigned int

The truncated SubstreamID that will appear in the PRIQ entry.

ArchMsg.Error.rl_cmdq

There is an error associated with the command queue. Fields:

cmd_id unsigned int

Command ID of the command in error.

cons unsigned int

CONS of the command.

what string

What is wrong.

ArchMsg.Error.rl_priq_streamid_truncated

The SMMU received a PCIe PRI request with a StreamID) that was larger than that which the SMMU has been configured for. The StreamID that appears in the PRIQ entry will be truncated. Fields:

actual_streamid unsigned int

The actual StreamID that this request has.

sidsize unsigned int

The bit width of the SMMU for StreamIDs, as indicated by SMMU_IDR1.SIDSIZE.

trans_id unsigned int

The transaction ID that identifies this request.

trunc_streamid unsigned int

The truncated StreamID that will appear in the PRIQ entry.

ArchMsg.Error.rl_priq_substreamid_truncated

The SMMU received a PCIe PRI request with a PASID prefix (SubstreamID) that was larger than that which the SMMU has been configured for. The SubstreamID that appears in the PRIQ entry will be truncated. Fields:

actual_substreamid unsigned int

The actual SubstreamID that this request has.

ssidsize unsigned int

The bit width of the SMMU for SubstreamIDs, as indicated by SMMU_IDR1.SSIDSIZE.

trans_id unsigned int

The transaction ID that identifies this request.

trunc_substreamid unsigned int

The truncated SubstreamID that will appear in the PRIQ entry.

ArchMsg.Error.s_cmdq

There is an error associated with the command queue. Fields:

cmd_id unsigned int

Command ID of the command in error.

cons unsigned int

CONS of the command.

what string

What is wrong.

ArchMsg.Error.tlb_entries_overlap

A TLB entry was attempted to be inserted into the TLB and was determined that it overlaps an existing entry. This check is not perfect but will catch simple errors. Fields:

do_f_tlb_conflict bool

Chosen to perform an F_TLB_CONFLICT.

end_address_of_new_entry unsigned int

End address of new entry.

end_address_of_old_entry unsigned int

End address of old entry.

index_of_new_entry unsigned int

Index of new entry.

index_of_old_entry unsigned int

Index of old entry.

start_address_of_new_entry unsigned int

Start address of new entry.

start_address_of_old_entry unsigned int

Start address of old entry.

why enum

Why it is an error for these two entries to overlap.

ArchMsg.Error.tlb_entry_not_invalidated_due_to_ril

An entry in the cache was not invalidated even though in the right address range because of the RIL fields of the command the entry do not match. Fields:

cmd_num unsigned int

The NUM field of the RIL part of the command.

cmd_ril_tg enum

The RIL_TG field of the RIL part of the command.

cmd_ril_ttl unsigned int

The TTL field of the RIL part of the command, zero means any level. 0x80 means match level 0 (and is from a DVM message).

cmd_scale unsigned int

The SCALE field of the RIL part of the command.

entry_id unsigned int

The entry id that is being invalidated.

tlb_entry string

The TLB entry.

ArchMsg.Error.vatos_sel_vmid_out_of_range

The SMMU_(S_)VATOS_SEL.VMID field was programmed with a VMID that was too wide for this implementation (SMMU_IDR0.VMID16 == 0). DISPLAY %{{ssd_ns:(s-|ns-)}}VMID:%{vmid} is out of range. Fields:

ssd_ns bool

The security state of the VATOS interface.

vmid unsigned int

The VMID programmed.

ArchMsg.Info.info

These are information messages about what is happening in the SMMU. DISPLAY %{output}. Fields:

output string

The stream output.

ArchMsg.Warning.CMD_RESUME_no_transaction_resumed

A CMD_RESUME was issued that matched no transaction. Fields:

stag unsigned int

STAG in the CMD_RESUME.

streamid unsigned int

StreamID in the CMD_RESUME.

streamid_ns bool

The StreamID was for the non-secure world.

ArchMsg.Warning.NoStreamID_bad_pas_or_mpam_sp

When RME is not supported then a NoStreamID transaction with PAS[1] == 1 or MPAM_SP[1] == 1 is treated as though PAS[1] == 0 and MPAM_SP[1] == 0. This is usually a system construction error and is not expected to occur. Fields:

address unsigned int

The Physical Address.

effective_mpam_sp enum

The effective MPAM_SP.

effective_pas enum

The effective PAS.

extendedid unsigned int

The ExtendedID of the transaction.

incoming_mpam_sp enum

The incoming MPAM_SP.

incoming_pas enum

The incoming PAS.

managerid64 unsigned int

The ManagerID64 of the transaction,.

userflags unsigned int

The UserFlags of the transaction.

ArchMsg.Warning.atc_inv_strange

Something was odd about the CMD_ATC_INV. DISPLAY CMD_ATC_INV strange as: %{why}.
Fields:

cmd_id unsigned int

Command id.

ssd_of_cmdq enum

The SSD of the CMDQ.

why string

Why the CMD_ATC_INV was strange.

ArchMsg.Warning.bad_axi_stream_msi_addr_to_match_s

The signal was not driven at reset by the external system, but was driven after we first started using the default value we had assumed. The external system, if it is going to drive the signal, must do so at first reset so that both sides of the connection can agree on the value of this signal. Fields:

default_value unsigned int

The default value of the signal we have been assuming.

value unsigned int

The value of the signal.

ArchMsg.Warning.bad_conf_system_supports_bgptm

The signal was not driven at reset by the external system, but was driven after we first started using the default value we had assumed. The external system, if it is going to drive the signal, must do so at first reset so that both sides of the connection can agree on the value of this signal. Fields:

default_value bool

The default value of the signal we have been assuming.

value bool

The value of the signal.

ArchMsg.Warning.bad_l0gptsz

The signal was not driven at reset by the external system, but was driven after we first started using the default value we had assumed. The external system, if it is going to drive the signal, must do so at first reset so that both sides of the connection can agree on the value of this signal. Fields:

default_value unsigned int

The default value of the signal we have been assuming.

value unsigned int

The value of the signal.

ArchMsg.Warning.bad_legacy_tz_en

The signal was not driven at reset by the external system, but was driven after we first started using the default value we had assumed. The external system, if it is going to drive the signal, must do so at first reset so that both sides of the connection can agree on the value of this signal. Fields:

default_value bool

The default value of the signal we have been assuming.

value bool

The value of the signal.

ArchMsg.Warning.bad_s_gbpa_abort_init

The signal was not driven at reset by the external system, but was driven after we first started using the default value we had assumed. The external system, if it is going to drive the signal, must do so at first reset so that both sides of the connection can agree on the value of this signal. Fields:

default_value bool

The default value of the signal we have been assuming.

value bool

The value of the signal.

ArchMsg.Warning.bad_sec_override

The signal was not driven at reset by the external system, but was driven after we first started using the default value we had assumed. The external system, if it is going to drive the signal, must do so at first reset so that both sides of the connection can agree on the value of this signal. Fields:

default_value bool

The default value of the signal we have been assuming.

value bool

The value of the signal.

ArchMsg.Warning.bad_sup_btm

The signal was not driven at reset by the external system, but was driven after we first started using the default value we had assumed. The external system, if it is going to drive the signal, must do so at first reset so that both sides of the connection can agree on the value of this signal. Fields:

default_value bool

The default value of the signal we have been assuming.

value bool

The value of the signal.

ArchMsg.Warning.bad_sup_cohacc

The signal was not driven at reset by the external system, but was driven after we first started using the default value we had assumed. The external system, if it is going to drive

the signal, must do so at first reset so that both sides of the connection can agree on the value of this signal. Fields:

default_value bool

The default value of the signal we have been assuming.

value bool

The value of the signal.

ArchMsg.Warning.bad_sup_httu

The signal was not driven at reset by the external system, but was driven after we first started using the default value we had assumed. The external system, if it is going to drive the signal, must do so at first reset so that both sides of the connection can agree on the value of this signal. Fields:

default_value bool

The default value of the signal we have been assuming.

value bool

The value of the signal.

ArchMsg.Warning.bad_sup_sev

The signal was not driven at reset by the external system, but was driven after we first started using the default value we had assumed. The external system, if it is going to drive the signal, must do so at first reset so that both sides of the connection can agree on the value of this signal. Fields:

default_value bool

The default value of the signal we have been assuming.

value bool

The value of the signal.

ArchMsg.Warning.bad_tbu0_reset_in

The signal was not driven at reset by the external system, but was driven after we first started using the default value we had assumed. The external system, if it is going to drive the signal, must do so at first reset so that both sides of the connection can agree on the value of this signal. Fields:

default_value bool

The default value of the signal we have been assuming.

value bool

The value of the signal.

ArchMsg.Warning.bad_tcu_reset_in

The signal was not driven at reset by the external system, but was driven after we first started using the default value we had assumed. The external system, if it is going to drive the signal, must do so at first reset so that both sides of the connection can agree on the value of this signal. Fields:

default_value bool

The default value of the signal we have been assuming.

value bool

The value of the signal.

ArchMsg.Warning.contig_bit_gives_too_large_region_for_TxSZ

If the contig bit was used then the size of the contig region would be larger than that indicated by TxSZ. Fields:

desc_kind enum

Descriptor kind.

el enum

Exception level.

input_address unsigned int

Input address.

log2_contig_region_size unsigned int

Log2(contig region size in bytes).

output_address unsigned int

Output address.

stage_and_level unsigned int

The stage in bits [7:4] and the level in [3:0]. Level -1 is 0xF.

vmid unsigned int

VMID if appropriate.

ArchMsg.Warning.contig_bit_has_inconsistent_input_and_output_address

If the contig bit was used then the some bits of the output address held in the descriptor and the input address must match. Fields:

desc_kind enum

Descriptor kind.

el enum

Exception level.

input_address unsigned int

Input address.

log2_contig_region_size unsigned int

Log2(contig region size in bytes).

match_mask unsigned int

Bits that must match.

output_address unsigned int

Output address.

stage_and_level unsigned int

The stage in bits [7:4] and the level in [3:0]. Level -1 is 0xF.

vmid unsigned int

VMID if appropriate.

ArchMsg.Warning.fetch_from_memory_type_not_supporting_httu

A descriptor fetch from an HTTU-enabled translation regime to an unsupported memory type was made. Whilst the fetch itself may succeed, if an update to the descriptor was attempted then it would fail. Fields:

address unsigned int

The address of the descriptor fetch.

desc_inner enum

Inner cacheability of descriptor.

desc_outer enum

Outer cacheability of descriptor.

desc_prot enum

The instruction/data, bus-ns/s and privileged/user marking.

desc_sh enum

Shareability of descriptor.

ssd enum

The SSD of the transaction.

ssd_ns bool

Is the Security State Determination of the transaction non-secure.

stage unsigned int

The stage at which we had a problem.

streamid unsigned int

The StreamID of the transaction.

substreamid unsigned int

The SubstreamID of the transaction (or ~0u if no substreamid).

trans_id unsigned int

Transaction id.

ArchMsg.Warning.gpt_read_invalid_descriptor

A GPT read has completed and not found something that looks like a descriptor. Fields:

abort enum

Non-zero if the access aborted/failed.

address_checking unsigned int

The address that we are checking.

address_gptdesc unsigned int

The address of the GPT descriptor.

data unsigned int

The data fetch if it didn't abort.

level unsigned int

Level of the walk.

pas_checking enum

The PAS of the address that we are checking.

trans_id unsigned int

The trans_id of the transaction we are checking if appropriate.

what_checking enum

What kind of access are we checking.

ArchMsg.Warning.msi_address_truncated

An MSI was generated, but the address was silently truncated due to the limited downstream address bus width. Fields:

pmcg_index_and_counter unsigned int

If this is a PMCG interrupt, then the top 16 bits are the PMCG index, the lower 16 bits are counter index.

ras_error_group_id_node_id_and_record_index_in_node unsigned int

If this is a RAS interrupt then this is the error group record id in bits[23:16], the node_id in [15:8] and the record index in the node in [7:0].

truncated_address unsigned int

The truncated address.

untruncated_address unsigned int

The untruncated address.

which enum

Which MSI this is.

ArchMsg.Warning.msi_lost

An MSI was attempted to be sent, but couldn't be sent. Fields:

id unsigned int

ID of this interrupt transaction.

kind enum

What kind of interrupt.

why enum

Why this interrupt was denied.

ArchMsg.Warning.pmcg_non_secure_world

A commentary on potential mis-programming of the PMCG when transitioning the PMCG between security states. Fields:

commentary string

The commentary.

ArchMsg.Warning.pmcg_programming_violates_security

A commentary on potential mis-programming of the PMCG when transitioning the PMCG between different security states. This is under the assumption that an agent from another security state could be writing to the PMCG. Fields:

commentary string

The commentary.

ArchMsg.Warning.priq_auto_response_failed_to_find_STE

The PRIQ was going to generate an auto-response, but failed to find an STE and so is returning a Failure message to the EndPoint which should disable the PRI interface of the EndPoint. Fields:

trans_id unsigned int

The ID of the PPR that is being lost.

why enum

Why the PPR is being lost.

ArchMsg.Warning.priq_overflow_bad_acking

Indicates that an overflow condition was acknowledged by writing to:- SMMU_PRIQ_CONS.OVACKFLG but an overflow condition did not exist. Fields:

new_ovackflg bool

The new value of the SMMU_PRIQ_CONS.OVACKFLG.

ArchMsg.Warning.priq_smmuen_forces_effective_priqen_low

If SMMUEN == 0, then the effective value of PRIQEN is 0. This warning is triggered when PRIQEN == 1 && SMMUEN == 0; which may not be what was intended. The PRIQ cannot be active if SMMUEN == 0.

ArchMsg.Warning.rl_priq_auto_response_failed_to_find_STE

The PRIQ was going to generate an auto-response, but failed to find an STE and so is returning a Failure message to the EndPoint which should disable the PRI interface of the EndPoint. Fields:

trans_id unsigned int

The ID of the PPR that is being lost.

why enum

Why the PPR is being lost.

ArchMsg.Warning.rl_priq_overflow_bad_acking

Indicates that an overflow condition was acknowledged by writing to:-
SMMU_PRIQ_CONS.OVACKFLG but an overflow condition did not exist. Fields:

new_ovackflg bool

The new value of the SMMU_PRIQ_CONS.OVACKFLG.

ArchMsg.Warning.rl_priq_smmuen_forces_effective_priqen_low

If SMMUEN == 0, then the effective value of PRIQEN is 0. This warning is triggered when
PRIQEN == 1 && SMMUEN == 0; which may not be what was intended. The PRIQ cannot
be active if SMMUEN == 0.

ArchMsg.Warning.sev_lost

A SEV was lost because it isn't supported according to SMMU_IDRO.SEV. DISPLAY SEV was
lost because: %{why}. Fields:

why enum

Why the SEV was generated.

ArchMsg.Warning.smmu_pcie_rc_is_in_reset_ignoring_atc_invalidate

A CMD_PRI_RESP or CMD_ATC_INV was sent with a streamid that mapped to a PCIe
Root Complex that is in reset. The message will be discarded and act as though completed
successfully. Fields:

port_index unsigned int

The port index (node index) that is in reset but we would have sent it to.

streamid unsigned int

The StreamID that failed to map.

ArchMsg.Warning.smmu_pcie_rc_not_found_for_streamid

A CMD_PRI_RESP or CMD_ATC_INV was sent with a streamid that did not map to a PCIe
Root Complex. This might be that the SW used an incorrect StreamID or it might be that
the model has not been connected correctly. ATC Invalidate messages complete as though
successful and PRI Requests are ignored. Fields:

streamid unsigned int

The StreamID that failed to map.

ArchMsg.Warning.smmu_pmusnapshot

Something strange happened on the pmusnapshot_req/pmusnapshot_ack interface. Fields:

pin_index unsigned int

If the PMCG index corresponds to an array of signals, this is the index in the array, or 0
otherwise.

pmcg_index enum

The PMCG index.

warning string

The warning message.

ArchMsg.Warning.suspicious_overlapping_entries

Two DPT TLB entries are overlapping but they differ in ways that are potentially a SW error.
Fields:

new_AC enum

The value of the 'AC' field that controls access to this region.

new_FWB bool

The region is FWB.

new_VMID unsigned int

The VMID, if any, associated with this region.

new_entry_index unsigned int

The entry index of the new DPT TLB entry.

new_entry_is_vmsa_formed bool

The new entry was inserted as a result of VMSA information.

new_input_end_incl_address unsigned int

The end inclusive address of the input range that this matches.

new_input_start_address unsigned int

The start address of the input range that this matches.

new_output_pas enum

The output PAS of this DPT region.

new_ssd enum

The SSD of the streams this region captures.

new_writeable bool

True if this region is writeable.

old_AC enum

The value of the 'AC' field that controls access to this region.

old_FWB bool

The region is FWB.

old_VMID unsigned int

The VMID, if any, associated with this region.

old_entry_index unsigned int

The entry index of the old DPT TLB entry.

old_entry_is_vmsa_formed bool

The old entry was inserted as a result of VMSA information.

old_input_end_incl_address unsigned int

The end inclusive address of the input range that this matches.

old_input_start_address unsigned int

The start address of the input range that this matches.

old_output_pas enum

The output PAS of this DPT region.

old_ssd enum

The SSD of the streams this region captures.

old_writeable bool

True if this region is writeable.

pas_differ bool

The output PAS of the two entries differ.

vmids_differ bool

The VMIDs are used by at least one of the AC schemes and are different.

vmsa_formed_writeable_while_DPT_entry_is_not bool

The VMSA-formed entry is writeable but the DPT Entry says it is not writeable.

ArchMsg.Warning.warning

These messages are about unusual (but not necessarily incorrect) activity occurring on the SMMU. DISPLAY %{output}. Fields:

output string

The stream output.

ArchMsg.Warning.warning_effective_EOPD_differs_from_what_would_be_cached_in_TLB

Effective value of EOPD differs from what would be cached in the TLB DISPLAY transaction (%{transaction_id}), sid (%{sid}), ssid (%{ssid}), ssd (%{ssd}), effective EOPD (%{effective_EOPD}), cached EOPD (%{cached_EOPD}). Fields:

cached_EOPD bool

The EOPD value that would be cached in the TLB.

effective_EOPD bool

The effective value of EOPD.

ssd enum

SSD.

streamid unsigned int

StreamID or ~0ull if NoStreamID.

substreamid unsigned int

SubstreamID or ~0u if SSValid == false.

transaction_id unsigned int

The transaction ID.

DPTTLB.tlb_commentary

A commentary from the cache about what is going on. Fields:

output string

The stream output.

DPTTLB.tlb_entry_invalidated

An entry in the cache has been invalidated. Fields:

entry_id unsigned int

The entry id that is being invalidated.

reason enum

Why the entry is being invalidated.

EVENTQ_config

Configuration of interrupt updated. Fields:

address unsigned int

Address of the MSI.

data unsigned int

Data payload of the MSI.

irqen bool

Appropriate bit of SMMU_s_IRQ_CTRL controlling this interrupt.

memattr enum

Memory type.

ns bool

PAS of bus attribute is non-secure.

pas enum

PAS of the MSI.

ra bool

read-allocate attribute.

sh enum

Shareability.

smmu_idr0_msi bool

Are MSIs supported for this security world.

tr bool

transient attribute.

wa bool

write-allocate attributes.

EVENTQ_irqen

The IRQEN has been changed. Fields:

new_value bool

The new value of IRQEN.

EVENTQ_irqen_ack

The IRQEN change has been acked. Fields:

new_value bool

The new value of IRQEN.

GERROR_config

Configuration of interrupt updated. Fields:

address unsigned int

Address of the MSI.

data unsigned int

Data payload of the MSI.

irqen bool

Appropriate bit of SMMU_s_IRQ_CTRL controlling this interrupt.

memattr enum

Memory type.

ns bool

PAS of bus attribute is non-secure.

pas enum

PAS of the MSI.

ra bool

read-allocate attribute.

sh enum

Shareability.

smmu_idr0_msi bool

Are MSIs supported for this security world.

tr bool

transient attribute.

wa bool

write-allocate attributes.

GERROR_irqen

The IRQEN has been changed. Fields:

new_value bool

The new value of IRQEN.

GERROR_irqen_ack

The IRQEN change has been acked. Fields:

new_value bool

The new value of IRQEN.

GPTTLB.tlb_commentary

A commentary from the cache about what is going on. Fields:

output string

The stream output.

GPTTLB.tlb_entry_invalidated

An entry in the cache has been invalidated. Fields:

entry_id unsigned int

The entry id that is being invalidated.

reason enum

Why the entry is being invalidated.

HACDBS_PROCESSING_COMPLETE_config

Configuration of interrupt updated. Fields:

address unsigned int

Address of the MSI.

data unsigned int

Data payload of the MSI.

irgen bool

Appropriate bit of SMMU_s_IRQ_CTRL controlling this interrupt.

memattr enum

Memory type.

ns bool

PAS of bus attribute is non-secure.

pas enum

PAS of the MSI.

ra bool

read-allocate attribute.

sh enum

Shareability.

smmu_idr0_msi bool

Are MSIs supported for this security world.

tr bool

transient attribute.

wa bool

write-allocate attributes.

HACDBS_PROCESSING_COMPLETE_irgen

The IRQEN has been changed. Fields:

new_value bool

The new value of IRQEN.

HACDBS_PROCESSING_COMPLETE_irqen_ack

The IRQEN change has been acked. Fields:

new_value bool

The new value of IRQEN.

HDBSS_TABLE_FULL_config

Configuration of interrupt updated. Fields:

address unsigned int

Address of the MSI.

data unsigned int

Data payload of the MSI.

irqen bool

Appropriate bit of SMMU_s_IRQ_CTRL controlling this interrupt.

memattr enum

Memory type.

ns bool

PAS of bus attribute is non-secure.

pas enum

PAS of the MSI.

ra bool

read-allocate attribute.

sh enum

Shareability.

smmu_idr0_msi bool

Are MSIs supported for this security world.

tr bool

transient attribute.

wa bool

write-allocate attributes.

HDBSS_TABLE_FULL_irqen

The IRQEN has been changed. Fields:

new_value bool

The new value of IRQEN.

HDBSS_TABLE_FULL_irqen_ack

The IRQEN change has been acked. Fields:

new_value bool

The new value of IRQEN.

MMU_S3_integration_mode_end_ras_level_interrupt_restored

RAS level sensitive interrupt restored due to integration mode ending. Fields:

is_tcu bool

Is TCU integration mode.

ras_interrupt enum

RAS interrupt being restored.

tbu_index unsigned int

TBU index. ~0u if it's TCU.

value bool

Level of the interrupt signal.

MMU_S3_integration_mode_pmcg_interrupt_lost

PMCG interrupt lost due to being in integration mode. Fields:

is_tcu enum

Is TCU integration mode.

pmcg_interrupt enum

PMCG interrupt being dropped.

tbu_index unsigned int

TBU index. ~0u if it's TCU.

MMU_S3_integration_mode_pmusnapshot_ack_lost

pmusnapshot_ack lost due to being in integration mode. Fields:

is_tcu enum

Is TCU integration mode.

tbu_index unsigned int

TBU index. ~0u if it's TCU.

MMU_S3_integration_mode_ras_interrupt_lost

RAS interrupt lost due to being in integration mode. Fields:

is_tcu enum

Is TCU integration mode.

ras_interrupt enum

RAS interrupt being dropped.

tbu_index unsigned int

TBU index. ~0u if it's TCU.

MMU_S3_integration_mode_ras_level_interrupt_lost

RAS level sensitive interrupt lost due to being in integration mode. Fields:

is_tcu bool

Is TCU integration mode.

ras_interrupt enum

RAS interrupt being dropped.

tbu_index unsigned int

TBU index. ~0u if it's TCU.

value bool

Level of the interrupt signal.

MMU_S3_integration_mode_start_ras_level_interrupt_cleared

RAS level sensitive interrupt cleared due to integration mode starting. Fields:

is_tcu bool

Is TCU integration mode.

ras_interrupt enum

RAS interrupt being cleared.

tbu_index unsigned int

TBU index. ~0u if it's TCU.

MMU_S3_integration_mode_tcu_evento_lost

Evento lost due to being in integration mode.

MMU_S3_integration_mode_tcu_interrupt_lost

Interrupt lost due to being in integration mode. Fields:

interrupt enum

Interrupt that is being dropped.

MMU_S3_integration_register_change

An integration register write occurred and it's modified, which drives a signal. Fields:

is_tcu enum

Is TCU integration mode.

new_value bool

New value of the signal.

old_value bool

Old value of the signal.

register enum

The register being modified.

register_value unsigned int

Value written to the register.

signal string

Signal affected by the register write.

tbu_index unsigned int

TBU index. ~0u if it's TCU.

PRIQ_config

Configuration of interrupt updated. Fields:

address unsigned int

Address of the MSI.

data unsigned int

Data payload of the MSI.

irqen bool

Appropriate bit of SMMU_s_IRQ_CTRL controlling this interrupt.

memattr enum

Memory type.

ns bool

PAS of bus attribute is non-secure.

pas enum

PAS of the MSI.

ra bool

read-allocate attribute.

sh enum

Shareability.

smmu_idr0_msi bool

Are MSIs supported for this security world.

tr bool

transient attribute.

wa bool

write-allocate attributes.

PRIQ_irqen

The IRQEN has been changed. Fields:

new_value bool

The new value of IRQEN.

PRIQ_irqen_ack

The IRQEN change has been acked. Fields:

new_value bool

The new value of IRQEN.

R_EVENTQ_config

Configuration of interrupt updated. Fields:

address unsigned int

Address of the MSI.

data unsigned int

Data payload of the MSI.

irqen bool

Appropriate bit of SMMU_s_IRQ_CTRL controlling this interrupt.

memattr enum

Memory type.

ns bool

PAS of bus attribute is non-secure.

pas enum

PAS of the MSI.

ra bool

read-allocate attribute.

sh enum

Shareability.

smmu_idr0_msi bool

Are MSIs supported for this security world.

tr bool

transient attribute.

wa bool

write-allocate attributes.

R_EVENTQ_irqen

The IRQEN has been changed. Fields:

new_value bool

The new value of IRQEN.

R_EVENTQ_irqen_ack

The IRQEN change has been acked. Fields:

new_value bool

The new value of IRQEN.

R_GERROR_config

Configuration of interrupt updated. Fields:

address unsigned int

Address of the MSI.

data unsigned int

Data payload of the MSI.

irqen bool

Appropriate bit of SMMU_s_IRQ_CTRL controlling this interrupt.

memattr enum

Memory type.

ns bool

PAS of bus attribute is non-secure.

pas enum

PAS of the MSI.

ra bool

read-allocate attribute.

sh enum

Shareability.

smmu_idr0_msi bool

Are MSIs supported for this security world.

tr bool

transient attribute.

wa bool

write-allocate attributes.

R_ERROR_irqen

The IRQEN has been changed. Fields:

new_value bool

The new value of IRQEN.

R_ERROR_irqen_ack

The IRQEN change has been acked. Fields:

new_value bool

The new value of IRQEN.

R_HACDBS_PROCESSING_COMPLETE_config

Configuration of interrupt updated. Fields:

address unsigned int

Address of the MSI.

data unsigned int

Data payload of the MSI.

irqen bool

Appropriate bit of SMMU_s_IRQ_CTRL controlling this interrupt.

memattr enum

Memory type.

ns bool

PAS of bus attribute is non-secure.

pas enum

PAS of the MSI.

ra bool

read-allocate attribute.

sh enum

Shareability.

smmu_idr0_msi bool

Are MSIs supported for this security world.

tr bool

transient attribute.

wa bool

write-allocate attributes.

R_HACDBS_PROCESSING_COMPLETE_irqen

The IRQEN has been changed. Fields:

new_value bool

The new value of IRQEN.

R_HACDBS_PROCESSING_COMPLETE_irqen_ack

The IRQEN change has been acked. Fields:

new_value bool

The new value of IRQEN.

R_HDBSS_TABLE_FULL_config

Configuration of interrupt updated. Fields:

address unsigned int

Address of the MSI.

data unsigned int

Data payload of the MSI.

irqen bool

Appropriate bit of SMMU_s_IRQ_CTRL controlling this interrupt.

memattr enum

Memory type.

ns bool

PAS of bus attribute is non-secure.

pas enum

PAS of the MSI.

ra bool

read-allocate attribute.

sh enum

Shareability.

smmu_idr0_msi bool

Are MSIs supported for this security world.

tr bool

transient attribute.

wa bool

write-allocate attributes.

R_HDBSS_TABLE_FULL_irqen

The IRQEN has been changed. Fields:

new_value bool

The new value of IRQEN.

R_HDBSS_TABLE_FULL_irqen_ack

The IRQEN change has been acked. Fields:

new_value bool

The new value of IRQEN.

R_PRIQ_config

Configuration of interrupt updated. Fields:

address unsigned int

Address of the MSI.

data unsigned int

Data payload of the MSI.

irqen bool

Appropriate bit of SMMU_s_IRQ_CTRL controlling this interrupt.

memattr enum

Memory type.

ns bool

PAS of bus attribute is non-secure.

pas enum

PAS of the MSI.

ra bool

read-allocate attribute.

sh enum

Shareability.

smmu_idr0_msi bool

Are MSIs supported for this security world.

tr bool

transient attribute.

wa bool

write-allocate attributes.

R_PRIQ_irqen

The IRQEN has been changed. Fields:

new_value bool

The new value of IRQEN.

R_PRIQ_irqen_ack

The IRQEN change has been acked. Fields:

new_value bool

The new value of IRQEN.

SMMU_CR0ACK_SMMUEN_hazarded_by_priq

The SMMU_r_CR0ACK.SMMUEN cannot acknowledge the change to SMMUEN because there are outstanding PRIQ writes.

SMMU_CR0ACK_SMMUEN_update

The acknowledge to SMMU_CR0.SMMUEN was performed. Fields:

new_value bool

The new value of the register.

SMMU_CR0_SMMUEN_old_set_complete

A set of transactions associated with the old value of SMMU_CR0.SMMUEN completed. Fields:

last bool

This is the last set completing.

old_value bool

The old value of the set completing.

SMMU_CR0_SMMUEN_write

A write to SMMU_CR0.SMMUEN occurred. Fields:

new_value bool

The new value of the register.

old_value bool

The old value of the register.

SMMU_GBPA_old_set_complete

A set of transactions associated with the old value of SMMU_GBPA completed. Fields:

last bool

This is the last set completing.

old_value unsigned int

The old value of the set completing.

SMMU_GBPA_update

The Update flag to SMMU_GBPA was lowered. Fields:

new_value unsigned int

The new value of the register.

SMMU_GBPA_write

A write to SMMU_GBPA occurred. Fields:

new_value unsigned int

The new value of the register.

old_value unsigned int

The old value of the register.

SMMU_ROOT_CR0ACK_ACCESSEN_update

The acknowledge to SMMU_ROOT_CR0.ACCESSSEN was performed. Fields:

new_value bool

The new value of the register.

SMMU_ROOT_CR0ACK_GPCEN_update

The acknowledge to SMMU_ROOT_CR0.GPCEN was performed. Fields:

new_value bool

The new value of the register.

SMMU_ROOT_CR0_ACCESSEN_old_set_complete

A set of transactions associated with the old value of SMMU_ROOT_CR0.ACCESSSEN completed. Fields:

last bool

This is the last set completing.

old_value bool

The old value of the set completing.

SMMU_ROOT_CR0_ACCESSEN_write

A write to SMMU_ROOT_CR0.ACCESSSEN occurred. Fields:

new_value bool

The new value of the register.

old_value bool

The old value of the register.

SMMU_ROOT_CR0_GPCEN_old_set_complete

A set of transactions associated with the old value of SMMU_ROOT_CR0.GPCEN completed.
Fields:

last bool

This is the last set completing.

old_value bool

The old value of the set completing.

SMMU_ROOT_CR0_GPCEN_write

A write to SMMU_ROOT_CR0.GPCEN occurred. Fields:

new_value bool

The new value of the register.

old_value bool

The old value of the register.

SMMU_R_CR0ACK_SMMUEN_hazarded_by_rl_priq

The SMMU_r_CR0ACK.SMMUEN cannot acknowledge the change to SMMUEN because there are outstanding PRIQ writes.

SMMU_R_CR0ACK_SMMUEN_update

The acknowledge to SMMU_R_CR0.SMMUEN was performed. Fields:

new_value bool

The new value of the register.

SMMU_R_CR0_SMMUEN_old_set_complete

A set of transactions associated with the old value of SMMU_R_CR0.SMMUEN completed.
Fields:

last bool

This is the last set completing.

old_value bool

The old value of the set completing.

SMMU_R_CR0_SMMUEN_write

A write to SMMU_R_CR0.SMMUEN occurred. Fields:

new_value bool

The new value of the register.

old_value bool

The old value of the register.

SMMU_S_CR0ACK_SMMUEN_update

The acknowledge to SMMU_S_CR0.SMMUEN was performed. Fields:

new_value bool

The new value of the register.

SMMU_S_CR0_SMMUEN_old_set_complete

A set of transactions associated with the old value of SMMU_S_CR0.SMMUEN completed.

Fields:

last bool

This is the last set completing.

old_value bool

The old value of the set completing.

SMMU_S_CR0_SMMUEN_write

A write to SMMU_S_CR0.SMMUEN occurred. Fields:

new_value bool

The new value of the register.

old_value bool

The old value of the register.

SMMU_S_GBPA_old_set_complete

A set of transactions associated with the old value of SMMU_S_GBPA completed. Fields:

last bool

This is the last set completing.

old_value unsigned int

The old value of the set completing.

SMMU_S_GBPA_update

The Update flag to SMMU_S_GBPA was lowered. Fields:

new_value unsigned int

The new value of the register.

SMMU_S_GBPA_write

A write to SMMU_S_GBPA occurred. Fields:

new_value unsigned int

The new value of the register.

old_value unsigned int

The old value of the register.

S_EVENTQ_config

Configuration of interrupt updated. Fields:

address unsigned int

Address of the MSI.

data unsigned int

Data payload of the MSI.

irqen bool

Appropriate bit of SMMU_s_IRQ_CTRL controlling this interrupt.

memattr enum

Memory type.

ns bool

PAS of bus attribute is non-secure.

pas enum

PAS of the MSI.

ra bool

read-allocate attribute.

sh enum

Shareability.

smmu_idr0_msi bool

Are MSIs supported for this security world.

tr bool

transient attribute.

wa bool

write-allocate attributes.

S_EVENTQ_irqen

The IRQEN has been changed. Fields:

new_value bool

The new value of IRQEN.

S_EVENTQ_irqen_ack

The IRQEN change has been acked. Fields:

new_value bool

The new value of IRQEN.

S_GERROR_config

Configuration of interrupt updated. Fields:

address unsigned int

Address of the MSI.

data unsigned int

Data payload of the MSI.

irqen bool

Appropriate bit of SMMU_s_IRQ_CTRL controlling this interrupt.

memattr enum

Memory type.

ns bool

PAS of bus attribute is non-secure.

pas enum

PAS of the MSI.

ra bool

read-allocate attribute.

sh enum

Shareability.

smmu_idr0_msi bool

Are MSIs supported for this security world.

tr bool

transient attribute.

wa bool

write-allocate attributes.

S_ERROR_irqen

The IRQEN has been changed. Fields:

new_value bool

The new value of IRQEN.

S_ERROR_irqen_ack

The IRQEN change has been acked. Fields:

new_value bool

The new value of IRQEN.

S_HACDBS_PROCESSING_COMPLETE_config

Configuration of interrupt updated. Fields:

address unsigned int

Address of the MSI.

data unsigned int

Data payload of the MSI.

irqen bool

Appropriate bit of SMMU_s_IRQ_CTRL controlling this interrupt.

memattr enum

Memory type.

ns bool

PAS of bus attribute is non-secure.

pas enum

PAS of the MSI.

ra bool

read-allocate attribute.

sh enum

Shareability.

smmu_idr0_msi bool

Are MSIs supported for this security world.

tr bool

transient attribute.

wa bool

write-allocate attributes.

S_HACDBS_PROCESSING_COMPLETE_irqen

The IRQEN has been changed. Fields:

new_value bool

The new value of IRQEN.

S_HACDBS_PROCESSING_COMPLETE_irqen_ack

The IRQEN change has been acked. Fields:

new_value bool

The new value of IRQEN.

S_HDBSS_TABLE_FULL_config

Configuration of interrupt updated. Fields:

address unsigned int

Address of the MSI.

data unsigned int

Data payload of the MSI.

irqen bool

Appropriate bit of SMMU_s_IRQ_CTRL controlling this interrupt.

memattr enum

Memory type.

ns bool

PAS of bus attribute is non-secure.

pas enum

PAS of the MSI.

ra bool

read-allocate attribute.

sh enum

Shareability.

smmu_idr0_msi bool

Are MSIs supported for this security world.

tr bool

transient attribute.

wa bool

write-allocate attributes.

S_HDBSS_TABLE_FULL_irqen

The IRQEN has been changed. Fields:

new_value bool

The new value of IRQEN.

S_HDBSS_TABLE_FULL_irqen_ack

The IRQEN change has been acked. Fields:

new_value bool

The new value of IRQEN.

TLB.tlb_commentary

A commentary from the cache about what is going on. Fields:

output string

The stream output.

TLB.tlb_entry_invalidated

An entry in the cache has been invalidated. Fields:

entry_id unsigned int

The entry id that is being invalidated.

reason enum

Why the entry is being invalidated.

atc_inv_nop

The CMD_ATC_INV command is ignored as a **NOP**. This may be emitted multiple times if the CMD_ATC_INV is being ignored for multiple reasons. Fields:

address unsigned int

Untranslated Address to invalidate.

cmd_id unsigned int

Command id.

cons unsigned int

CONS of the command.

global bool

Global flag.

log2_size_in_bytes unsigned int

Log 2 of the size in bytes of the region to invalidate.

ssd_of_queue enum

The SSD of the CMDQ.

ssd_of_streamid enum

The SSD of the StreamID.

streamid unsigned int

StreamID in the invalidate command.

substreamid unsigned int

SubstreamID or ~0u if SSValid == false.

why enum

Why was NOPped.

atos_complete_fault

The ATOS operation completed with a fault. Fields:

effective_st1translate bool

Because of the settings and/or the ATOS type then the effective st1translate can be different.

fault_faddr unsigned int

The fault FADDR.

fault_faultcode enum

The fault code.

fault_reason enum

The fault reason.

ssd_ns bool

This is a non-secure ATOS operation.

st1translate bool

The StreamID/SubstreamID combination should have been translated by stage 1.

st2translate bool

The translation suffered a stage 2 translation.

state enum

The final transaction state of the ATOS operation.

vatos_vmid unsigned int

The VMID of the VATOS operation, or ~0u if not a VATOS operation.

atos_complete_fault_inv_req

The ATOS operation completed, faulted and generated an INV_REQ response. Fields:

ssd_ns bool

This is a non-secure ATOS operation.

vatos_vmid unsigned int

The VMID of the VATOS operation, or ~0u if not a VATOS operation.

why enum

Why the ATOS operation generated an INV_REQ.

atos_complete_success

The ATOS operation completed successfully. Fields:

base_addr unsigned int

The actual base address of region.

effective_st1translate bool

Because of the settings and/or the ATOS type then the effective st1translate can be different.

par_addr unsigned int

The PAR.ADDR field.

par_mair unsigned int

The memory attributes encoded as a MAIR.

par_ns bool

The PAR.NS field, for an SSD-ns request then this will always be 0.

par_sh enum

Shareability.

par_size bool

The PAR.Size field.

size_in_bytes unsigned int

The actual size in bytes of the region.

ssd_ns bool

This is a non-secure ATOS operation.

st1translate bool

The StreamID/SubstreamID combination should have been translated by stage 1.

st2translate bool

The translation suffered a stage 2 translation.

state enum

The final transaction state of the ATOS operation.

vatos_vmid unsigned int

The VMID of the VATOS operation, or ~0u if not a VATOS operation.

atos_run_set

The SMMU_s_GATOS_CTRL.RUN field was set to start the ATOS operation. Fields:

ssd_ns bool

This is a non-secure ATOS operation.

vatos bool

This is a VATOS operation.

atos_starting

The ATOS operation is starting. Fields:

addr unsigned int

The input address to the ATOS operation.

httui bool

Inhibit HTTU update.

ind bool

Instruction Data.

pnu bool

Privileged not User.

rnw bool

Read not Write.

ssd_ns bool

This is a non-secure ATOS operation.

ssec bool

If this is a secure ATOS operation then this is if it is secure or not.

streamid unsigned int

The StreamID requested.

substreamid unsigned int

The SubstreamID requested, or ~0u if no SubstreamID.

type enum

The requested ATOS type.

vatos_vmid unsigned int

The VMID of the VATOS operation, or ~0u if not a VATOS operation.

axi_stream_msi_addr_to_match_s

Address to use to send SMMU originated MSIs directly to the GIC. This pin is sampled on the negedge of the reset pin and has to be driven beforehand for the value to take effect. Fields:

value unsigned int

The value of the signal.

cd_cc.CD_commentary

A commentary from the cache about what is going on. Fields:

output string

The stream output.

cd_cc.cd_entry_invalidated

An entry in the cache has been invalidated. Fields:

entry_id unsigned int

The entry id that is being invalidated.

reason enum

Why the entry is being invalidated.

cd_entry_allocated

An CD entry has been allocated. Fields:

AssuredTranslation bool

The CD (and any L1CD) was fetched from an AssuredOnly stage 2 page and so might be capable of creating AssuredTranslations.

cd string

A textual description of the CD.

entry_id unsigned int

The id of the cache entry. Top bit is set if is a debug entry.

ns enum

For the non-secure world.

ssd enum

The SSD of the entry.

streamid unsigned int

The StreamID of the entry this will match.

substreamid unsigned int

The SubstreamID of the entry this will match. This may be zero for transactions without a SubstreamID.

conf_system_supports_bgptm

The pin is driven. This indicates system supports broadcast TLBI by PA operations. This pin is sampled on the negedge of the reset pin and has to be driven beforehand for the value to take effect. Fields:

value bool

The value of the signal.

dpttlb_entry_allocated

A DPT TLB entry has been allocated. Fields:

AC enum

The value of the 'AC' field that controls access to this region.

FWB bool

The region is FWB.

VMID unsigned int

The VMID, if any, associated with this region.

index unsigned int

Index of the TLB entry.

input_end_incl_address unsigned int

The end inclusive address of the input range that this matches.

input_start_address unsigned int

The start address of the input range that this matches.

output_pas enum

The output PAS of this DPT region.

ssd enum

The SSD of the streams this region captures.

trans_id unsigned int

The trans_id of the transaction that caused this allocation.

vmsa_formed bool

The entry was formed from VMSA information rather than from a DPT walk.

writeable bool

True if this region is writeable.

dpttlb_invalidate_intersects_but_does_not_cover_entry_range

ENCODED_SIZE < size of the region covered by the DPT entry so invalidation is not architecturally guaranteed. No invalidation is performed. Fields:

dpttlb_entry_id unsigned int

ID of the TLB entry.

entry_end_incl_address unsigned int

Last address covered by the TLB entry.

entry_start_address unsigned int

First address covered by the TLB entry.

invalidate_end_incl_address unsigned int

Last address covered by the invalidation range.

invalidate_start_address unsigned int

First address covered by the invalidation range.

ssd enum

The security state of the TLB entry.

dpttlb_overlapping_entries

Two DPT TLB entries are overlapping. Fields:

new_AC enum

The value of the 'AC' field that controls access to this region.

new_FWB bool

The region is FWB.

new_VMID unsigned int

The VMID, if any, associated with this region.

new_entry_index unsigned int

The entry index of the new DPT TLB entry.

new_entry_is_vmsa_formed bool

The new entry was inserted as a result of VMSA information.

new_input_end_incl_address unsigned int

The end inclusive address of the input range that this matches.

new_input_start_address unsigned int

The start address of the input range that this matches.

new_output_pas enum

The output PAS of this DPT region.

new_ssd enum

The SSD of the streams this region captures.

new_writeable bool

True if this region is writeable.

old_AC enum

The value of the 'AC' field that controls access to this region.

old_FWB bool

The region is FWB.

old_VMID unsigned int

The VMID, if any, associated with this region.

old_entry_index unsigned int

The entry index of the old DPT TLB entry.

old_entry_is_vmsa_formed bool

The old entry was inserted as a result of VMSA information.

old_input_end_incl_address unsigned int

The end inclusive address of the input range that this matches.

old_input_start_address unsigned int

The start address of the input range that this matches.

old_output_pas enum

The output PAS of this DPT region.

old_ssd enum

The SSD of the streams this region captures.

old_writeable bool

True if this region is writeable.

dvm_tlbinvalidate_complete

The DVM TLB Invalidate message completed. Fields:

id unsigned int

The unique id of this DVM message.

ok bool

The DVM message was OK.

dvm_tlbinvalidate_received

A DVM message for a TLB Invalidate has been received. Fields:

address unsigned int

The VA or IPA to use if match_address.

asid unsigned int

The ASID to match if match_asid.

by_ipa bool

The operation is for an IPA operation if match_address.

id unsigned int

The unique id of this DVM message.

ignored enum

The DVM message was ignored.

last_level bool

The operation is for last level if supported.

match_address bool

Match the address field.

match_asid bool

Match the asid field.

match_vmid bool

Match the vmid field.

num unsigned int

If a range operation, the NUM field. If a single-address operation this is 0.

prot enum

The protection level for which this TLB Invalidate will operate on.

security_world enum

The security world that this will apply to.

smmu_scale unsigned int

If a range operation, then the SCALE field with the meaning in the SMMU architecture which is different to the PE architecture. If a single-address operation this is 0.

stage1_only bool

The operation is for stage 1 only if supported.

tg enum

If a single-address or address-range operation, then the Translation Granule hint. Address-range operations always supply a Translation Granule.

translation_table_level enum

The leaf level of the translation table.

vmid unsigned int

The VMID to match if match_vmid.

found_tlb_entry_has_different_aset

Architecturally, a particular ASID either should be ASET0 or ASET1. However, we have managed to find a TLB entry that has a different ASET than that which we were searching for. This indicates a programming error. You should examine all contexts with this particular ASID/VMID and ensure they are consistent. Fields:

aset unsigned int

The ASET of this entry.

asid unsigned int

ASID if appropriate.

index unsigned int

Index of TLB entry.

input_start_address unsigned int

The start address of the input range that this matches.

scheme enum

The tagging scheme used.

ssd enum

The SSD of the tagging scheme.

ssd_ns enum

Security State.

vmid unsigned int

VMID if appropriate.

gpf_far

An attempt was made to record a fault in SMMU_ROOT_GPF_FAR. Fields:

fault_address unsigned int

The PA input address that faulted.

fpas enum

The PAS of the input PA that suffered the fault.

is_debug bool

The record is due to a debug transaction.

lost bool

The record is lost because an existing record is active. If `is_debug`, then the record is never written, but this reports if it would have been lost if it was a real transaction.

syndrome enum

The syndrome for the fault.

gpt_cfg_far

An attempt was made to record a fault in `SMMU_ROOT_GPT_CFG_FAR`. Fields:

cfg_err enum

The configuration syndrome.

fault_address unsigned int

The PA input address that faulted.

fpas enum

The PAS of the input PA that suffered the fault.

is_debug bool

The record is due to a debug transaction.

lost bool

The record is lost because an existing record is active. If `is_debug`, then the record is never written, but this reports if it would have been lost if it was a real transaction.

syndrome enum

The syndrome for the fault.

gpt_read_block_descriptor

A GPT read has completed and found an LOGPT block descriptor. Fields:

address_checking unsigned int

The address that we are checking.

address_gptdesc unsigned int

The address of the GPT descriptor.

begin_address unsigned int

The address range this block covers.

data unsigned int

The data fetch if it didn't abort.

end_incl_address unsigned int

The address range this block covers.

gpi enum

GPI.

gpi_violation bool

The descriptor is invalid because the GPI value is invalid.

has_res0_violation bool

The descriptor is invalid because of **RES0** violation.

pas_checking enum

The PAS of the address that we are checking.

trans_id unsigned int

The trans_id of the transaction we are checking if appropriate.

what_checking enum

What kind of access are we checking.

gpt_read_contig_descriptor

A GPT read has completed and found an L1GPT contig descriptor. Fields:

address_checking unsigned int

The address that we are checking.

address_gptdesc unsigned int

The address of the GPT descriptor.

begin_address unsigned int

The address range this block covers.

data unsigned int

The data fetch if it didn't abort.

end_incl_address unsigned int

The address range this block covers.

gpi enum

GPI.

gpi_violation bool

The descriptor is invalid because the GPI value is invalid.

has_res0_violation bool

The descriptor is invalid because of **RES0** violation.

pas_checking enum

The PAS of the address that we are checking.

size_violation bool

The encoded size was not a valid value.

trans_id unsigned int

The trans_id of the transaction we are checking if appropriate.

what_checking enum

What kind of access are we checking.

gpt_read_granules_descriptor

A GPT read has completed and found an L1GPT granules descriptor. Fields:

address_checking unsigned int

The address that we are checking.

address_gptdesc unsigned int

The address of the GPT descriptor.

begin_address unsigned int

The address range this block covers.

data unsigned int

The data fetch if it didn't abort.

end_incl_address unsigned int

The address range this block covers.

gpi enum

GPI.

gpi_violation bool

The descriptor is invalid because the GPI value is invalid.

pas_checking enum

The PAS of the address that we are checking.

trans_id unsigned int

The trans_id of the transaction we are checking if appropriate.

what_checking enum

What kind of access are we checking.

which_gpi_granule unsigned int

Which GPI granule the address being checked will use.

gpt_read_table_descriptor

A GPT read has completed and found an LOGPT table descriptor. Fields:

address_checking unsigned int

The address that we are checking.

address_gptdesc unsigned int

The address of the GPT descriptor.

data unsigned int

The data fetch if it didn't abort.

has_res0_violation bool

The descriptor is invalid because of **RES0** violation.

misaligned_violation bool

The descriptor is invalid because the base address is misaligned.

out_of_range_violation bool

The descriptor is invalid because the base address is out of range of the PPS.

pas_checking enum

The PAS of the address that we are checking.

table_address unsigned int

The table address.

trans_id unsigned int

The trans_id of the transaction we are checking if appropriate.

what_checking enum

What kind of access are we checking.

gpttlb_entry_allocated

A GPT TLB entry has been allocated. Fields:

GPI enum

The GPI of the entry.

index unsigned int

Index of TLB entry.

input_end_incl_address unsigned int

The end inclusive address of the input range that this matches.

input_start_address unsigned int

The start address of the input range that this matches.

origin enum

How this entry was formed.

httu_update_abandoned_update

The HTTU update of a descriptor in memory was potentially possible, but it was behind an update that failed to apply cleanly. Fields:

AF enum

Whether an AF update should/could occur.

DBM enum

Whether a DBM update should/could occur.

address unsigned int

Address of HTTU update.

httu_inner enum

Inner cacheability of descriptor to update.

httu_outer enum

Outer cacheability of descriptor to update.

httu_prot enum

The instruction/data, bus-ns/s and privileged/user marking.

httu_sh enum

Shareability of descriptor to update.

mpam_pmg_and_partid unsigned int

Bits[23:16] are the MPAM PMG. Bits[15:0] are the physical PARTID.

mpam_sp enum

The MPAM_SP of the transaction.

original_descriptor unsigned int

The original descriptor that the model observed.

stage_and_level unsigned int

The stage (bits[7:4]) and level (bits[3:0]). A level of 0xF means -1.

trans_id unsigned int

The transaction id that this HTTU update is associated with.

what enum

What this descriptor represents.

httu_update_end_update

The attempted update of a descriptor in memory has occurred. Fields:

is_big_endian bool

The descriptor is big-endian in memory.

original_descriptor unsigned int

The original descriptor value.

result enum

The result of the attempt to update.

trans_id unsigned int

The transaction id that this HTTU update is associated with.

try_to_change_to_descriptor unsigned int

The new descriptor value that tried to replace the original.

value_that_was_in_memory unsigned int

The value that the compare-and-swap operation returned as the value that was in memory.

httu_update_not_done

A discretionary HTTU update could occur and the implementation choose not to do it. A mandatory_do_if_linked_performed means that this is a stage 2 leaf descriptor that maps a subsequent stage 1 leaf descriptor whose update is discretionary and if that subsequent discretionary update is going to occur then this update becomes mandatory. Fields:

AF enum

Whether an AF update should/could occur.

DBM enum

Whether a DBM update should/could occur.

address unsigned int

Address of HTTU update.

httu_inner enum

Inner cacheability of descriptor to update.

httu_outer enum

Outer cacheability of descriptor to update.

httu_prot enum

The instruction/data, bus-ns/s and privileged/user marking.

httu_sh enum

Shareability of descriptor to update.

mpam_pmg_and_partid unsigned int

Bits[23:16] are the MPAM PMG. Bits[15:0] are the physical PARTID.

mpam_sp enum

The MPAM_SP of the transaction.

original_descriptor unsigned int

The original descriptor that the model observed.

stage_and_level unsigned int

The stage (bits[7:4]) and level (bits[3:0]). A level of 0xF means -1.

trans_id unsigned int

The transaction id that this HTTU update is associated with.

what enum

What this descriptor represents.

httu_update_start_update

An HTTU update could occur and the implementation chose to try it. A mandatory_do_if_linked_performed means that this is a stage 2 leaf descriptor that maps a subsequent stage 1 leaf descriptor whose update is discretionary and if that subsequent discretionary update is going to occur then this update becomes mandatory. Fields:

AF enum

Whether an AF update should/could occur.

DBM enum

Whether a DBM update should/could occur.

address unsigned int

Address of HTTU update.

httu_inner enum

Inner cacheability of descriptor to update.

httu_outer enum

Outer cacheability of descriptor to update.

httu_prot enum

The instruction/data, bus-ns/s and privileged/user marking.

httu_sh enum

Shareability of descriptor to update.

is_big_endian bool

The descriptor will be written to memory as big-endian.

mecid unsigned int

The masked MECID used for the update transaction, or ~0u if not appropriate.

mpam_pmg_and_partid unsigned int

Bits[23:16] are the MPAM PMG. Bits[15:0] are the physical PARTID.

mpam_sp enum

The MPAM_SP of the transaction.

original_descriptor unsigned int

The original descriptor that the model observed.

stage_and_level unsigned int

The stage (bits[7:4]) and level (bits[3:0]). A level of 0xF means -1.

trans_id unsigned int

The transaction id that this HTTU update is associated with.

try_to_change_to_descriptor unsigned int

The new descriptor value that is going to try replace the original.

what enum

What this descriptor represents.

will_do_AF bool

What the implementation chose to do for the AF update.

will_do_DBM bool

What the implementation chose to do for the DBM update.

interrupt_returned

An interrupt/MSI returned from downstream. Fields:

InD_NS_PnU enum

Instruction/Data, Non-secure/Secure/Root/Realm, Privileged/User. Bit[1] = PAS[0], and bit[3] = PAS[1].

address unsigned int

Address of the MSI.

data unsigned int

Data payload of the MSI.

dcmdqcp_and_index unsigned int

If this not a DCMDQ MSI CMD_SYNC, then this is 0xFFFFFFFF. If it is then the DCMDQ QCP index is in bits[31:16] and the index in the QCP is in [7:0].

gpc_denied_msi bool

True if the MSI was denied as it failed its GPC checks. Thus the field 'ok' will be false.

id unsigned int

ID of this interrupt transaction.

inner_cache enum

The actual attributes of the access.

kind enum

What kind of interrupt.

mecid unsigned int

The MECID of the MSI, if applicable.

mpam_ns bool

For a non-RME system, this is true if the MPAM space is Non-secure. For an RME system, this is MPAM_SP[0].

mpam_pmg_and_partid unsigned int

Bits[23:16] are the MPAM PMG. Bits[15:0] are the physical PARTID.

mpam_sp enum

The MPAM_SP security state of the MPAM partition. 0 – secure, 1 – non-secure, 2 – root, 3 – realm.

msi_valid bool

MSI will attempt to be sent.

ok bool

Did the access return OK or an abort?.

outer_cache enum

The actual attributes of the access.

pas enum

The PAS of the MSI, if applicable.

pmcg_index_and_counter unsigned int

If this is a PMCG interrupt, then the top 16 bits are the PMCG index, the lower 16 bits are counter index.

ras_error_group_id_node_id_and_record_index_in_node unsigned int

If this is a RAS interrupt then this is the error group record id in bits[23:16], the node_id in [15:8] and the record index in the node in [7:0].

sh enum

Shareability.

interrupt_sent

An interrupt is raised. If it sends an MSI then this is *after* any device-dependent transform on the architectural attributes and so may differ from what is programmed. Fields:

InD_NS_PnU enum

Instruction/Data, Non-secure/Secure/Root/Realm, Privileged/User. Bit[1] = PAS[0], and bit[3] = PAS[1].

address unsigned int

Address of the MSI.

data unsigned int

Data payload of the MSI.

dcmdq_device_id unsigned int

If this is a DCMDQ then this is the DeviceID of the MSI write. Otherwise, 0xFFFF'ffff.

dcmdq_qcp_and_index unsigned int

If this not a DCMDQ MSI CMD_SYNC, then this is 0xFFFF'ffff. If it is then the DCMDQ QCP index is in bits[31:16] and the index in the QCP is in [7:0].

id unsigned int

ID of this interrupt transaction.

inner_cache enum

The actual attributes of the access.

kind enum

What kind of interrupt.

mecid unsigned int

The MECID of the MSI, if applicable.

mpam_ns bool

For a non-RME system, this is true if the MPAM space is Non-secure. For an RME system, this is MPAM_SP[0].

mpam_pmg_and_partid unsigned int

Bits[23:16] are the MPAM PMG. Bits[15:0] are the physical PARTID.

mpam_sp enum

The MPAM_SP security state of the MPAM partition. 0 – secure, 1 – non-secure, 2 – root, 3 – realm.

msi_valid bool

MSI will attempt to be sent.

outer_cache enum

The actual attributes of the access.

pas enum

The PAS of the MSI, if applicable.

pmcg_index_and_counter unsigned int

If this is a PMCG interrupt, then the top 16 bits are the PMCG index, the lower 16 bits are counter index.

qSID unsigned int

If this is a DCMDQ then this is the qSID of the MSI write. Otherwise, 0xFFFF'ffff.

ras_error_group_id_node_id_and_record_index_in_node unsigned int

If this is a RAS interrupt then this is the error group record id in bits[23:16], the node_id in [15:8] and the record index in the node in [7:0].

sh enum

Shareability.

l0gptsz

Log2size in bytes of the region covered by an LOGPT entry. The default value is `rme_l0gpt_entry_covers_log2size_in_bytes`. This is the value reported in `SMMU_ROOT_GPT_BASE_CFG.LOGPTSZ`. This pin is sampled on the negedge of the reset pin and has to be driven beforehand for the value to take effect. Fields:

value unsigned int

The value of the signal.

l1cd_cc.L1CD_commentary

A commentary from the cache about what is going on. Fields:

output string

The stream output.

l1cd_cc.L1CD_entry_invalidated

An entry in the cache has been invalidated. Fields:

entry_id unsigned int

The entry id that is being invalidated.

reason enum

Why the entry is being invalidated.

l1cd_entry_allocated

An L1 CD entry has been allocated. Fields:

AssuredTranslation bool

The L1CD was fetched from an AssuredOnly stage 2 page and so might be capable of creating AssuredTranslations.

entry_id unsigned int

The id of the cache entry. Top bit is set if is a debug entry.

normalised_substreamid unsigned int

The first SubstreamID of the range of SubstreamIDs that this L1CD entry will match.

ns enum

For the non-secure world.

pa_12 unsigned int

The PA of the L2 CD table. This is `L2Ptr << 12`.

ssd enum

The SSD of the entry.

streamid unsigned int

The StreamID this CD is for.

valid bool

Is the entry valid.

l1ste_cc.L1STE_commentary

A commentary from the cache about what is going on. Fields:

output string

The stream output.

l1ste_cc.L1STE_entry_invalidated

An entry in the cache has been invalidated. Fields:

entry_id unsigned int

The entry id that is being invalidated.

reason enum

Why the entry is being invalidated.

l1ste_entry_allocated

An L1 STE entry has been allocated. Fields:

entry_id unsigned int

The id of the cache entry. Top bit is set if is a debug entry.

first_streamid_of_range unsigned int

The first StreamID of the range of StreamIDs that this L1STE entry will match.

ns enum

For the non-secure world.

num_entries_in_l2 unsigned int

The number of entries in the L2 table. This is $2^{**}(\text{Span}-1)$ or 0 if invalid.

pa_l2 unsigned int

The PA of the L2 ST table. This is L2Ptr << 6 and aligned to the size of the table.

ssd enum

The SSD of the entry.

legacy_tz_en

The pin is driven. If high then the SMMU does not support RME. If low and the ID codes indicate support, then the SMMU supports RME. This pin is sampled on the negedge of the reset pin and has to be driven beforehand for the value to take effect. Fields:

value bool

The value of the signal.

level_interrupt_sent

A level interrupt changed state. Fields:

kind enum

What kind of interrupt.

ras_error_group_id_node_id_and_record_index_in_node unsigned int

If this is a level RAS interrupt then this is the error group record id in bits[23:16], the node_id in [15:8] and the record index in the node in [7:0].

set_high bool

Level interrupt state.

ns_cmd_sync_completed_irq

“Non-secure” CMD_SYNC completed its IRQ action. Fields:

cmd_id unsigned int

The ID of the command being issued.

cons_incl_wrap unsigned int

CONS pointer of CMD_SYNC.

is_synthetic bool

This is a synthetic sync. The cons_incl_wrap is 0xFFFF'ffff.

ns_cmd_sync_completed_no_action

Non-secure CMD_SYNC completed, there was no CMD_SYNC action (SIG_NONE). Fields:

cmd_id unsigned int

The ID of the command being issued.

cons_incl_wrap unsigned int

CONS pointer of CMD_SYNC.

is_synthetic bool

This is a synthetic sync. The cons_incl_wrap is 0xFFFF'ffff.

ns_cmd_sync_completed_sev

Non-secure CMD_SYNC completed its action of SIG_SEV. Fields:

cmd_id unsigned int

The ID of the command being issued.

cons_incl_wrap unsigned int

CONS pointer of CMD_SYNC.

is_synthetic bool

This is a synthetic sync. The cons_incl_wrap is 0xFFFF'ffff.

ns_cmd_sync_completed_with_error

Non-secure CMD_SYNC completed, however, there was an error associated with the completion of the CMD_SYNC. Fields:

error enum

The error associated with the CMD_SYNC.

cmd_id unsigned int

The ID of the command being issued.

cons_incl_wrap unsigned int

CONS pointer of CMD_SYNC.

is_synthetic bool

This is a synthetic sync. The cons_incl_wrap is 0xFFFF'ffff.

ns_cmd_sync_issued

Non-secure CMD_SYNC has now completed issuing and we now know if something is hazarding the CMD_SYNC from completing. Fields:

cmd_id unsigned int

The ID of the command being issued.

cons_incl_wrap unsigned int

CONS pointer of CMD_SYNC.

hazarded bool

The CMD_SYNC is currently hazarded from completing.

is_synthetic bool

This is a synthetic sync. The cons_incl_wrap is 0xFFFF'ffff.

ns_cmd_sync_issuing

Non-secure CMD_SYNC is starting to issue, we are going to start accounting for which things have to be part of the prior set of things that have to complete before the CMD_SYNC can complete. Fields:

cmd_id unsigned int

The ID of the command being issued.

cons_incl_wrap unsigned int

CONS pointer of CMD_SYNC.

is_synthetic bool

This is a synthetic sync. The cons_incl_wrap is 0xFFFF'ffff.

ns_cmd_sync_starting_completion_action_irq

Non-secure CMD_SYNC starting IRQ completion action. Fields:

cmd_id unsigned int

The ID of the command being issued.

cons_incl_wrap unsigned int

CONS pointer of CMD_SYNC.

is_synthetic bool

This is a synthetic sync. The cons_incl_wrap is 0xFFFF'ffff.

msiaddress unsigned int

The MSI address to use (or 0 if not sending an MSI).

msidata unsigned int

The MSI data to send.

ns_cmd_sync_starting_completion_action_sev

Non-secure CMD_SYNC starting its completion action of SIG_SEV. Fields:

cmd_id unsigned int

The ID of the command being issued.

cons_incl_wrap unsigned int

CONS pointer of CMD_SYNC.

is_synthetic bool

This is a synthetic sync. The cons_incl_wrap is 0xFFFF'ffff.

ns_cmdq_cmd_sync_error

The CMD_SYNC can fault if an ATC Invalidate timeout occurs. The timeout will be attached to the next CMD_SYNC on the queue on which the invalidate was launched. Fields:

cerror enum

The CERROR to attach.

cmd_queue unsigned int

The top bits indicate the CMDQ id. These bits are attached to all cmd_ids generated for this queue. The bits [63:44] contain the queue ID. [63:62] 0: s-ssd, 1: reserved, 2: ns-ssd. 3: rl-ssd [61] is ECMDQ [59:52] QCP index [51:44] ECMDQ index in QCP.

herror enum

The HERROR to attach, or HERROR_NONE if we are attaching a CERROR.

ns_cmdq_cmd_sync_error_delivered

The CMD_SYNC can fault if an ATC Invalidate timeout occurs. This is the CMD_SYNC delivering that fault. Fields:

cerror enum

The CERROR to attach.

cmd_id unsigned int

Command ID of the CMD_SYNC.

ns_cmdq_ecmdq_enack

Trace the changes to ENACK. Fields:

cerror enum

The error being exposed, if any.

cmd_queue unsigned int

The top bits indicate the CMDQ id. These bits are attached to all cmd_ids generated for this queue. The bits [63:44] contain the queue ID. [63:62] 0: s-ssd, 1: reserved, 2: ns-ssd. 3: rl-ssd [61] is ECMDQ [59:52] QCP index [51:44] ECMDQ index in QCP.

enack bool

The value of ENACK.

forced_error bool

The queue was enabled with ERR != ERRACK and so the implementation chooses to trigger the error reporting mechanism.

ns_cmdq_issue

A command is actually being executed. Fields:

cmd_id unsigned int

Command ID of the command being executed.

cons unsigned int

CONS of the command.

what string

What is being executed.

ns_cmdq_pointers_state

The command queue pointers. Fields:

cmd_queue unsigned int

The top bits indicate the CMDQ id. These bits are attached to all cmd_ids generated for this queue. The bits [63:44] contain the queue ID. [63:62] 0: s-ssd, 1: reserved, 2: ns-ssd. 3: rl-ssd [61] is ECMDQ [59:52] QCP index [51:44] ECMDQ index in QCP.

disable_fetch bool

The model has decided that as you programmed PROD/CONS inconsistently that it would not fetch as it has no confidence as to which are commands and which are not.

q_issuable_and_issued_cons unsigned int

The CONS pointer of the region that is issuable and/or issued (including wrap).

q_issuable_and_issued_prod unsigned int

The PROD pointer of the region that is issuable and/or issued (including wrap).

q_programmer_cons unsigned int

The CONS pointer as seen by the programmer (including wrap).

q_programmer_prod unsigned int

The PROD pointer as seen by the programmer (including wrap).

q_unfetched_cons unsigned int

The CONS pointer of the region unfetched (including wrap).

q_unfetched_prod unsigned int

The PROD pointer of the region unfetched (including wrap).

ns_cmdq_state

A description of what the CMDQ can do now. Fields:

can_cerror bool

There is a CERROR that can be recognised.

can_cmd_sync bool

There is a CMD_SYNC that can be recognised.

can_fetch bool

The SMMU can fetch.

can_herror bool

There is an ERROR that can be recognised.

can_issue bool

There are fetched commands that are eligible to be issued.

can_update_cons bool

The programmer view of CMDQ_CONS can be moved to say some have been consumed.

cmd_queue unsigned int

The top bits indicate the CMDQ id. These bits are attached to all cmd_ids generated for this queue. The bits [63:44] contain the queue ID. [63:62] 0: s-ssd, 1: reserved, 2: ns-ssd. 3: rl-ssd [61] is ECMDQ [59:52] QCP index [51:44] ECMDQ index in QCP.

cmd_sync enum

Why a CMD_SYNC cannot be issued.

cmdqen bool

CMDQEN, when enabled then the CMDQ can fetch commands.

cmdqen_1_to_0_unacknowledged bool

CMDQEN has been set to 0, but we have yet to acknowledge it.

current-real-error enum

The current ERROR that would be visible to the programmer if there was an error.

fetch-disabled_due_to_programmer_error bool

The PROD/CONS pointer was inconsistent and the model decided to stop processing commands.

fetch-number_of_unfetched_commands unsigned int

The number of commands that have yet to be fetched from memory, or ~0u if CMDQEN == 0.

fetch-state enum

The state of the fetch state machine.

fetch-too_many_commands_outstanding_so_not_fetching_any_more bool

If there are too many issued commands, or pending to be issued then we won't fetch any more.

issue-number_of_issuable_commands unsigned int

The number of commands that have been fetched but not yet issued.

issue-number_of_unissuable_commands unsigned int

This might include commands behind an illegal command, or behind a CMD_SYNC. Under some circumstances this can include a command representing an external abort.

issue-state enum

The internal issued state machine state.

outstanding_fetches_since_reset unsigned int

The number of outstanding fetches.

synthetic_sync_can_be_completed bool

The synthetic sync that is outstanding can be completed.

waiting_for_synthetic_sync bool

There is a synthetic sync that is outstanding.

ns_eventq_adding_to_cmd_sync_prior_set

The event created by the specified transaction id must become visible to the programmer before the CMD_SYNC can complete. Fields:

trans_id unsigned int

The transaction id that caused the event.

ns_eventq_auto_retry_stalled_transaction

A transaction that was stalled but not yet reported on the Event Queue was retried. The retry can happen if the transaction didn't write because it was de-duplicated behind a reported stalling transaction and that stalling transaction was CMD_RESUMEd, or if the queue was unwriteable and the queue is now writeable. Fields:

trans_id unsigned int

Transaction id of the auto-retried transaction.

why enum

Why the transaction retried.

ns_eventq_cmd_sync_unhazarded

The CMD_SYNC has been unhazarded as the prior set is empty and as much of it as required is visible in the programmers' view. Fields:

prod_incl_wrap unsigned int

The prod index including wrap bit that was required to become visible.

prog_prod_incl_wrap unsigned int

The current programmer-visible prod including wrap bit.

ns_eventq_eventqen_and_eventqenack

Trace the values of EVENTQEN and EVENTQENACK as they change. Fields:

about enum

Some more information about the queue state.

cons_incl_wrap unsigned int

The CONS pointer (including wrap bits).

eventqen bool

EVENTQEN.

eventqenack bool

EVENTQENACK.

prod_incl_wrap unsigned int

The PROD pointer (including wrap bits).

ns_eventq_external_abort

A particular eventq record got an external abort. The event will subsequently appear in the ns/s_eventq_lost_event_records trace. Fields:

prod unsigned int

The prod pointer (excluding wrap bit) of the event record that aborted.

trans_id unsigned int

The transaction id that produced the event record that aborted.

ns_eventq_lost_event_records

Event records that are lost and never record appear on this trace. Fields:

dword0 unsigned int

Double word 0 of event.

dword1 unsigned int

Double word 1 of event.

dword2 unsigned int

Double word 2 of event.

dword3 unsigned int

Double word 3 of event.

event enum

Event code that was thrown away.

prod unsigned int

The PROD index of the event if appropriate or ~0u otherwise.

reason enum

The reason the record was lost.

trans_id unsigned int

The transaction id that this corresponds to.

ns_eventq_overflow_acknowledged

When events are lost then they appear on the eventq_lost_event_records_* trace. This will activate the overflow flag in the corresponding SMMU_s_EVENTQ_PROD register. Software can then acknowledge that flag using the overflow acknowledge flag in SMMU_s_EVENTQ_CONS. Fields:

bad bool

If true then the programmer has attempted to acknowledge an overflow condition that didn't exist.

count_of_events_lost_due_to_overflow unsigned int

This is the number of events that were lost because the event queue was full. This does not include any lost because of an EVENTQ_ABT_ERR.

ns_eventq_pending_event_records

Event records that are pending to be dealt with by the eventq state machine. A specific event can be traced multiple times if it becomes part of the prior set of transactions of a CMD_SYNC that has to complete before the CMD_SYNC is allowed to be completed. Fields:

dword0 unsigned int

Double word 0 of event.

dword1 unsigned int

Double word 1 of event.

dword2 unsigned int

Double word 2 of event.

dword3 unsigned int

Double word 3 of event.

event enum

Event code.

is_prior_set bool

The transaction forms part of the prior set of a CMD_SYNC that have to become visible (or thrown away if necessary) before the CMD_SYNC can complete.

trans_id unsigned int

The transaction id that this corresponds to.

ns_eventq_prod_cons_updated

This records when the programmer visible PROD/CONS pointer is updated. More can be visible in memory than are advertised in the programmer-visible PROD pointer. This also traces the latest PROD that could have been advertised. Fields:

mem_prod unsigned int

The PROD pointer value for all records that are visible in the memory.

num_events_available unsigned int

The number of events available to the programmer.

prog_cons unsigned int

The programmer-visible CONS pointer.

prog_prod unsigned int

The programmer-visible PROD pointer.

what enum

What changed?.

ns_eventq_raising_eventq_abt_err

EVENTQ_ABT_ERR can now be raised.

ns_eventq_removing_from_cmd_sync_prior_set

The event captured by a CMD_SYNC and created by the specified transaction id is now in memory. Fields:

prod unsigned int

The prod index (no wrap bit) that must become visible to the programmer to see this record. If not head_of_line then the record will not become visible and this field is meaningless.

reason enum

The reason the event was removed.

trans_id unsigned int

The transaction id that caused the event.

ns_eventq_stashing_unreported_stalled_transaction_for_auto_retry

A stall event record would have been lost, instead it is stashed away to auto-retry when it can. Fields:

dword0 unsigned int

Double word 0 of event.

dword1 unsigned int

Double word 1 of event.

dword2 unsigned int

Double word 2 of event.

dword3 unsigned int

Double word 3 of event.

event enum

Event code that was thrown away.

reason enum

The reason the record was lost.

trans_id unsigned int

The transaction id that this corresponds to.

ns_eventq_write_event

An event is committed to being written. Fields:

dword0 unsigned int

Double word 0 of event.

dword1 unsigned int

Double word 1 of event.

dword2 unsigned int

Double word 2 of event.

dword3 unsigned int

Double word 3 of event.

event enum

Event code that was thrown away.

prod unsigned int

The PROD pointer (without wrap) that this event will be written to.

stall_event bool

Describes an event for a stalled transaction.

trans_id unsigned int

Transaction ID that generated this event.

ns_eventq_write_event_visible_in_memory

An event is now visible in memory. Fields:

abort bool

The write aborted.

dword0 unsigned int

Double word 0 of event.

dword1 unsigned int

Double word 1 of event.

dword2 unsigned int

Double word 2 of event.

dword3 unsigned int

Double word 3 of event.

event enum

Event code that was thrown away.

prod unsigned int

The PROD pointer (without wrap) that this event will be written to.

stall_event bool

Describes an event for a stalled transaction.

trans_id unsigned int

Transaction ID that generated this event.

ns_gerror_inverted

A GERROR bit was attempted to be inverted to record a fault. Fields:

already_different_to_gerrorn bool

True if the bit was already different to the corresponding bit in GERRORN and so it was not inverted.

gerrorn unsigned int

GERRORN register value.

interrupt_potentially_sent bool

An interrupt is potentially sent, depending on IRQEN.
GERROR(N).MSI_GERROR_ABT_ERR does not send an interrupt.

new_gerror unsigned int

GERROR register value after any inversion.

which enum

Which bit was attempted to be inverted.

ns_gerrorn_acknowledge

A GERROR was acknowledged by SW. Fields:

which enum

Which GERROR was acknowledged.

pmcg_irq_config

The interrupt configuration of the Performance Monitor Counter Group (PMCG) changed.
Fields:

address unsigned int

Address of the MSI.

data unsigned int

Data payload of the MSI.

irqen bool

SMMU_PMCGL_CTRL.IRQEN.

memattr enum

Memory type.

mpam_ns bool

The NS state of the MPAM PARTID and MPAM PMG.

mpam_pmg_and_partid unsigned int

Bits[23:16] are the MPAM PMG. Bits[15:0] are the physical PARTID.

msi_supported bool

Are MSIs supported for this security world.

ns bool

Non-secure bus attribute.

number_of_interrupts_in_flight unsigned int

The number of interrupts that have been committed to be produced or in flight.

pmcg_index unsigned int

Index of the PMCG.

sh enum

Shareability.

smmu_pmcg_gmpam_Update bool

The SMMU_PMC_GMPAM.Update flag. Only when this is zero are writes predictable.

ssd_ns bool

True if this PMCG controlled by the non-secure world.

pmcg_irqen

A trace of SMMU_PMC_IRQ_CTRL.IRQEN. Fields:

new_value bool

The new value of IRQEN.

pmcg_index unsigned int

Index of the PMCG.

ssd_ns bool

True if this PMCG controlled by the non-secure world.

pmcg_irqen_ack

A trace of SMMU_PMC_IRQ_CTRLACK.IRQEN. Fields:

new_value bool

The new value of IRQEN.

pmcg_index unsigned int

Index of the PMCG.

ssd_ns bool

True if this PMCG controlled by the non-secure world.

pmcg_merging_interrupts

An interrupt was wanted to be generated, but one was already pending so the two were merged together. Fields:

pmcg_index unsigned int

Index of the PMCG.

ssd_ns bool

True if this PMCG controlled by the non-secure world.

pmcg_trigger

A PMCG counter has been triggered. Fields:

NoStreamID bool

True if the transaction is a NoStreamID transaction.

counter_index unsigned int

The index of the counter within the PMCG.

event_id unsigned int

The event id that has been triggered.

ns_event bool

Is the event associated with non-secure state.

pmcg_index unsigned int

Index of the PMCG.

prior_counter_value unsigned int

The Counter value *before* the event has incremented it.

ssd enum

The security state associated with the event.

streamid unsigned int

The StreamID associated with the event, if there is one.

tbu_index unsigned int

The TBU index of the transaction, or ~0u if not applicable.

pmu_active_counter

Traces what active counters are in a PMCG and what StreamIDs it might filter on. Those counters that trace StreamIDs for multiple security states, or those that are not filtered by StreamID, will appear multiple times, once for each security state. All active counters for a PMCG are traced one after another. Fields:

NoStreamID bool

True if NoStreamID transactions will be traced.

begin_streamid unsigned int

The start StreamID to filter on.

counter_index unsigned int

The counter index within the PMCG.

end_incl_streamid unsigned int

The end inclusive StreamID to filter on.

evcnt unsigned int

The current count.

event_id unsigned int

The event ID to filter.

ns bool

Are the StreamIDs non-secure?.

pmcg_index unsigned int

The index of the PMCG that this counter belongs to.

ssd enum

SSD of the StreamID.

tbu_index_to_match unsigned int

The TBU index that must match, or ~0u if no matching applicable.

pmu_all_counters_in_pmcg_became_inactive

The PMCG was tracing some events and now is not tracing any. Fields:

pmcg_index unsigned int

The index of the PMCG.

pmu_capture

For some reason, a capture event occurred. Fields:

pmcg_index unsigned int

The index of the PMCG that the capture occurred on.

why enum

Why did the capture occur?.

pmu_counter_configured_to_use_unsupported_event

An enabled counter was configured to use a unsupported event. Fields:

counter_index unsigned int

The counter index within the PMCG.

event_id unsigned int

The unsupported event id.

pmcg_index unsigned int

The index of the PMCG that this counter belongs to.

pmu_counter_overflowed

A counter in a particular PMCG overflowed. Fields:

already_overflowed bool

True if the overflow flag was already set.

capture bool

True if it captured the other counter values.

counter_index unsigned int

The counter index within the PMCG.

interrupt bool

True if going to attempt to generate an interrupt.

interrupt_action enum

The interrupt action that is going to occur.

pmcg_index unsigned int

The index of the PMCG that this counter belongs to.

ssd_ns bool

The PMCG is controlled by the Non-secure security state.

priq_auto_response

A PRIQ auto response is generated. Fields:

prgindex unsigned int

The PRG Index of the response.

resp enum

The response to send.

streamid unsigned int

The StreamID of the response.

substreamid unsigned int

The PASID/SubstreamID attached to the response, or ~0u if there is none.

trans_id unsigned int

The ID of the PPR that is being lost.

why enum

Why the PPR is being lost.

priq_lost_ppr

The PRIQ is not reporting a PPR into the queue. It may or may not generate an auto-response. Fields:

is_stop_marker bool

The PPR being lost is a Stop Marker.

resp enum

What, if any, auto-response will be.

trans_id unsigned int

The ID of the PPR that is being lost.

why enum

Why the PPR is being lost.

priq_overflow_acking

Indicates that an overflow condition was acknowledged by writing to:- SMMU_PRIQ_CONS.OVACKFLG. Fields:

new_ovackflg bool

The new value of the SMMU_PRIQ_CONS.OVACKFLG.

priq_overflow_asserting

Indicates that we are toggling the SMMU_PRIQ_PROD.OVFLG because we lost a PRI request due to the PRIQ being full and an existing overflow condition does not already exist. Fields:

new_ovflg bool

The new value of the SMMU_PRIQ_PROD.OVFLG.

trans_id unsigned int

The transaction ID of the PPR that caused the overflow.

priq_received

A PRIQ Request has been received and is queued waiting processing. This does not mean that it has been written to the PRIQ but has been placed in a pending queue awaiting a decision about what to do about it. Fields:

L bool

The Last in the Page Request Group (PRG).

R bool

Read permission is requested.

W bool

Read permission is requested.

prgindex unsigned int

The Page Request Group (PRG) index that is being used to identify this request to the EndPoint.

stop_marker bool

If LRW == 0b100 then this indicates this is a stop marker.

streamid unsigned int

The StreamID this request is associated with.

substreamid unsigned int

The SubstreamID (PASID) this request is associated with, of ~0u if no SubstreamID.

substreamid_P bool

If has a SubstreamID (!= ~0u) then Privileged Mode Requested.

substreamid_X bool

If has a SubstreamID (!= ~0u) then Execution Requested (R must also be 1).

trans_id unsigned int

The transaction ID that identifies this request.

untranslated_address unsigned int

The Untranslated Address that this is requesting.

priq_state

The trace of various fields that indicate the state of the PRIQ. Fields:

cons_incl_wrap unsigned int

The value of SMMU_PRIQ.CONS.RD_and_RD_wrap.

number_of_pprs unsigned int

The number of PPRs as indicated by the CONS/PROD.

number_of_pprs_still_to_deal_with unsigned int

This is the number of PPRs that are currently waiting to either be written to the PRIQ, or auto-responded to.

number_of_priq_writes_in_flight unsigned int

The number of writes to the PRIQ that are currently in flight.

ovackflg bool

The OVACKFLG which if different to OVFLG is used to indicate that the PRIQ overflowed.

ovflg bool

The OVFLG which if different to OVACKFLG is used to indicate that the PRIQ overflowed.

priq_abt_err bool

There is an active SMMU_GERROR{N}.PRIQ_ABT_ERR.

priqen bool

The value of SMMU_CRO.PRIQEN. The *effective* value is 0 if SMMUEN == 0.

priqenack bool

The value of SMMU_CROACK.PRIQEN.

prod_incl_wrap unsigned int

The value of SMMU_PRIQ.CONS.WR_and_WR_wrap.

queue_disabled_due_to_prior_programming_error bool

The queue was disabled as the programmer got CONS/PROD into an inconsistent state. The model will disable the PRIQ until SW disables and re-enables the queue via SMMU_CRO.PRIQEN.

smmuen bool

The value of SMMU_CRO.SMMUEN. If this is 0 then the effective PRIQEN is 0.

smmuenack bool

The value of SMMU_CRO.SMMUEN. If this is 0 then the effective PRIQEN is 0.

table_size_in_elements unsigned int

The size of the table in the number of items it can hold.

priq_write_aborted

A PRIQ write aborted. The PRIQ now goes into an error state and will start auto-responding to PRI requests. Fields:

prod_incl_wrap unsigned int

The PROD including the Wrap bit where we were trying to write.

trans_id unsigned int

The transaction ID of the PPR that aborted.

priq_write_ok

A PRIQ write completed OK. Fields:

prod_incl_wrap unsigned int

The PROD including the Wrap bit where we wrote..

trans_id unsigned int

The transaction ID of the PPR.

priq_write_start

A PRIQ request has been received and is going to be attempt to be written to the queue.
Fields:

L bool

The Last in the Page Request Group (PRG).

R bool

Read permission is requested.

W bool

Read permission is requested.

prgindex unsigned int

The Page Request Group (PRG) index that is being used to identify this request to the EndPoint.

prod_incl_wrap unsigned int

PROD position including the wrap bit.

stop_marker bool

If LRW == 0b100 then this indicates this is a stop marker.

streamid unsigned int

The StreamID this request is associated with.

substreamid unsigned int

The SubstreamID (PASID) this request is associated with, of ~0u if no SubstreamID.

substreamid_P bool

If has a SubstreamID (!= ~0u) then Privileged Mode Requested.

substreamid_X bool

If has a SubstreamID (!= ~0u) then Execution Requested (R must also be 1).

trans_id unsigned int

The transaction ID that identifies this request.

untranslated_address unsigned int

The Untranslated Address that this is requesting.

ptw_read

Page Table Walk (read). This is the result of the physical access that the SMMU is making.
Fields:

abort enum

Non-zero if the access aborted/failed.

adomain enum

The actual attributes of the access that was used, after IMP DEF mangling.

aprot enum

The actual attributes of the access that was used, after IMP DEF mangling. PAS[1] (bit[3]), Instruction/Data (bit[2]), PAS[0] (bit[1]), Privileged/User (bit[0]).

data unsigned int

The data fetch if it didn't abort.

desckind enum

The kind of descriptor.

inner_cache enum

The actual attributes of the access that was used, after IMP DEF mangling.

input_address unsigned int

Input address this lookup is for (after TBI normalisation).

mecid unsigned int

The MECID of the TTD fetch.

ns enum

Non-Secure on the bus.

outer_cache enum

The actual attributes of the access that was used, after IMP DEF mangling.

pa_address unsigned int

Physical address.

pas enum

The PAS of the TTD fetch.

ssd enum

SSD of the transaction.

ssd_ns enum

Security state of stream.

streamid unsigned int

The StreamID this is for.

substreamid unsigned int

The SubstreamID this is for.

trans_id unsigned int

Transaction id.

ttb_grain_stage_and_level unsigned int

Which TTB used in bits[31:24]. Which Grain size used in bit[23:16]:- 0 – 4 KiB 1 – 16 KiB 2 – 64 KiB 3 – 64 KiB ARMv8.2 where descriptor bits [15:12] for bits [51:48] of output address 4 – 4 KiB FEAT_LPA2 where address is {bits[9:8], bits[49:12]} 5 – 16 KiB FEAT_LPA2 where address is {bits[9:8], bits[49:14], 2'b00} 6 – 64 KiB ARMv8.2 where descriptor bits [15:12] are ignored Stage in bits[15:8]. Level in bits[7:0], level -1 is 0xF.

ptw_read_st1_invalid_long_descriptor

Page Table Walk (read). The descriptor that the PTW fetched is invalid. Fields:

desckind enum

The kind of descriptor.

input_address unsigned int

Input address this lookup is for (after TBI normalisation).

mecid unsigned int

The MECID of the TTD fetch.

ns enum

Non-Secure on the bus.

pa_address unsigned int

Physical address.

pas enum

The PAS of the TTD fetch.

ssd enum

SSD of the transaction.

ssd_ns enum

Security state of stream.

streamid unsigned int

The StreamID this is for.

substreamid unsigned int

The SubstreamID this is for.

trans_id unsigned int

Transaction id.

ttb_grain_stage_and_level unsigned int

Which TTB used in bits[31:24]. Which Grain size used in bit[23:16]:- 0 – 4 KiB 1 – 16 KiB 2 – 64 KiB 3 – 64 KiB ARMv8.2 where descriptor bits [15:12] for bits [51:48] of output address 4 – 4 KiB FEAT_LPA2 where address is {bits[9:8], bits[49:12]} 5 – 16 KiB FEAT_LPA2 where address is {bits[9:8], bits[49:14], 2'b00} 6 – 64 KiB ARMv8.2 where descriptor bits [15:12] are ignored Stage in bits[15:8]. Level in bits[7:0], level -1 is 0xF.

ptw_read_st1_leaf_long_descriptor

Page Table Walk (read). The descriptor that the PTW fetched is a block or page and this is the decode. Fields:

AF bool

Access Flag.

AMEC enum

Use Primary or Alternative MECID.

AP21 enum

The access permissions.

AttrIdx210 unsigned int

The attribute index into the MAIR0/1. If AIE is implemented then this is the full index AttrIdx[3:0].

DBM bool

Dirty Bit Modifier. This only has meaning if running in AArch64 mode.

NS bool

The encoding is for non-secure if this is a secure fetch.

PIE_Dirty enum

The Dirty bit if S2PIE is in use.

PIE_PIIIndex unsigned int

The S2PIIndex if S2PIE is in use, or 0xFFFF if not.

PXN bool

Privileged eXecute Never.

Protected enum

Is the descriptor producing an AssuredTranslation.

SH10 enum

The shareability.

XN bool

eXecute Never.

contiguous bool

Contiguous hint.

desckind enum

The kind of descriptor.

hwu_pbha unsigned int

Top four bits are appropriate CD.HWU, *bottom bits[62:59] of descriptor. Page Based Hardware Attributes: only valid on a bit-per-bit basis enabled by CD.HWU0/CD.HWU1**.

input_address unsigned int

Input address this lookup is for (after TBI normalisation).

mecid unsigned int

The MECID of the TTD fetch.

nG bool

not Global.

nT bool

SMMUv3.2: If true then do not cache this entry in such a way that it can cause a TLB conflict abort (F_TLB_CONFLICT). The entry must produce a consistent result.

ns enum

Non-Secure on the bus.

output_address unsigned int

Output address.

pa_address unsigned int

Physical address.

pas enum

The PAS of the TTD fetch.

ssd enum

SSD of the transaction.

ssd_ns enum

Security state of stream.

streamid unsigned int

The StreamID this is for.

substreamid unsigned int

The SubstreamID this is for.

trans_id unsigned int

Transaction id.

ttb_grain_stage_and_level unsigned int

Which TTB used in bits[31:24]. Which Grain size used in bit[23:16]:- 0 – 4 KiB 1 – 16 KiB 2 – 64 KiB 3 – 64 KiB ARMv8.2 where descriptor bits [15:12] for bits [51:48] of output address 4 – 4 KiB FEAT_LPA2 where address is {bits[9:8], bits[49:12]} 5 – 16 KiB FEAT_LPA2 where address is {bits[9:8], bits[49:14], 2'b00} 6 – 64 KiB ARMv8.2 where descriptor bits [15:12] are ignored Stage in bits[15:8]. Level in bits[7:0], level -1 is 0xF.

ptw_read_st1_table_long_descriptor

Page Table Walk (read). The descriptor that the PTW fetched is a Table, this decodes the fields. Fields:

AF enum

.

APTable enum

Remove permissions independently of subsequent descriptors.

NSTable bool

The next level table descriptor is forced to non-secure.

PXNTable bool

Force PXN independently of subsequent descriptors.

Protected enum

Is the descriptor capable of producing an AssuredTranslation.

TableAddress unsigned int

Address of the next table.

XNTable bool

Force XN independently of subsequent descriptors.

desckind enum

The kind of descriptor.

input_address unsigned int

Input address this lookup is for (after TBI normalisation).

mecid unsigned int

The MECID of the TTD fetch.

ns enum

Non-Secure on the bus.

pa_address unsigned int

Physical address.

pas enum

The PAS of the TTD fetch.

ssd enum

SSD of the transaction.

ssd_ns enum

Security state of stream.

streamid unsigned int

The StreamID this is for.

substreamid unsigned int

The SubstreamID this is for.

trans_id unsigned int

Transaction id.

ttb_grain_stage_and_level unsigned int

Which TTB used in bits[31:24]. Which Grain size used in bit[23:16]:- 0 – 4 KiB 1 – 16 KiB 2 – 64 KiB 3 – 64 KiB ARMv8.2 where descriptor bits [15:12] for bits [51:48] of output address 4 – 4 KiB FEAT_LPA2 where address is {bits[9:8], bits[49:12]} 5 – 16 KiB FEAT_LPA2 where address is {bits[9:8], bits[49:14], 2'b00} 6 – 64 KiB ARMv8.2 where descriptor bits [15:12] are ignored Stage in bits[15:8]. Level in bits[7:0], level -1 is 0xF.

ptw_read_st2_invalid_descriptor

Page Table Walk (read). The descriptor that the PTW fetched is invalid. Fields:

desckind enum

The kind of descriptor.

input_address unsigned int

Input address this lookup is for (after TBI normalisation).

mecid unsigned int

The MECID of the TTD fetch.

ns enum

Non-Secure on the bus.

pa_address unsigned int

Physical address.

pas enum

The PAS of the TTD fetch.

ssd enum

SSD of the transaction.

ssd_ns enum

Security state of stream.

streamid unsigned int

The StreamID this is for.

substreamid unsigned int

The SubstreamID this is for.

trans_id unsigned int

Transaction id.

ttdb_grain_stage_and_level unsigned int

Which TTB used in bits[31:24]. Which Grain size used in bit[23:16]:- 0 – 4 KiB 1 – 16 KiB 2 – 64 KiB 3 – 64 KiB ARMv8.2 where descriptor bits [15:12] for bits [51:48] of output address 4 – 4 KiB FEAT_LPA2 where address is {bits[9:8], bits[49:12]} 5 – 16 KiB FEAT_LPA2 where address is {bits[9:8], bits[49:14], 2'b00} 6 – 64 KiB ARMv8.2 where descriptor bits [15:12] are ignored Stage in bits[15:8]. Level in bits[7:0], level -1 is 0xF.

ptw_read_st2_leaf_descriptor

Page Table Walk (read). The descriptor that the PTW fetched is a block or page and this is the decode. Fields:

AF bool

Access Flag.

AMEC enum

Use Primary or Alternative MECID.

AssuredOnly enum

The descriptor is marked as AssuredOnly.

DBM bool

Dirty Bit Modifier. This only has meaning if running in AArch64 mode.

HAP21 enum

The access permissions.

MemAttr3_0 enum

The memory attributes.

NS enum

Whether this descriptor forces NS.

PIE_Dirty enum

The Dirty bit if S2PIE is in use.

PIE_PIndex unsigned int

The S2PIndex if S2PIE is in use, or 0xFFFF if not.

POE_POIndex unsigned int

The S2POIndex if S2POE is in use, or 0xFFFF if not.

SH10 enum

The shareability.

XN enum

eXecute Never.

contiguous bool

Contiguous hint.

desckind enum

The kind of descriptor.

input_address unsigned int

Input address this lookup is for (after TBI normalisation).

mecid unsigned int

The MECID of the TTD fetch.

nT bool

SMMUv3.2: If true then do not cache this entry in such a way that it can cause a TLB conflict abort (F_TLB_CONFLICT). The entry must produce a consistent result.

ns enum

Non-Secure on the bus.

output_address unsigned int

Output address.

pa_address unsigned int

Physical address.

pas enum

The PAS of the TTD fetch.

s2hwu_pbha unsigned int

Top four bits are STE.S2HWU, *bottom bits[62:59] of descriptor. Page Based Hardware Attributes: only valid on a bit-per-bit basis enabled by STE.S2HWU.*

ssd enum

SSD of the transaction.

ssd_ns enum

Security state of stream.

streamid unsigned int

The StreamID this is for.

substreamid unsigned int

The SubstreamID this is for.

trans_id unsigned int

Transaction id.

ttb_grain_stage_and_level unsigned int

Which TTB used in bits[31:24]. Which Grain size used in bit[23:16]:- 0 – 4 KiB 1 – 16 KiB 2 – 64 KiB 3 – 64 KiB ARMv8.2 where descriptor bits [15:12] for bits [51:48] of output address 4 – 4 KiB FEAT_LPA2 where address is {bits[9:8], bits[49:12]} 5 – 16 KiB FEAT_LPA2 where address is {bits[9:8], bits[49:14], 2'b00} 6 – 64 KiB ARMv8.2 where descriptor bits [15:12] are ignored Stage in bits[15:8]. Level in bits[7:0], level -1 is 0xF.

ptw_read_st2_table_descriptor

Page Table Walk (read). The descriptor that the PTW fetched is a Table, this decodes the fields. Fields:

AF enum

.

APTable enum

Remove permissions independently of subsequent descriptors.

PXNTable bool

Force PXN independently of subsequent descriptors.

TableAddress unsigned int

Address of the next table.

XNTable bool

Force XN independently of subsequent descriptors.

desckind enum

The kind of descriptor.

input_address unsigned int

Input address this lookup is for (after TBI normalisation).

mecid unsigned int

The MECID of the TTD fetch.

ns enum

Non-Secure on the bus.

pa_address unsigned int

Physical address.

pas enum

The PAS of the TTD fetch.

ssd enum

SSD of the transaction.

ssd_ns enum

Security state of stream.

streamid unsigned int

The StreamID this is for.

substreamid unsigned int

The SubstreamID this is for.

trans_id unsigned int

Transaction id.

ttb_grain_stage_and_level unsigned int

Which TTB used in bits[31:24]. Which Grain size used in bit[23:16]:- 0 – 4 KiB 1 – 16 KiB 2 – 64 KiB 3 – 64 KiB ARMv8.2 where descriptor bits [15:12] for bits [51:48] of output address 4 – 4 KiB FEAT_LPA2 where address is {bits[9:8], bits[49:12]} 5 – 16 KiB FEAT_LPA2 where address is {bits[9:8], bits[49:14], 2'b00} 6 – 64 KiB ARMv8.2 where descriptor bits [15:12] are ignored Stage in bits[15:8]. Level in bits[7:0], level -1 is 0xF.

raw_register_end_read

The raw register read transaction. This is the transaction as directed to the register port.
Fields:

address unsigned int

The address of the transaction.

id unsigned int

Id of this transaction.

ok bool

Was the access OK? The bus response will always be OK, but is the register access reasonable.

pas enum

The PAS of the transaction.

read_data unsigned int

The data read.

raw_register_end_write

The raw register write transaction. This is the transaction as directed to the register port.
Fields:

address unsigned int

The address of the transaction.

id unsigned int

Id of this transaction.

ok bool

Was the access OK? The bus response will always be OK, but is the register access reasonable.

pas enum

The PAS of the transaction.

raw_register_start_read

The raw register read transaction. This is the transaction as directed to the register port.
Fields:

address unsigned int

Address of transaction.

id unsigned int

Id of this transaction. Top bit is set if is a debug transaction.

ns enum

Is the transaction non-secure?.

number_of_beats unsigned int

Number of beats.

pas enum

The PAS of the transaction.

size_of_beat_in_bytes unsigned int

Size of each beat in bytes.

raw_register_start_write

The raw register write transaction. This is the transaction as directed to the register port.
Fields:

address unsigned int

Address of transaction.

id unsigned int

Id of this transaction. Top bit is set if is a debug transaction.

ns enum

Is the transaction non-secure?.

number_of_beats unsigned int

Number of beats.

pas enum

The PAS of the transaction.

size_of_beat_in_bytes unsigned int

Size of each beat in bytes.

write_data unsigned int

The data to write.

register_disallowed_read_string

A text representation of the read of a register that was disallowed. Fields:

out string

The text description of the register value read.

register_disallowed_write_string

A text representation of the write of a register write that was disallowed. Fields:

in string

The text description of the register value written.

register_read_reserved

A text representation of an access to a register address that is reserved. Fields:

in string

The text description of the register value.

register_read_string

A text representation of the read of a register. Fields:

out string

The text description of the register value read.

register_write_reserved

A text representation of an access to a register address that is reserved or a write to a **RES0** field in a register. Fields:

in string

The text description of the register value.

register_write_string

A text representation of the write of a register. Fields:

in string

The text description of the register value written.

rl_cmd_sync_completed_irq

“Realm” CMD_SYNC completed its IRQ action. Fields:

cmd_id unsigned int

The ID of the command being issued.

cons_incl_wrap unsigned int

CONS pointer of CMD_SYNC.

is_synthetic bool

This is a synthetic sync. The cons_incl_wrap is 0xFFFF'ffff.

rl_cmd_sync_completed_no_action

Realm CMD_SYNC completed, there was no CMD_SYNC action (SIG_NONE). Fields:

cmd_id unsigned int

The ID of the command being issued.

cons_incl_wrap unsigned int

CONS pointer of CMD_SYNC.

is_synthetic bool

This is a synthetic sync. The cons_incl_wrap is 0xFFFF'ffff.

rl_cmd_sync_completed_sev

Realm CMD_SYNC completed its action of SIG_SEV. Fields:

cmd_id unsigned int

The ID of the command being issued.

cons_incl_wrap unsigned int

CONS pointer of CMD_SYNC.

is_synthetic bool

This is a synthetic sync. The cons_incl_wrap is 0xFFFF'ffff.

rl_cmd_sync_completed_with_error

Realm CMD_SYNC completed, however, there was an error associated with the completion of the CMD_SYNC. Fields:

error enum

The error associated with the CMD_SYNC.

cmd_id unsigned int

The ID of the command being issued.

cons_incl_wrap unsigned int

CONS pointer of CMD_SYNC.

is_synthetic bool

This is a synthetic sync. The cons_incl_wrap is 0xFFFF'ffff.

rl_cmd_sync_issued

Realm CMD_SYNC has now completed issuing and we now know if something is hazarding the CMD_SYNC from completing. Fields:

cmd_id unsigned int

The ID of the command being issued.

cons_incl_wrap unsigned int

CONS pointer of CMD_SYNC.

hazarded bool

The CMD_SYNC is currently hazarded from completing.

is_synthetic bool

This is a synthetic sync. The cons_incl_wrap is 0xFFFF'ffff.

rl_cmd_sync_issuing

Realm CMD_SYNC is starting to issue, we are going to start accounting for which things have to be part of the prior set of things that have to complete before the CMD_SYNC can complete. Fields:

cmd_id unsigned int

The ID of the command being issued.

cons_incl_wrap unsigned int

CONS pointer of CMD_SYNC.

is_synthetic bool

This is a synthetic sync. The cons_incl_wrap is 0xFFFF'ffff.

rl_cmd_sync_starting_completion_action_irq

Realm CMD_SYNC starting IRQ completion action. Fields:

cmd_id unsigned int

The ID of the command being issued.

cons_incl_wrap unsigned int

CONS pointer of CMD_SYNC.

is_synthetic bool

This is a synthetic sync. The cons_incl_wrap is 0xFFFF'ffff.

msiaddress unsigned int

The MSI address to use (or 0 if not sending an MSI).

msidata unsigned int

The MSI data to send.

rl_cmd_sync_starting_completion_action_sev

Realm CMD_SYNC starting its completion action of SIG_SEV. Fields:

cmd_id unsigned int

The ID of the command being issued.

cons_incl_wrap unsigned int

CONS pointer of CMD_SYNC.

is_synthetic bool

This is a synthetic sync. The cons_incl_wrap is 0xFFFF'ffff.

rl_cmdq_cmd_sync_error

The CMD_SYNC can fault if an ATC Invalidate timeout occurs. The timeout will be attached to the next CMD_SYNC on the queue on which the invalidate was launched. Fields:

cerror enum

The CERROR to attach.

cmd_queue unsigned int

The top bits indicate the CMDQ id. These bits are attached to all cmd_ids generated for this queue. The bits [63:44] contain the queue ID. [63:62] 0: s-ssd, 1: reserved, 2: ns-ssd. 3: rl-ssd [61] is ECMDQ [59:52] QCP index [51:44] ECMDQ index in QCP.

herror enum

The HERROR to attach, or HERROR_NONE if we are attaching a CERROR.

rl_cmdq_cmd_sync_error_delivered

The CMD_SYNC can fault if an ATC Invalidate timeout occurs. This is the CMD_SYNC delivering that fault. Fields:

cerror enum

The CERROR to attach.

cmd_id unsigned int

Command ID of the CMD_SYNC.

rl_cmdq_ecmdq_enack

Trace the changes to ENACK. Fields:

cerror enum

The error being exposed, if any.

cmd_queue unsigned int

The top bits indicate the CMDQ id. These bits are attached to all cmd_ids generated for this queue. The bits [63:44] contain the queue ID. [63:62] 0: s-ssd, 1: reserved, 2: ns-ssd. 3: rl-ssd [61] is ECMDQ [59:52] QCP index [51:44] ECMDQ index in QCP.

enack bool

The value of ENACK.

forced_error bool

The queue was enabled with ERR != ERRACK and so the implementation chooses to trigger the error reporting mechanism.

rl_cmdq_issue

A command is actually being executed. Fields:

cmd_id unsigned int

Command ID of the command being executed.

cons unsigned int

CONS of the command.

what string

What is being executed.

rl_cmdq_pointers_state

The command queue pointers. Fields:

cmd_queue unsigned int

The top bits indicate the CMDQ id. These bits are attached to all cmd_ids generated for this queue. The bits [63:44] contain the queue ID. [63:62] 0: s-ssd, 1: reserved, 2: ns-ssd. 3: rl-ssd [61] is ECMDQ [59:52] QCP index [51:44] ECMDQ index in QCP.

disable_fetch bool

The model has decided that as you programmed PROD/CONS inconsistently that it would not fetch as it has no confidence as to which are commands and which are not.

q_issuable_and_issued_cons unsigned int

The CONS pointer of the region that is issuable and/or issued (including wrap).

q_issuable_and_issued_prod unsigned int

The PROD pointer of the region that is issuable and/or issued (including wrap).

q_programmer_cons unsigned int

The CONS pointer as seen by the programmer (including wrap).

q_programmer_prod unsigned int

The PROD pointer as seen by the programmer (including wrap).

q_unfetched_cons unsigned int

The CONS pointer of the region unfetched (including wrap).

q_unfetched_prod unsigned int

The PROD pointer of the region unfetched (including wrap).

rl_cmdq_state

A description of what the CMDQ can do now. Fields:

can_cerror bool

There is a CERROR that can be recognised.

can_cmd_sync bool

There is a CMD_SYNC that can be recognised.

can_fetch bool

The SMMU can fetch.

can_herror bool

There is an HERROR that can be recognised.

can_issue bool

There are fetched commands that are eligible to be issued.

can_update_cons bool

The programmer view of CMDQ_CONS can be moved to say some have been consumed.

cmd_queue unsigned int

The top bits indicate the CMDQ id. These bits are attached to all cmd_ids generated for this queue. The bits [63:44] contain the queue ID. [63:62] 0: s-ssd, 1: reserved, 2: ns-ssd. 3: rl-ssd [61] is ECMDQ [59:52] QCP index [51:44] ECMDQ index in QCP.

cmd_sync enum

Why a CMD_SYNC cannot be issued.

cmdqen bool

CMDQEN, when enabled then the CMDQ can fetch commands.

cmdqen_1_to_0_unacknowledged bool

CMDQEN has been set to 0, but we have yet to acknowledge it.

current-real-cerror enum

The current CERROR that would be visible to the programmer if there was an error.

fetch-disabled_due_to_programmer_error bool

The PROD/CONS pointer was inconsistent and the model decided to stop processing commands.

fetch-number_of_unfetched_commands unsigned int

The number of commands that have yet to be fetched from memory, or ~0u if CMDQEN == 0.

fetch-state enum

The state of the fetch state machine.

fetch-too_many_commands_outstanding_so_not_fetching_any_more bool

If there are too many issued commands, or pending to be issued then we won't fetch any more.

issue-number_of_issuable_commands unsigned int

The number of commands that have been fetched but not yet issued.

issue-number_of_unissuable_commands unsigned int

This might include commands behind an illegal command, or behind a CMD_SYNC. Under some circumstances this can include a command representing an external abort.

issue-state enum

The internal issued state machine state.

outstanding_fetches_since_reset unsigned int

The number of outstanding fetches.

synthetic_sync_can_be_completed bool

The synthetic sync that is outstanding can be completed.

waiting_for_synthetic_sync bool

There is a synthetic sync that is outstanding.

rl_eventq_adding_to_cmd_sync_prior_set

The event created by the specified transaction id must become visible to the programmer before the CMD_SYNC can complete. Fields:

trans_id unsigned int

The transaction id that caused the event.

rl_eventq_auto_retry_stalled_transaction

A transaction that was stalled but not yet reported on the Event Queue was retried. The retry can happen if the transaction didn't write because it was de-duplicated behind a reported stalling transaction and that stalling transaction was CMD_RESUMEd, or if the queue was unwriteable and the queue is now writeable. Fields:

trans_id unsigned int

Transaction id of the auto-retried transaction.

why enum

Why the transaction retried.

rl_eventq_cmd_sync_unhazarded

The CMD_SYNC has been unhazarded as the prior set is empty and as much of it as required is visible in the programmers' view. Fields:

prod_incl_wrap unsigned int

The prod index including wrap bit that was required to become visible.

prog_prod_incl_wrap unsigned int

The current programmer-visible prod including wrap bit.

rl_eventq_eventqen_and_eventqenack

Trace the values of EVENTQEN and EVENTQENACK as they change. Fields:

about enum

Some more information about the queue state.

cons_incl_wrap unsigned int

The CONS pointer (including wrap bits).

eventqen bool

EVENTQEN.

eventqenack bool

EVENTQENACK.

prod_incl_wrap unsigned int

The PROD pointer (including wrap bits).

rl_eventq_external_abort

A particular eventq record got an external abort. The event will subsequently appear in the ns/s_eventq_lost_event_records trace. Fields:

prod unsigned int

The prod pointer (excluding wrap bit) of the event record that aborted.

trans_id unsigned int

The transaction id that produced the event record that aborted.

rl_eventq_lost_event_records

Event records that are lost and never record appear on this trace. Fields:

dword0 unsigned int

Double word 0 of event.

dword1 unsigned int

Double word 1 of event.

dword2 unsigned int

Double word 2 of event.

dword3 unsigned int

Double word 3 of event.

event enum

Event code that was thrown away.

prod unsigned int

The PROD index of the event if appropriate or ~0u otherwise.

reason enum

The reason the record was lost.

trans_id unsigned int

The transaction id that this corresponds to.

rl_eventq_overflow_acknowledged

When events are lost then they appear on the eventq_lost_event_records_* trace. This will activate the overflow flag in the corresponding SMMU_s_EVENTQ_PROD register. Software can then acknowledge that flag using the overflow acknowledge flag in SMMU_s_EVENTQ_CONS. Fields:

bad bool

If true then the programmer has attempted to acknowledge an overflow condition that didn't exist.

count_of_events_lost_due_to_overflow unsigned int

This is the number of events that were lost because the event queue was full. This does not include any lost because of an EVENTQ_ABT_ERR.

rl_eventq_pending_event_records

Event records that are pending to be dealt with by the eventq state machine. A specific event can be traced multiple times if it becomes part of the prior set of transactions of a CMD_SYNC that has to complete before the CMD_SYNC is allowed to be completed. Fields:

dword0 unsigned int

Double word 0 of event.

dword1 unsigned int

Double word 1 of event.

dword2 unsigned int

Double word 2 of event.

dword3 unsigned int

Double word 3 of event.

event enum

Event code.

is_prior_set bool

The transaction forms part of the prior set of a CMD_SYNC that have to become visible (or thrown away if necessary) before the CMD_SYNC can complete.

trans_id unsigned int

The transaction id that this corresponds to.

rl_eventq_prod_cons_updated

This records when the programmer visible PROD/CONS pointer is updated. More can be visible in memory than are advertised in the programmer-visible PROD pointer. This also traces the latest PROD that could have been advertised. Fields:

mem_prod unsigned int

The PROD pointer value for all records that are visible in the memory.

num_events_available unsigned int

The number of events available to the programmer.

prog_cons unsigned int

The programmer-visible CONS pointer.

prog_prod unsigned int

The programmer-visible PROD pointer.

what enum

What changed?.

rl_eventq_raising_eventq_abt_err

EVENTQ_ABT_ERR can now be raised.

rl_eventq_removing_from_cmd_sync_prior_set

The event captured by a CMD_SYNC and created by the specified transaction id is now in memory. Fields:

prod unsigned int

The prod index (no wrap bit) that must become visible to the programmer to see this record. If not head_of_line then the record will not become visible and this field is meaningless.

reason enum

The reason the event was removed.

trans_id unsigned int

The transaction id that caused the event.

rl_eventq_stashing_unreported_stalled_transaction_for_auto_retry

A stall event record would have been lost, instead it is stashed away to auto-retry when it can. Fields:

dword0 unsigned int

Double word 0 of event.

dword1 unsigned int

Double word 1 of event.

dword2 unsigned int

Double word 2 of event.

dword3 unsigned int

Double word 3 of event.

event enum

Event code that was thrown away.

reason enum

The reason the record was lost.

trans_id unsigned int

The transaction id that this corresponds to.

rl_eventq_write_event

An event is committed to being written. Fields:

dword0 unsigned int

Double word 0 of event.

dword1 unsigned int

Double word 1 of event.

dword2 unsigned int

Double word 2 of event.

dword3 unsigned int

Double word 3 of event.

event enum

Event code that was thrown away.

prod unsigned int

The PROD pointer (without wrap) that this event will be written to.

stall_event bool

Describes an event for a stalled transaction.

trans_id unsigned int

Transaction ID that generated this event.

rl_eventq_write_event_visible_in_memory

An event is now visible in memory. Fields:

abort bool

The write aborted.

dword0 unsigned int

Double word 0 of event.

dword1 unsigned int

Double word 1 of event.

dword2 unsigned int

Double word 2 of event.

dword3 unsigned int

Double word 3 of event.

event enum

Event code that was thrown away.

prod unsigned int

The PROD pointer (without wrap) that this event will be written to.

stall_event bool

Describes an event for a stalled transaction.

trans_id unsigned int

Transaction ID that generated this event.

rl_gerror_inverted

A GERROR bit was attempted to be inverted to record a fault. Fields:

already_different_to_gerrorn bool

True if the bit was already different to the corresponding bit in GERRORN and so it was not inverted.

gerrorn unsigned int

GERRORN register value.

interrupt_potentially_sent bool

An interrupt is potentially sent, depending on IRQEN.
GERROR(N).MSI_GERROR_ABT_ERR does not send an interrupt.

new_gerror unsigned int

GERROR register value after any inversion.

which enum

Which bit was attempted to be inverted.

rl_gerrorn_acknowledge

A GERROR was acknowledged by SW. Fields:

which enum

Which GERROR was acknowledged.

rl_priq_auto_response

A PRIQ auto response is generated. Fields:

prgindex unsigned int

The PRG Index of the response.

resp enum

The response to send.

streamid unsigned int

The StreamID of the response.

substreamid unsigned int

The PASID/SubstreamID attached to the response, or ~0u if there is none.

trans_id unsigned int

The ID of the PPR that is being lost.

why enum

Why the PPR is being lost.

rl_priq_lost_ppr

The PRIQ is not reporting a PPR into the queue. It may or may not generate an auto-response. Fields:

is_stop_marker bool

The PPR being lost is a Stop Marker.

resp enum

What, if any, auto-response will be.

trans_id unsigned int

The ID of the PPR that is being lost.

why enum

Why the PPR is being lost.

rl_priq_overflow_acking

Indicates that an overflow condition was acknowledged by writing to:- SMMU_PRIQ_CONS.OVACKFLG. Fields:

new_ovackflg bool

The new value of the SMMU_PRIQ_CONS.OVACKFLG.

rl_priq_overflow_asserting

Indicates that we are toggling the SMMU_PRIQ_PROD.OVFLG because we lost a PRI request due to the PRIQ being full and an existing overflow condition does not already exist. Fields:

new_ovflg bool

The new value of the SMMU_PRIQ_PROD.OVFLG.

trans_id unsigned int

The transaction ID of the PPR that caused the overflow.

rl_priq_received

A PRIQ Request has been received and is queued waiting processing. This does not mean that it has been written to the PRIQ but has been placed in a pending queue awaiting a decision about what to do about it. Fields:

L bool

The Last in the Page Request Group (PRG).

R bool

Read permission is requested.

W bool

Read permission is requested.

prgindex unsigned int

The Page Request Group (PRG) index that is being used to identify this request to the EndPoint.

stop_marker bool

If LRW == 0b100 then this indicates this is a stop marker.

streamid unsigned int

The StreamID this request is associated with.

substreamid unsigned int

The SubstreamID (PASID) this request is associated with, of ~0u if no SubstreamID.

substreamid_P bool

If has a SubstreamID (!= ~0u) then Privileged Mode Requested.

substreamid_X bool

If has a SubstreamID (!= ~0u) then Execution Requested (R must also be 1).

trans_id unsigned int

The transaction ID that identifies this request.

untranslated_address unsigned int

The Untranslated Address that this is requesting.

rl_priq_state

The trace of various fields that indicate the state of the PRIQ. Fields:

cons_incl_wrap unsigned int

The value of SMMU_PRIQ.CONS.RD_and_RD_wrap.

number_of_pprs unsigned int

The number of PPRs as indicated by the CONS/PROD.

number_of_pprs_still_to_deal_with unsigned int

This is the number of PPRs that are currently waiting to either be written to the PRIQ, or auto-responded to.

number_of_priq_writes_in_flight unsigned int

The number of writes to the PRIQ that are currently in flight.

ovackflg bool

The OVACKFLG which if different to OVFLG is used to indicate that the PRIQ overflowed.

ovflg bool

The OVFLG which if different to OVACKFLG is used to indicate that the PRIQ overflowed.

priq_abt_err bool

There is an active SMMU_GERROR{N}.PRIQ_ABT_ERR.

priqen bool

The value of SMMU_CRO.PRIQEN. The *effective* value is 0 if SMMUEN == 0.

priqenack bool

The value of SMMU_CROACK.PRIQEN.

prod_incl_wrap unsigned int

The value of SMMU_PRIQ.CONS.WR_and_WR_wrap.

queue_disabled_due_to_prior_programming_error bool

The queue was disabled as the programmer got CONS/PROD into an inconsistent state. The model will disable the PRIQ until SW disables and re-enables the queue via SMMU_CRO.PRIQEN.

smmuen bool

The value of SMMU_CRO.SMMUEN. If this is 0 then the effective PRIQEN is 0.

smmuenack bool

The value of SMMU_CRO.SMMUEN. If this is 0 then the effective PRIQEN is 0.

table_size_in_elements unsigned int

The size of the table in the number of items it can hold.

rl_priq_write_aborted

A PRIQ write aborted. The PRIQ now goes into an error state and will start auto-responding to PRI requests. Fields:

prod_incl_wrap unsigned int

The PROD including the Wrap bit where we were trying to write.

trans_id unsigned int

The transaction ID of the PPR that aborted.

rl_priq_write_ok

A PRIQ write completed OK. Fields:

prod_incl_wrap unsigned int

The PROD including the Wrap bit where we wrote..

trans_id unsigned int

The transaction ID of the PPR.

rl_priq_write_start

A PRIQ request has been received and is going to be attempt to be written to the queue.

Fields:

L bool

The Last in the Page Request Group (PRG).

R bool

Read permission is requested.

W bool

Read permission is requested.

prgindex unsigned int

The Page Request Group (PRG) index that is being used to identify this request to the EndPoint.

prod_incl_wrap unsigned int

PROD position including the wrap bit.

stop_marker bool

If LRW == 0b100 then this indicates this is a stop marker.

streamid unsigned int

The StreamID this request is associated with.

substreamid unsigned int

The SubstreamID (PASID) this request is associated with, of ~0u if no SubstreamID.

substreamid_P bool

If has a SubstreamID (!= ~0u) then Privileged Mode Requested.

substreamid_X bool

If has a SubstreamID (!= ~0u) then Execution Requested (R must also be 1).

trans_id unsigned int

The transaction ID that identifies this request.

untranslated_address unsigned int

The Untranslated Address that this is requesting.

s_cmd_sync_completed_irq

“Secure” CMD_SYNC completed its IRQ action. Fields:

cmd_id unsigned int

The ID of the command being issued.

cons_incl_wrap unsigned int

CONS pointer of CMD_SYNC.

is_synthetic bool

This is a synthetic sync. The cons_incl_wrap is 0xFFFF'ffff.

s_cmd_sync_completed_no_action

Secure CMD_SYNC completed, there was no CMD_SYNC action (SIG_NONE). Fields:

cmd_id unsigned int

The ID of the command being issued.

cons_incl_wrap unsigned int

CONS pointer of CMD_SYNC.

is_synthetic bool

This is a synthetic sync. The cons_incl_wrap is 0xFFFF'ffff.

s_cmd_sync_completed_sev

Secure CMD_SYNC completed its action of SIG_SEV. Fields:

cmd_id unsigned int

The ID of the command being issued.

cons_incl_wrap unsigned int

CONS pointer of CMD_SYNC.

is_synthetic bool

This is a synthetic sync. The cons_incl_wrap is 0xFFFF'ffff.

s_cmd_sync_completed_with_error

Secure CMD_SYNC completed, however, there was an error associated with the completion of the CMD_SYNC. Fields:

error enum

The error associated with the CMD_SYNC.

cmd_id unsigned int

The ID of the command being issued.

cons_incl_wrap unsigned int

CONS pointer of CMD_SYNC.

is_synthetic bool

This is a synthetic sync. The cons_incl_wrap is 0xFFFF'ffff.

s_cmd_sync_issued

Secure CMD_SYNC has now completed issuing and we now know if something is hazarding the CMD_SYNC from completing. Fields:

cmd_id unsigned int

The ID of the command being issued.

cons_incl_wrap unsigned int

CONS pointer of CMD_SYNC.

hazarded bool

The CMD_SYNC is currently hazarded from completing.

is_synthetic bool

This is a synthetic sync. The cons_incl_wrap is 0xFFFF'ffff.

s_cmd_sync_issuing

Secure CMD_SYNC is starting to issue, we are going to start accounting for which things have to be part of the prior set of things that have to complete before the CMD_SYNC can complete. Fields:

cmd_id unsigned int

The ID of the command being issued.

cons_incl_wrap unsigned int

CONS pointer of CMD_SYNC.

is_synthetic bool

This is a synthetic sync. The cons_incl_wrap is 0xFFFF'ffff.

s_cmd_sync_starting_completion_action_irq

Secure CMD_SYNC starting IRQ completion action. Fields:

cmd_id unsigned int

The ID of the command being issued.

cons_incl_wrap unsigned int

CONS pointer of CMD_SYNC.

is_synthetic bool

This is a synthetic sync. The cons_incl_wrap is 0xFFFF'ffff.

msiaddress unsigned int

The MSI address to use (or 0 if not sending an MSI).

msidata unsigned int

The MSI data to send.

s_cmd_sync_starting_completion_action_sev

Secure CMD_SYNC starting its completion action of SIG_SEV. Fields:

cmd_id unsigned int

The ID of the command being issued.

cons_incl_wrap unsigned int

CONS pointer of CMD_SYNC.

is_synthetic bool

This is a synthetic sync. The cons_incl_wrap is 0xFFFF'ffff.

s_cmdq_cmd_sync_error

The CMD_SYNC can fault if an ATC Invalidate timeout occurs. The timeout will be attached to the next CMD_SYNC on the queue on which the invalidate was launched. Fields:

cerror enum

The CERROR to attach.

cmd_queue unsigned int

The top bits indicate the CMDQ id. These bits are attached to all cmd_ids generated for this queue. The bits [63:44] contain the queue ID. [63:62] 0: s-ssd, 1: reserved, 2: ns-ssd. 3: rl-ssd [61] is ECMDQ [59:52] QCP index [51:44] ECMDQ index in QCP.

herror enum

The HERROR to attach, or HERROR_NONE if we are attaching a CERROR.

s_cmdq_cmd_sync_error_delivered

The CMD_SYNC can fault if an ATC Invalidate timeout occurs. This is the CMD_SYNC delivering that fault. Fields:

cerror enum

The CERROR to attach.

cmd_id unsigned int

Command ID of the CMD_SYNC.

s_cmdq_ecmdq_enack

Trace the changes to ENACK. Fields:

cerror enum

The error being exposed, if any.

cmd_queue unsigned int

The top bits indicate the CMDQ id. These bits are attached to all cmd_ids generated for this queue. The bits [63:44] contain the queue ID. [63:62] 0: s-ssd, 1: reserved, 2: ns-ssd. 3: rl-ssd [61] is ECMDQ [59:52] QCP index [51:44] ECMDQ index in QCP.

enack bool

The value of ENACK.

forced_error bool

The queue was enabled with ERR != ERRACK and so the implementation chooses to trigger the error reporting mechanism.

s_cmdq_issue

A command is actually being executed. Fields:

cmd_id unsigned int

Command ID of the command being executed.

cons unsigned int

CONS of the command.

what string

What is being executed.

s_cmdq_pointers_state

The command queue pointers. Fields:

cmd_queue unsigned int

The top bits indicate the CMDQ id. These bits are attached to all cmd_ids generated for this queue. The bits [63:44] contain the queue ID. [63:62] 0: s-ssd, 1: reserved, 2: ns-ssd. 3: rl-ssd [61] is ECMDQ [59:52] QCP index [51:44] ECMDQ index in QCP.

disable_fetch bool

The model has decided that as you programmed PROD/CONS inconsistently that it would not fetch as it has no confidence as to which are commands and which are not.

q_issuable_and_issued_cons unsigned int

The CONS pointer of the region that is issuable and/or issued (including wrap).

q_issuable_and_issued_prod unsigned int

The PROD pointer of the region that is issuable and/or issued (including wrap).

q_programmer_cons unsigned int

The CONS pointer as seen by the programmer (including wrap).

q_programmer_prod unsigned int

The PROD pointer as seen by the programmer (including wrap).

q_unfetched_cons unsigned int

The CONS pointer of the region unfetched (including wrap).

q_unfetched_prod unsigned int

The PROD pointer of the region unfetched (including wrap).

s_cmdq_state

A description of what the CMDQ can do now. Fields:

can_cerror bool

There is a CERROR that can be recognised.

can_cmd_sync bool

There is a CMD_SYNC that can be recognised.

can_fetch bool

The SMMU can fetch.

can_herror bool

There is an HERROR that can be recognised.

can_issue bool

There are fetched commands that are eligible to be issued.

can_update_cons bool

The programmer view of CMDQ_CONS can be moved to say some have been consumed.

cmd_queue unsigned int

The top bits indicate the CMDQ id. These bits are attached to all cmd_ids generated for this queue. The bits [63:44] contain the queue ID. [63:62] 0: s-ssd, 1: reserved, 2: ns-ssd. 3: rl-ssd [61] is ECMDQ [59:52] QCP index [51:44] ECMDQ index in QCP.

cmd_sync enum

Why a CMD_SYNC cannot be issued.

cmdqen bool

CMDQEN, when enabled then the CMDQ can fetch commands.

cmdqen_1_to_0_unacknowledged bool

CMDQEN has been set to 0, but we have yet to acknowledge it.

current-real-cerror enum

The current CERROR that would be visible to the programmer if there was an error.

fetch-disabled_due_to_programmer_error bool

The PROD/CONS pointer was inconsistent and the model decided to stop processing commands.

fetch-number_of_unfetched_commands unsigned int

The number of commands that have yet to be fetched from memory, or ~0u if CMDQEN == 0.

fetch-state enum

The state of the fetch state machine.

fetch-too_many_commands_outstanding_so_not_fetching_any_more bool

If there are too many issued commands, or pending to be issued then we won't fetch any more.

issue-number_of_issuable_commands unsigned int

The number of commands that have been fetched but not yet issued.

issue-number_of_unissuable_commands unsigned int

This might include commands behind an illegal command, or behind a CMD_SYNC. Under some circumstances this can include a command representing an external abort.

issue-state enum

The internal issued state machine state.

outstanding_fetches_since_reset unsigned int

The number of outstanding fetches.

synthetic_sync_can_be_completed bool

The synthetic sync that is outstanding can be completed.

waiting_for_synthetic_sync bool

There is a synthetic sync that is outstanding.

s_eventq_adding_to_cmd_sync_prior_set

The event created by the specified transaction id must become visible to the programmer before the CMD_SYNC can complete. Fields:

trans_id unsigned int

The transaction id that caused the event.

s_eventq_auto_retry_stalled_transaction

A transaction that was stalled but not yet reported on the Event Queue was retried. The retry can happen if the transaction didn't write because it was de-duplicated behind a reported stalling transaction and that stalling transaction was CMD_RESUMEd, or if the queue was unwriteable and the queue is now writeable. Fields:

trans_id unsigned int

Transaction id of the auto-retried transaction.

why enum

Why the transaction retried.

s_eventq_cmd_sync_unhazardded

The CMD_SYNC has been unhazardded as the prior set is empty and as much of it as required is visible in the programmers' view. Fields:

prod_incl_wrap unsigned int

The prod index including wrap bit that was required to become visible.

prog_prod_incl_wrap unsigned int

The current programmer-visible prod including wrap bit.

s_eventq_eventqen_and_eventqenack

Trace the values of EVENTQEN and EVENTQENACK as they change. Fields:

about enum

Some more information about the queue state.

cons_incl_wrap unsigned int

The CONS pointer (including wrap bits).

eventqen bool

EVENTQEN.

eventqenack bool

EVENTQENACK.

prod_incl_wrap unsigned int

The PROD pointer (including wrap bits).

s_eventq_external_abort

A particular eventq record got an external abort. The event will subsequently appear in the ns/s_eventq_lost_event_records trace. Fields:

prod unsigned int

The prod pointer (excluding wrap bit) of the event record that aborted.

trans_id unsigned int

The transaction id that produced the event record that aborted.

s_eventq_lost_event_records

Event records that are lost and never record appear on this trace. Fields:

dword0 unsigned int

Double word 0 of event.

dword1 unsigned int

Double word 1 of event.

dword2 unsigned int

Double word 2 of event.

dword3 unsigned int

Double word 3 of event.

event enum

Event code that was thrown away.

prod unsigned int

The PROD index of the event if appropriate or ~0u otherwise.

reason enum

The reason the record was lost.

trans_id unsigned int

The transaction id that this corresponds to.

s_eventq_overflow_acknowledged

When events are lost then they appear on the eventq_lost_event_records_* trace. This will activate the overflow flag in the corresponding SMMU_s_EVENTQ_PROD register. Software can then acknowledge that flag using the overflow acknowledge flag in SMMU_s_EVENTQ_CONS. Fields:

bad bool

If true then the programmer has attempted to acknowledge an overflow condition that didn't exist.

count_of_events_lost_due_to_overflow unsigned int

This is the number of events that were lost because the event queue was full. This does not include any lost because of an EVENTQ_ABT_ERR.

s_eventq_pending_event_records

Event records that are pending to be dealt with by the eventq state machine. A specific event can be traced multiple times if it becomes part of the prior set of transactions of a CMD_SYNC that has to complete before the CMD_SYNC is allowed to be completed. Fields:

dword0 unsigned int

Double word 0 of event.

dword1 unsigned int

Double word 1 of event.

dword2 unsigned int

Double word 2 of event.

dword3 unsigned int

Double word 3 of event.

event enum

Event code.

is_prior_set bool

The transaction forms part of the prior set of a CMD_SYNC that have to become visible (or thrown away if necessary) before the CMD_SYNC can complete.

trans_id unsigned int

The transaction id that this corresponds to.

s_eventq_prod_cons_updated

This records when the programmer visible PROD/CONS pointer is updated. More can be visible in memory than are advertised in the programmer-visible PROD pointer. This also traces the latest PROD that could have been advertised. Fields:

mem_prod unsigned int

The PROD pointer value for all records that are visible in the memory.

num_events_available unsigned int

The number of events available to the programmer.

prog_cons unsigned int

The programmer-visible CONS pointer.

prog_prod unsigned int

The programmer-visible PROD pointer.

what enum

What changed?.

s_eventq_raising_eventq_abt_err

EVENTQ_ABT_ERR can now be raised.

s_eventq_removing_from_cmd_sync_prior_set

The event captured by a CMD_SYNC and created by the specified transaction id is now in memory. Fields:

prod unsigned int

The prod index (no wrap bit) that must become visible to the programmer to see this record. If not head_of_line then the record will not become visible and this field is meaningless.

reason enum

The reason the event was removed.

trans_id unsigned int

The transaction id that caused the event.

s_eventq_stashing_unreported_stalled_transaction_for_auto_retry

A stall event record would have been lost, instead it is stashed away to auto-retry when it can. Fields:

dword0 unsigned int

Double word 0 of event.

dword1 unsigned int

Double word 1 of event.

dword2 unsigned int

Double word 2 of event.

dword3 unsigned int

Double word 3 of event.

event enum

Event code that was thrown away.

reason enum

The reason the record was lost.

trans_id unsigned int

The transaction id that this corresponds to.

s_eventq_write_event

An event is committed to being written. Fields:

dword0 unsigned int

Double word 0 of event.

dword1 unsigned int

Double word 1 of event.

dword2 unsigned int

Double word 2 of event.

dword3 unsigned int

Double word 3 of event.

event enum

Event code that was thrown away.

prod unsigned int

The PROD pointer (without wrap) that this event will be written to.

stall_event bool

Describes an event for a stalled transaction.

trans_id unsigned int

Transaction ID that generated this event.

s_eventq_write_event_visible_in_memory

An event is now visible in memory. Fields:

abort bool

The write aborted.

dword0 unsigned int

Double word 0 of event.

dword1 unsigned int

Double word 1 of event.

dword2 unsigned int

Double word 2 of event.

dword3 unsigned int

Double word 3 of event.

event enum

Event code that was thrown away.

prod unsigned int

The PROD pointer (without wrap) that this event will be written to.

stall_event bool

Describes an event for a stalled transaction.

trans_id unsigned int

Transaction ID that generated this event.

s_gbpa_abort_init

The pin is driven. This is the reset value of SMMU_S_GBPA.ABORT. This pin is sampled on the negedge of the reset pin and has to be driven beforehand for the value to take effect. Fields:

value bool

The value of the signal.

s_gerror_inverted

A GERROR bit was attempted to be inverted to record a fault. Fields:

already_different_to_gerrorn bool

True if the bit was already different to the corresponding bit in GERRORN and so it was not inverted.

gerrorn unsigned int

GERRORN register value.

interrupt_potentially_sent bool

An interrupt is potentially sent, depending on IRQEN.
GERROR(N).MSI_GERROR_ABT_ERR does not send an interrupt.

new_gerror unsigned int

GERROR register value after any inversion.

which enum

Which bit was attempted to be inverted.

s_gerrorn_acknowledge

A GERROR was acknowledged by SW. Fields:

which enum

Which GERROR was acknowledged.

s_priq_auto_response

A PRIQ auto response is generated. Fields:

prgindex unsigned int

The PRG Index of the response.

resp enum

The response to send.

streamid unsigned int

The StreamID of the response.

substreamid unsigned int

The PASID/SubstreamID attached to the response, or ~0u if there is none.

trans_id unsigned int

The ID of the PPR that is being lost.

why enum

Why the PPR is being lost.

s_priq_lost_ppr

The PRIQ is not reporting a PPR into the queue. It may or may not generate an auto-response. Fields:

is_stop_marker bool

The PPR being lost is a Stop Marker.

resp enum

What, if any, auto-response will be.

trans_id unsigned int

The ID of the PPR that is being lost.

why enum

Why the PPR is being lost.

s_priq_received

A PRIQ Request has been received and is queued waiting processing. This does not mean that it has been written to the PRIQ but has been placed in a pending queue awaiting a decision about what to do about it. Fields:

L bool

The Last in the Page Request Group (PRG).

R bool

Read permission is requested.

W bool

Read permission is requested.

prgindex unsigned int

The Page Request Group (PRG) index that is being used to identify this request to the EndPoint.

stop_marker bool

If LRW == 0b100 then this indicates this is a stop marker.

streamid unsigned int

The StreamID this request is associated with.

substreamid unsigned int

The SubstreamID (PASID) this request is associated with, of ~0u if no SubstreamID.

substreamid_P bool

If has a SubstreamID (!= ~0u) then Privileged Mode Requested.

substreamid_X bool

If has a SubstreamID (!= ~0u) then Execution Requested (R must also be 1).

trans_id unsigned int

The transaction ID that identifies this request.

untranslated_address unsigned int

The Untranslated Address that this is requesting.

sec_override

Controls non-secure accesses to some registers. Fields:

value bool

The value of the signal.

sev

Send a SEV. Fields:

why enum

Why the SEV was generated.

smmu_atc_inv

The CMD_ATC_INV command is sent. Fields:

address unsigned int

Untranslated Address to invalidate.

cmd_id unsigned int

Command id.

cons unsigned int

CONS of the command.

global bool

Global flag.

log2_size_in_bytes unsigned int

Log 2 of the size in bytes of the region to invalidate.

ns_number_in_flight unsigned int

Number of ATC invalidates in flight issued from the Non-secure CMDQ.

ns_number_waiting_to_go_out unsigned int

The total number of ATC invalidates waiting to be pushed out from the Non-secure CMDQ.

rl_number_in_flight unsigned int

Number of ATC invalidates in flight issued from the Realm CMDQ.

rl_number_waiting_to_go_out unsigned int

The total number of ATC invalidates waiting to be pushed out from the Realm CMDQ.

s_number_in_flight unsigned int

Number of ATC invalidates in flight issued from the Secure CMDQ.

s_number_waiting_to_go_out unsigned int

The total number of ATC invalidates waiting to be pushed out from the Secure CMDQ.

ssd_of_queue enum

The SSD of the CMDQ.

ssd_of_streamid enum

The SSD of the StreamID.

streamid unsigned int

StreamID in the invalidate command.

substreamid unsigned int

SubstreamID or ~0u if SSValid == false.

smmu_atc_inv_completed

The CMD_ATC_INV command completed. Fields:

cmd_id unsigned int

Command id, if top-bit is set then was issued from the Non-secure CMDQ.

ns_number_in_flight unsigned int

Number of ATC invalidates in flight now this one has completed, that were issued by the Non-secure CMDQ.

ns_number_waiting_to_go_out unsigned int

The total number of ATC invalidates waiting to be pushed out from the Non-secure CMDQ.

response enum

The response.

rl_number_in_flight unsigned int

Number of ATC invalidates in flight now this one has completed, that were issued by the Realm CMDQ.

rl_number_waiting_to_go_out unsigned int

The total number of ATC invalidates waiting to be pushed out from the Realm CMDQ.

s_number_in_flight unsigned int

Number of ATC invalidates in flight now this one has completed, that were issued by the Secure CMDQ.

s_number_waiting_to_go_out unsigned int

The total number of ATC invalidates waiting to be pushed out from the Secure CMDQ.

ssd_of_cmdq enum

The SSD of the CMDQ.

ssd_of_streamid enum

The SSD of the StreamID.

smmu_atc_inv_end

An ATC invalidate request has been started. Fields:

PASID_global bool

If has a SubstreamID (PASID) then if it is 'global' or not.

address unsigned int

The Untranslated Address to use.

log2_size_in_bytes unsigned int

Log2 of the size of the region in bytes to invalidate.

response enum

The response to the ATC invalidate.

ssd enum

The SSD of the stream.

streamid unsigned int

The StreamID to use.

substreamid unsigned int

The SubstreamID to use (or ~0u if not used). This corresponds to the PASID.

smmu_atc_inv_start

An ATC invalidate request has been started. Fields:

PASID_global bool

If has a SubstreamID (PASID) then if it is 'global' or not.

address unsigned int

The Untranslated Address to use.

log2_size_in_bytes unsigned int

Log2 of the size of the region in bytes to invalidate.

ssd enum

The SSD of the stream.

streamid unsigned int

The StreamID to use.

substreamid unsigned int

The SubstreamID to use (or ~0u if not used). This corresponds to the PASID.

smmu_ats_initial

This is the initial ATS request. Fields:

XT bool

The XT bit.

ia unsigned int

Input address.

max_number_of_replies unsigned int

The maximum number of replies allowed to return.

no_write bool

The NW (no write flag) of the ATS request. If clear then the requester is going to do a write.

pasid_execute_requested bool

If there is a PASID (substreamid != ~0u) then this represents the requester asking for execution permissions.

pasid_privileged_mode_requested bool

If there is a PASID (substreamid != ~0u) then this represents the requester asking for the privileged mode's permissions.

protected_mode bool

Is the ATS Request protected-mode?.

source_cxl bool

Does the ATS request have Source.CXL set?.

ssd enum

The SSD of the ATS request.

ssd_ns bool

Incoming SSD is non-secure.

streamid unsigned int

StreamID of the ATS request.

substreamid unsigned int

SubstreamID (which is identical to the PASID) of the ATS request. If no PASID-prefix is sent then this is ~0u.

tbu unsigned int

Translation Buffer Unit number.

smmu_ats_reply_failure

This is an ATS reply indicating failure. Fields:

event enum

Equivalent event number that would have been generated for an equivalent ordinary transaction.

failure enum

What is the failure response code?.

state enum

The transaction state of the successfully ATS request.

smmu_ats_reply_success

This is an ATS reply, typically the SMMU will only return a single response, even if the requester indicated it could accept more replies. NOTE that the SMMU responds with 'success' in some cases when a fault is encountered and RW==0. Fields:

N bool

Non-snooped access. If one then the requester must clear the NoSnoop bit on transactions, unless otherwise enabled in a Function-specific manner.

P bool

Privileged mode. These permissions related to privileged mode.

RWX enum

Read/Write/Execute.

U bool

Untranslated access. If one, and RW !=0 then use UntranslatedAccesses for the allowed accesses by RW(X).

cxl_io bool

The CXL.io response.

inner enum

The inner cacheability attributes to use for TranslatedAccesses.

input_address unsigned int

Input address of the ATS request.

instcfg enum

The STE.INSTCFG field.

outer enum

The outer cacheability attributes to use for TranslatedAccesses.

pas enum

The PAS this mapping corresponds to. This holds the same information as the TE bit for realm streams.

privcfg enum

The STE.PRIVCFG field.

shareability unsigned int

The shareability to use for TranslatedAccesses.

size unsigned int

The size of the region covered by this translation.

state enum

The transaction state of the successfully ATS request.

translated_address unsigned int

If $RW \neq 0$ && $U \neq 0$, then the Translated Address that a TranslatedAccess can be made with.

smmu_axi_stream_msi

An SMMU generated MSI is directly sent through the axi_stream_msi_m port, typically connected to the GIC port axi_stream_msi_s. Fields:

TDEST unsigned int

Routing information for the data stream, typically identifying the GIC.

TID unsigned int

Data stream identifier for the SMMU.

axi_stream_msi_addr_to_match unsigned int

Current address to match for SMMU-originated MSIs to send out of the axi_stream_msi_m port.

data unsigned int

The MSI sent.

smmu_final_transaction

This is the transaction group request to remap has completed one set of remapping. For ordinary transactions, this represents a bundle of transactions with the same attributes but different addresses within a certain range around the address. If it stalls then it will report through this trace source, stall (stag_if_stalling != ~0u) and when resume will issue another smmu_initial_transaction as it undergoes remapping again. This trace source can also represent part of the process for ATOS/ATS or finding STE.PPAR for PRI requests that need to be auto-responded to. Fields:

axmmuflow enum

The AxMMUFLOW for this transaction group. stallable – allow the transaction to be stalled if configured. translated-access – is a PCIe Translated Access non-stallable – do not stall transaction even if configured. transfault-flow – do not report for Translation Related Faults, but tell client device to use a page request mechanism instead (for example PRI).

begin_input_address_range unsigned int

The start of the input address range that is size_of_region_in_bytes.

begin_ipa_range unsigned int

The start of the IPA range that is of size_of_region_in_bytes.

begin_output_address_range unsigned int

The start of the output address range that is of size_of_region_in_bytes.

cmo_point enum

The point associated with the CMO, if applicable.

input_address unsigned int

The input address of the transaction group.

input_inner enum

Inner cacheability for the input attributes.

input_outer enum

Outer cacheability for the input attributes.

input_prot enum

The instruction/data, bus-ns/s and privileged/user marking.

input_sh enum

Shareability for the input attributes.

ipa_address unsigned int

The IPA of the transaction.

kind enum

The kind of access this transaction group that this represents.

mecid unsigned int

The MECID of the transaction.

mpam_pmg_and_partid unsigned int

Bits[23:16] are the MPAM PMG. Bits[15:0] are the physical PARTID.

mpam_sp enum

The MPAM_SP of the transaction.

operation enum

The kind of operation that this represents.

output_address unsigned int

The input address of the transaction group.

output_inner enum

Inner cacheability for the output attributes.

output_outer enum

Outer cacheability for the output attributes.

output_prot enum

The instruction/data, bus-ns/s and privileged/user marking.

output_sh enum

Shareability for the output attributes.

output_vmid unsigned int

The output VMID/GBPA.IMPDEF or ~0u if not valid.

size_of_region_in_bytes unsigned int

An imp def size of region for which this translation is valid for.

ssd enum

The SSD of the transaction.

ssd_ns enum

The security state of the transaction.

stag_if_stalling unsigned int

This is the STAG used by the transaction if it is going to stall. It is ~0u if it is not going to stall.

state enum

The final transaction state.

streamid unsigned int

The StreamID of the transaction. ~0ull if NoStreamID.

substreamid unsigned int

The SubstreamID of the transaction or 0xFFFFffff if no SubstreamID.

tbu unsigned int

Translation Buffer Unit number.

trans_id unsigned int

Transaction id. Top bit set if debug transaction.

smmu_gpt_read

A GPT read has completed. Fields:

unsigned int

.

abort enum

Non-zero if the access aborted/failed.

address_checking unsigned int

The address that we are checking.

address_gptdesc unsigned int

The address of the GPT descriptor.

adomain enum

The shareability of the access.

data unsigned int

The data fetched if it didn't abort.

inner_cache enum

The architectural attributes used for the access.

level unsigned int

The level of the GPT walk.

mpam_pmg_and_partid unsigned int

Bits[23:16] are the MPAM PMG. Bits[15:0] are the physical PARTID.

mpam_sp enum

The MPAM_SP of this GPT access.

outer_cache enum

The architectural attributes used for the access.

pas_checking enum

The PAS of the address that we are checking.

post_access_check bool

Is this GPT walk after the access it is checking.

table_base unsigned int

The base address of the table.

trans_id unsigned int

The trans_id of the transaction we are checking if appropriate.

what_checking enum

What kind of access are we checking.

smmu_gpt_tlb_invalidate

A GPT TLB invalidate has been initiated. Fields:

address unsigned int

The address as it appears in the operation.

encoded_size unsigned int

The size as it is encoded in the operation.

kind enum

The kind of operation this is.

pgs_in_bytes unsigned int

The PGS size in bytes.

size_in_bytes unsigned int

For range operations, the size as it appears in the operation.

source enum

Where the TLBI came from.

state enum

Is the operation well formed.

trans_id unsigned int

The transaction id of this invalidate.

smmu_gpt_tlb_invalidate_complete

The GPT TLB invalidate completed. Fields:

source enum

Where the TLBI came from.

trans_id unsigned int

The transaction id of this invalidate.

smmu_initial_transaction

This is the transaction group request to remap is going to start one set of remapping. For ordinary transactions, this represents a bundle of transactions with the same attributes but different addresses within a certain range around the address. A stalling transaction will report through this trace source when it unstalls. This trace source can also represent part of the process for ATOS/ATS or finding STE.PPAR for PRI requests that need to be auto-responded to. Fields:

XT_and_output_pas_checking enum

The XT bit for PCIe Transactions. This specifies the requested check on the output PAS that the device asked for.

axmmuflow enum

The AxMMUFLOW for this transaction group. stallable – allow the transaction to be stalled if configured. translated-access – is a PCIe Translated Access non-stallable – do not stall transaction even if configured. transfault-flow – do not report for Translation Related Faults, but tell client device to use a page request mechanism instead (for example PRI).

cmo_point enum

The point associated with the CMO, if applicable.

dcmdq_qcp_and_index unsigned int

If this is a DCMDQ fetch being translated then this field indicates the DCMDQ QCP index in bits [23:8] and the index in the page in bits [7:0]. Otherwise this is 0xFFFFFFFF.

input_address unsigned int

The input address of the transaction group.

input_inner enum

Inner cacheability for the input attributes.

input_outer enum

Outer cacheability for the input attributes.

input_prot enum

The instruction/data, bus-ns/s and privileged/user marking.

input_sh enum

Shareability for the input attributes.

kind enum

The kind of access this transaction group that this represents.

mecid unsigned int

The incoming MECID for NoStreamID transactions. ~0u for all other transactions.

mpam_partid unsigned int

The MPAM_PARTID for NoStreamID transactions. ~0u for all other transactions.

mpam_pmg unsigned int

The MPAM_PMG for NoStreamID transactions. ~0u for all other transactions.

mpam_sp enum

The MPAM_SP for NoStreamID transactions. ~0u for all other transactions.

operation enum

The kind of operation that this represents.

protected_mode enum

The PM bit.

ssd enum

The SSD of the transaction.

ssd_ns enum

The security state of the transaction.

streamid unsigned int

The StreamID of the transaction. ~0ull means NoStreamID.

substreamid unsigned int

The SubstreamID of the transaction or 0xFFFFffff if no SubstreamID.

tbu unsigned int

Translation Buffer Unit number.

trans_id unsigned int

Transaction id. Top bit set if debug transaction.

smmu_normalize_input_transaction

If the input transaction is normalized before being processed then this traceevent will fire. This is usually dependent on parameters of the implementation. Fields:

incoming_inner enum

The inner cacheability attributes.

incoming_is_instruction bool

The incoming transaction is marked as 'instruction'.

incoming_is_privileged bool

The incoming transaction is marked as 'privileged'.

incoming_outer enum

The outer cacheability attributes.

incoming_pas enum

The PAS of the incoming transaction.

incoming_shareability unsigned int

The incoming shareability.

normalized_inner enum

The normalized inner cacheability attributes.

normalized_is_instruction bool

The incoming transaction is marked as 'instruction'.

normalized_is_privileged bool

The incoming transaction is marked as 'privileged'.

normalized_outer enum

The normalized outer cacheability attributes.

normalized_pas enum

The PAS this mapping corresponds to.

normalized_shareability unsigned int

The normalized shareability.

ssd enum

The SSD of the transaction.

streamid unsigned int

The StreamID of the transaction. ~0ull means NoStreamID.

substreamid unsigned int

The SubstreamID of the transaction or 0xffffffff if no SubstreamID.

trans_id unsigned int

ID of the original transaction.

smmu_pmusnapshot_ack

Acknowledge the pmusnapshot_req to indicate the snapshot has occurred. This is part of a four-phase handshake. Fields:

pin_index unsigned int

The pin index of the signal, or ~0u if not relevant.

pmcg_index enum

The PMCG index.

value bool

The value of the signal.

smmu_pmusnapshot_ack_overridden

The value of pmusnapshot_ack was overridden (likely due to being in integration mode). Fields:

pin_index unsigned int

The pin index of the signal, or ~0u if not relevant.

pmcg_index enum

The PMCG index.

value bool

The value of the signal.

smmu_pmusnapshot_ack_override_end

The overriding of the pmusnapshot_ack signal has ended. Fields:

pin_index unsigned int

The pin index of the signal, or ~0u if not relevant.

pmcg_index enum

The PMCG index.

smmu_pmusnapshot_req

Take a snap shot of the PMU values as though SMMU_PMCG_CAPR.CAPTURE had been written. This is part of a four-phase handshake. Fields:

pin_index unsigned int

The pin index of the signal, or ~0u if not relevant.

pmcg_index enum

The PMCG index.

value bool

The value of the signal.

smmu_poison_tw_data

Poison data has been returned to a table walk transaction. Fields:

bitmap_of_poison unsigned int

The bitmap of which beats of the transaction where poisoned.

is_cas bool

True if this is a compare-and-swap operation.

number_of_64bit_beats unsigned int

Number of beats this transaction fetched.

paddress unsigned int

Physical address of the table walk transaction.

pas enum

The PAS of the bus transaction.

ras_group_id unsigned int

If non-~0u then is the RAS group that we wish to attach the error to. Otherwise will use the IMP DEF policy.

ras_record_index_in_group unsigned int

If non-~0u then is the RAS record index in the group that we wish to attach the error to. Otherwise will use the IMP DEF policy.

what enum

What table walk was being performed.

smmu_pri_resp

The CMD_PRI_RESP command is queued to be sent to the PCIe system. Fields:

auto_response_trans_id unsigned int

trans_id of PRI request we are auto-responding to, or ~0ull if not valid.

cmd_id unsigned int

Command id, or ~0ull if not valid.

cons unsigned int

CONS of the command. ~0u if an auto-response.

prgindex unsigned int

PRG Index.

response enum

If has a SubstreamID (PASID) then if it is 'global' or not.

ssd enum

The SSD of the StreamID this corresponds to.

streamid unsigned int

StreamID in the invalidate command.

substreamid unsigned int

SubstreamID or ~0u if SSValid == false.

smmu_pri_resp_nop

The CMD_PRI_RESP command was NOPped. Fields:

cmd_id unsigned int

Command id.

cons unsigned int

CONS of the command.

prgindex unsigned int

PRG Index.

response enum

If has a SubstreamID (PASID) then if it is 'global' or not.

ssd enum

The SSD of the StreamID this corresponds to.

streamid unsigned int

StreamID in the invalidate command.

substreamid unsigned int

SubstreamID or ~0u if SSValid == false.

why enum

Why the CMD_PRI_RESP was NOPped.

smmu_priq_resp_fake_return

A PRIQ Response is posted to the PCIe subsystem and so has no acknowledgement that it is received. However, in the model then we artificially know when the the PRIQ Response has been delivered to the PCIe subsystem, even if the ATC has not yet acted on it. Fields:

prgindex unsigned int

The PRG Index this request corresponds to.

ssd enum

SSD of response.

streamid unsigned int

The StreamID to use.

substreamid unsigned int

The SubstreamID to use (or ~0u if not used). This corresponds to the PASID.

what enum

What happened.

smmu_priq_resp_start

A PRIQ Response has been posted to the PCIe subsystem. As the response is posted then there is no way of knowing when it is received by the EndPoint. Fields:

prgindex unsigned int

The PRG Index this request corresponds to.

response enum

If has a SubstreamID (PASID) then if it is 'global' or not.

ssd enum

SSD of response.

streamid unsigned int

The StreamID to use.

substreamid unsigned int

The SubstreamID to use (or ~0u if not used). This corresponds to the PASID.

smmu_start_gpt_read

A GPT read is about to be issued. Fields:

address_checking unsigned int

The address that we are checking.

address_gptdesc unsigned int

The address of the GPT descriptor.

adomain enum

The shareability of the access.

inner_cache enum

The architectural attributes used for the access.

level unsigned int

The level of the GPT walk.

mpam_pmg_and_partid unsigned int

Bits[23:16] are the MPAM PMG. Bits[15:0] are the physical PARTID.

mpam_sp enum

The MPAM_SP of this GPT access.

outer_cache enum

The architectural attributes used for the access.

pas_checking enum

The PAS of the address that we are checking.

post_access_check bool

Is this GPT walk after the access it is checking.

read_size unsigned int

The size of the GPT read in bytes.

table_base unsigned int

The base address of the table.

trans_id unsigned int

The trans_id of the transaction we are checking if appropriate.

what_checking enum

What kind of access are we checking.

smmu_thread_wait_wake

Traces a thread's wait/wake status. Fields:

current_ticks unsigned int

The current tick count of simulated time.

event enum

What is happening to this thread.

thread_index unsigned int

The ID of this thread.

ticks unsigned int

If the event relates to a time then this is held in this field. Otherwise, 0.

stall_transaction

A transaction is about to stall. Fields:

stag unsigned int

STAG.

streamid unsigned int

StreamID.

streamid_ns bool

The StreamID was for the non-secure world.

trans_id unsigned int

Transaction id.

stall_transaction_inhibited_by_STALL_MAX

A transaction is about to stall but the maximum number of transactions have stalled and we can't report this one to the event queue (even if non-full). Fields:

streamid unsigned int

StreamID.

streamid_ns bool

The StreamID was for the non-secure world.

trans_id unsigned int

Transaction id.

stall_transaction_resuming

A stalled transaction is resuming. Fields:

stag unsigned int

STAG if appropriate, or if was inhibited by STALL_MAX then 0xFAFA.

stallresult enum

What the transaction resumed to do.

streamid unsigned int

StreamID.

streamid_ns bool

The StreamID was for the non-secure world.

trans_id unsigned int

Transaction id.

start_ptw_read

Page Table Walk (read). This is the start of the physical access that the SMMU is making. Fields:

adomain enum

The actual attributes of the access that was used, after IMP DEF mangling.

aprot enum

The actual attributes of the access that was used, after IMP DEF mangling. PAS[1] (bit[3]), Instruction/Data (bit[2]), PAS[0] (bit[1]), Privileged/User (bit[0]).

desckind enum

The kind of descriptor.

inner_cache enum

The actual attributes of the access that was used, after IMP DEF mangling.

input_address unsigned int

Input address this lookup is for (after TBI normalisation).

mecid unsigned int

The MECID of the TTD fetch.

mpam_pmg_and_partid unsigned int

Bits[23:16] are the MPAM PMG. Bits[15:0] are the physical PARTID.

ns enum

Non-Secure on the bus.

outer_cache enum

The actual attributes of the access that was used, after IMP DEF mangling.

pa_address unsigned int

Physical address.

pas enum

The PAS of the TTD fetch.

ssd enum

SSD of the transaction.

ssd_ns enum

Security state of stream.

streamid unsigned int

The StreamID this is for.

substreamid unsigned int

The SubstreamID this is for.

trans_id unsigned int

Transaction id.

ttb_grain_stage_and_level unsigned int

Which TTB used in bits[31:24]. Which Grain size used in bit[23:16]:- 0 – 4 KiB 1 – 16 KiB 2 – 64 KiB 3 – 64 KiB ARMv8.2 where descriptor bits [15:12] for bits [51:48] of output address 4 – 4 KiB FEAT_LPA2 where address is {bits[9:8], bits[49:12]} 5 – 16 KiB FEAT_LPA2 where address is {bits[9:8], bits[49:14], 2'b00} 6 – 64 KiB ARMv8.2 where descriptor bits [15:12] are ignored Stage in bits[15:8]. Level in bits[7:0], level -1 is 0xF.

ste_cc.STE_commentary

A commentary from the cache about what is going on. Fields:

output string

The stream output.

ste_cc.STE_entry_invalidated

An entry in the cache has been invalidated. Fields:

entry_id unsigned int

The entry id that is being invalidated.

reason enum

Why the entry is being invalidated.

ste_entry_allocated

An STE entry has been allocated. Fields:

entry_id unsigned int

The id of the cache entry. Top bit is set if is a debug entry.

ns enum

For the non-secure world.

ssd enum

The SSD of the entry.

ste string

A textual description of the STE.

streamid unsigned int

The StreamID of the entry this will match.

sup_btm

The pin is driven. This indicates the system supports BTM. This pin is sampled on the negedge of the reset pin and has to be driven beforehand for the value to take effect. Fields:

value bool

The value of the signal.

sup_cohacc

The pin is driven. This indicates the system supports COHACC. This pin is sampled on the negedge of the reset pin and has to be driven beforehand for the value to take effect. Fields:

value bool

The value of the signal.

sup_httu

The pin is driven. This indicates the system supports HTTU. This pin is sampled on the negedge of the reset pin and has to be driven beforehand for the value to take effect. Fields:

value bool

The value of the signal.

sup_sev

The pin is driven. This indicates the system supports SEV. This pin is sampled on the negedge of the reset pin and has to be driven beforehand for the value to take effect. Fields:

value bool

The value of the signal.

tbu0_reset_in

The reset signal of the TBU0. Fields:

value bool

The value of the signal.

tcu_reset_in

The reset signal. Fields:

value bool

The value of the signal.

tlb_entry_allocated

A TLB entry has been allocated. Fields:

aset unsigned int

The ASET of this entry.

asid unsigned int

ASID if appropriate.

index unsigned int

Index of TLB entry.

input_end_incl_address unsigned int

The end inclusive address of the input range that this matches.

input_start_address unsigned int

The start address of the input range that this matches.

output_end_incl_address unsigned int

The end inclusive address of the output range.

output_start_address unsigned int

The start address of the output range.

scheme enum

The tagging scheme used.

ssd enum

The SSD of the tagging scheme.

ssd_ns enum

Security State.

tbi bool

Was the entry formed using Top Byte Ignore (TBI).

vmid unsigned int

VMID if appropriate.

tlb_info_tlb_entries_overlap

A TLB entry was inserted into the TLB and it overlaps an existing entry. This isn't a problem as it was inserted in such a way that it architecturally works. Fields:

end_address_of_new_entry unsigned int

End address of new entry.

end_address_of_old_entry unsigned int

End address of old entry.

how_inserted enum

How the entry was inserted.

index_of_new_entry unsigned int

Index of new entry.

index_of_old_entry unsigned int

Index of old entry.

start_address_of_new_entry unsigned int

Start address of new entry.

start_address_of_old_entry unsigned int

Start address of old entry.

verbose_commentary

This is a verbose commentary on the translation process the SMMU is performing. Fields:

output string

The stream output.

warning_MSI_EVENTQ_address_out_of_range_of_oas

The MSI address is configured outside of the range of the downstream address bus size.
Fields:

address unsigned int

The untruncated address of the MSI.

warning_MSI_GERROR_address_out_of_range_of_oas

The MSI address is configured outside of the range of the downstream address bus size.
Fields:

address unsigned int

The untruncated address of the MSI.

warning_MSI_HACDBS_PROCESSING_COMPLETE_address_out_of_range_of_oas

The MSI address is configured outside of the range of the downstream address bus size.
Fields:

address unsigned int

The untruncated address of the MSI.

warning_MSI_HDBSS_TABLE_FULL_address_out_of_range_of_oas

The MSI address is configured outside of the range of the downstream address bus size.
Fields:

address unsigned int

The untruncated address of the MSI.

warning_MSI_PRIQ_address_out_of_range_of_oas

The MSI address is configured outside of the range of the downstream address bus size.

Fields:

address unsigned int

The untruncated address of the MSI.

warning_MSI_R_EVENTQ_address_out_of_range_of_oas

The MSI address is configured outside of the range of the downstream address bus size.

Fields:

address unsigned int

The untruncated address of the MSI.

warning_MSI_R_GERROR_address_out_of_range_of_oas

The MSI address is configured outside of the range of the downstream address bus size.

Fields:

address unsigned int

The untruncated address of the MSI.

warning_MSI_R_HACDBS_PROCESSING_COMPLETE_address_out_of_range_of_oas

The MSI address is configured outside of the range of the downstream address bus size.

Fields:

address unsigned int

The untruncated address of the MSI.

warning_MSI_R_HDBSS_TABLE_FULL_address_out_of_range_of_oas

The MSI address is configured outside of the range of the downstream address bus size.

Fields:

address unsigned int

The untruncated address of the MSI.

warning_MSI_R_PRIQ_address_out_of_range_of_oas

The MSI address is configured outside of the range of the downstream address bus size.

Fields:

address unsigned int

The untruncated address of the MSI.

warning_MSI_S_EVENTQ_address_out_of_range_of_oas

The MSI address is configured outside of the range of the downstream address bus size.

Fields:

address unsigned int

The untruncated address of the MSI.

warning_MSI_S_GERROR_address_out_of_range_of_oas

The MSI address is configured outside of the range of the downstream address bus size.

Fields:

address unsigned int

The untruncated address of the MSI.

warning_MSI_S_HACDBS_PROCESSING_COMPLETE_address_out_of_range_of_oas

The MSI address is configured outside of the range of the downstream address bus size.

Fields:

address unsigned int

The untruncated address of the MSI.

warning_MSI_S_HDBSS_TABLE_FULL_address_out_of_range_of_oas

The MSI address is configured outside of the range of the downstream address bus size.

Fields:

address unsigned int

The untruncated address of the MSI.

warning_MSI_pmcg_address_out_of_range_of_oas

The MSI Address of the Performance Monitor Counter Group (PMCG) is out of range of the OAS and so will be silently truncated. Fields:

address unsigned int

The untruncated address of the MSI.

pmcg_index unsigned int

Index of the PMCG.

ssd_ns bool

True if this PMCG controlled by the non-secure world.

warning_discarding_interrupt_EVENTQ_as_irqen_low

Interrupt generation is turned off by SMMU_IRQ_CTRL.EVENTQ_IRQEN.

warning_discarding_interrupt_GERROR_as_irqen_low

Interrupt generation is turned off by SMMU_IRQ_CTRL.GERROR_IRQEN.

warning_discarding_interrupt_HACDBS_PROCESSING_COMPLETE_as_irqen_low

Interrupt generation is turned off by SMMU_IRQ_CTRL.HACDBS_IRQEN.

warning_discarding_interrupt_HDBSS_TABLE_FULL_as_irqen_low

Interrupt generation is turned off by SMMU_IRQ_CTRL.HDBSS_IRQEN.

warning_discarding_interrupt_PRIQ_as_irqen_low

Interrupt generation is turned off by SMMU_IRQ_CTRL.PRIQ_IRQEN.

warning_discarding_interrupt_R_EVENTQ_as_irqen_low

Interrupt generation is turned off by SMMU_R_IRQ_CTRL.EVENTQ_IRQEN.

warning_discarding_interrupt_R_GERROR_as_irqen_low

Interrupt generation is turned off by SMMU_R_IRQ_CTRL.GERROR_IRQEN.

warning_discarding_interrupt_R_HACDBS_PROCESSING_COMPLETE_as_irqen_low

Interrupt generation is turned off by SMMU_R_IRQ_CTRL.HACDBS_IRQEN.

warning_discarding_interrupt_R_HDBSS_TABLE_FULL_as_irqen_low

Interrupt generation is turned off by SMMU_R_IRQ_CTRL.HDBSS_IRQEN.

warning_discarding_interrupt_R_PRIQ_as_irqen_low

Interrupt generation is turned off by SMMU_R_IRQ_CTRL.PRIQ_IRQEN.

warning_discarding_interrupt_S_EVENTQ_as_irqen_low

Interrupt generation is turned off by SMMU_S_IRQ_CTRL.EVENTQ_IRQEN.

warning_discarding_interrupt_S_GERROR_as_irqen_low

Interrupt generation is turned off by SMMU_S_IRQ_CTRL.GERROR_IRQEN.

warning_discarding_interrupt_S_HACDBS_PROCESSING_COMPLETE_as_irqen_low

Interrupt generation is turned off by SMMU_IRQ_S_CTRL.HACDBS_IRQEN.

warning_discarding_interrupt_S_HDBSS_TABLE_FULL_as_irqen_low

Interrupt generation is turned off by SMMU_IRQ_S_CTRL.HDBSS_IRQEN.

warning_ns_gerrorn_bad_acknowledge

A GERROR was acknowledged by SW in the GERRORN register that did not have an active error. The result is **UNPREDICTABLE** if this will generate an interrupt or not. Fields:

which enum

Which GERROR was acknowledge when there was no active error.

warning_reg_after_doesnt_match_written_value

A write occurred that tried to set bits in a register, that for one reason or another, failed to get written. Fields:

desc string

The textual description of what happened.

warning_rl_gerrorn_bad_acknowledge

A GERROR was acknowledged by SW in the GERRORN register that did not have an active error. The result is **UNPREDICTABLE** if this will generate an interrupt or not. Fields:

which enum

Which GERROR was acknowledge when there was no active error.

warning_s_gerrorn_bad_acknowledge

A GERROR was acknowledged by SW in the GERRORN register that did not have an active error. The result is **UNPREDICTABLE** if this will generate an interrupt or not. Fields:

which_enum

Which GERROR was acknowledge when there was no active error.

what_going_to_do_with_terminated_event

A terminating transaction has produced an event, this tells you what the model is going to do with the event. Fields:

CD.S bool

The CD.S field if available.

S2 bool

The event is related to Stage 2.

STE.S1STALLD bool

The STE.S1STALLD field if available.

STE.S2S bool

The STE.S2S field if available.

aborts bool

The transaction will abort.

axmmuflow_enum

The AxMMUFLOW for this transaction group. stallable – allow the transaction to be stalled if configured. translated-access – is a PCIe Translated Access non-stallable – do not stall transaction even if configured. transfault-flow – do not report for Translation Related Faults, but tell client device to use a page request mechanism instead (for example PRI).

is_tr_fault bool

Is a Translation Related fault.

protected_mode bool

The transaction is protected-mode. As such, it cannot stall and will obey the report configuration bits.

reports bool

The transaction will attempt to report.

ssd_enum

The SSD of the transaction.

ssd_ns bool

The transaction is classified as SSD non-secure.

supports_stall_model bool

The implementation supports the stall model.

trans_id unsigned int

The transaction id.

why_abort_decision_enum

The reason why the transaction aborted/did not abort.

why_report_decision enum

The reason why the transaction reported/did not report.

2.129 Mali_C10

This section describes the trace sources.

ISP_Irq

IRQ signal %{IRQ_STATE:(SET|CLEAR)}. Fields:

IRQ_STATE bool

The IRQ state.

ISP_Read

Register address: %{REG_ADDRESS} Value: %{VALUE}. Fields:

REG_ADDRESS unsigned int

The address of the register read.

VALUE unsigned int

Read Value from the register.

ISP_Reset

Reset %{IN_RESET:(YES|NO)}. Fields:

IN_RESET bool

Reset state is IN_RESET or NOT_IN_RESET.

ISP_Write

Register address: %{REG_ADDRESS} Value %{VALUE}. Fields:

REG_ADDRESS unsigned int

The address of the register write.

VALUE unsigned int

Write Value to the register.

2.130 Mali_C55

This section describes the trace sources.

ISP_Irq

IRQ signal %{IRQ_STATE:(SET|CLEAR)}. Fields:

IRQ_STATE bool

The IRQ state.

ISP_Read

Register address: %{REG_ADDRESS} Value: %{VALUE}. Fields:

REG_ADDRESS unsigned int

The address of the register read.

VALUE unsigned int

Read Value from the register.

ISP_Reset

Reset %{IN_RESET:(YES|NO)}. Fields:

IN_RESET bool

Reset state is IN_RESET or NOT_IN_RESET.

ISP_Write

Register address: %{REG_ADDRESS} Value %{VALUE}. Fields:

REG_ADDRESS unsigned int

The address of the register write.

VALUE unsigned int

Write Value to the register.

2.131 Mali_C71

This section describes the trace sources.

ISP_Irq

IRQ signal %{IRQ_STATE:(SET|CLEAR)}. Fields:

IRQ_STATE bool

The IRQ state.

ISP_Read

Register address: %{REG_ADDRESS} Value: %{VALUE}. Fields:

REG_ADDRESS unsigned int

The address of the register read.

VALUE unsigned int

Read Value from the register.

ISP_Reset

Reset %{IN_RESET:(YES|NO)}. Fields:

IN_RESET bool

Reset state is IN_RESET or NOT_IN_RESET.

ISP_Write

Register address: %{REG_ADDRESS} Value %{VALUE}. Fields:

REG_ADDRESS unsigned int

The address of the register write.

VALUE unsigned int

Write Value to the register.

2.132 Mali_C720AE

This section describes the trace sources.

ISP_Irq

IRQ signal %{IRQ_STATE:(SET|CLEAR)}. Fields:

IRQ_STATE bool

The IRQ state.

ISP_Read

Register address: %{REG_ADDRESS} Value: %{VALUE}. Fields:

REG_ADDRESS unsigned int

The address of the register read.

VALUE unsigned int

Read Value from the register.

ISP_Reset

Reset %{IN_RESET:(YES|NO)}. Fields:

IN_RESET bool

Reset state is IN_RESET or NOT_IN_RESET.

ISP_Write

Register address: %{REG_ADDRESS} Value %{VALUE}. Fields:

REG_ADDRESS unsigned int

The address of the register write.

VALUE unsigned int

Write Value to the register.

2.133 Mali_C78

This section describes the trace sources.

ISP_Irq

IRQ signal %{IRQ_STATE:(SET|CLEAR)}. Fields:

IRQ_STATE bool

The IRQ state.

ISP_Read

Register address: %{REG_ADDRESS} Value: %{VALUE}. Fields:

REG_ADDRESS unsigned int

The address of the register read.

VALUE unsigned int

Read Value from the register.

ISP_Reset

Reset %{IN_RESET:(YES|NO)}. Fields:

IN_RESET bool

Reset state is IN_RESET or NOT_IN_RESET.

ISP_Write

Register address: %{REG_ADDRESS} Value %{VALUE}. Fields:

REG_ADDRESS unsigned int

The address of the register write.

VALUE unsigned int

Write Value to the register.

2.134 Mali_G1

This section describes the trace sources.

INFO_AccessInReset

GPU Register %{READ:(WRITE|READ)} to address: %{ADDR} FAILED as GPU is being reset.
Fields:

ADDR unsigned int

Address of the accessed register.

READ bool

Transaction type is read or write.

INFO_GpuDmiAcquired

GPU acquired DMI over address range [%{ADDR_LO},%{ADDR_HI}]. Fields:

ADDR_HI unsigned int

High-address of the DMI pointer.

ADDR_LO unsigned int

Low-address of the DMI pointer.

INFO_GpuDmiFailed

GPU failed to acquire DMI over address range [%{ADDR_LO},%{ADDR_HI}]. Fields:

ADDR_HI unsigned int

High-address of the DMI pointer.

ADDR_LO unsigned int

Low-address of the DMI pointer.

INFO_GpuDmiRevoked

GPU lost DMI over address range [%{ADDR_LO},%{ADDR_HI}]. Fields:

ADDR_HI unsigned int

High-address of the DMI pointer.

ADDR_LO unsigned int

Low-address of the DMI pointer.

INFO_GpuStatsMemoryAccess

Statistics about memory access performance, emitted every 1000000 accesses. Fields:

READ_ACCESSES_WITHOUT_DMI unsigned int

Total number of read accesses performed without DMI.

READ_ACCESSES_WITH_DMI unsigned int

Total number of read accesses performed with DMI.

WRITE_ACCESSES_WITHOUT_DMI unsigned int

Total number of write accesses performed without DMI.

WRITE_ACCESSES_WITH_DMI unsigned int

Total number of write accesses performed with DMI.

INFO_Irq

IRQ signal received (%{IRQ_NAME}) state %{IRQ_STATE:(SET|CLEAR)}. Fields:

IRQ_NAME string

The IRQ name.

IRQ_STATE bool

The IRQ state.

INFO_ReadRegister

GPU Register Offset: %{REG_OFFSET} Value: %{VALUE} RegName: %{REG_NAME}. Fields:

REG_NAME string

The name of the register read.

REG_OFFSET unsigned int

The address of the register read.

VALUE unsigned int

Read Value from the register.

INFO_Reset

GPU Reset %{STATE:(START|END)}. Fields:

STATE bool

Reset state is start or end.

INFO_WriteRegister

GPU Register Offset: %{REG_OFFSET} Updated: From value %{VALUE} To value %{UPDATED_VALUE} RegName: %{REG_NAME}. Fields:

REG_NAME string

The name of the register updated.

REG_OFFSET unsigned int

The address of the register updated.

UPDATED_VALUE unsigned int

New Value in the register.

VALUE unsigned int

Old Value in the register.

WARN_AccessToUnimplementedRegister

GPU Register %{READ:(WRITE|READ)} to address: %{ADDR} FAILED as the register is unimplemented. Fields:

ADDR unsigned int

Address of the accessed register.

READ bool

Transaction type is read or write.

WARN_ReadToWriteOnlyRegister

GPU Register Read to address: %{ADDR} FAILED as the register is write-only RegName: %{REG_NAME}. Fields:

ADDR unsigned int

Address of the accessed register.

REG_NAME string

The name of the register updated.

WARN_WriteToReadOnlyRegister

GPU Register Write to address: %{ADDR} FAILED as the register is read-only RegName: %{REG_NAME}. Fields:

ADDR unsigned int

Address of the accessed register.

REG_NAME string

The name of the register updated.

2.135 Mali_G71

This section describes the trace sources.

INFO_AccessInReset

GPU Register %{READ:(WRITE|READ)} to address: %{ADDR} FAILED as GPU is being reset. Fields:

ADDR unsigned int

Address of the accessed register.

READ bool

Transaction type is read or write.

INFO_GpuDmiAcquired

GPU acquired DMI over address range [%{ADDR_LO},%{ADDR_HI}]. Fields:

ADDR_HI unsigned int

High-address of the DMI pointer.

ADDR_LO unsigned int

Low-address of the DMI pointer.

INFO_GpuDmiFailed

GPU failed to acquire DMI over address range [%{ADDR_LO},%{ADDR_HI}]. Fields:

ADDR_HI unsigned int

High-address of the DMI pointer.

ADDR_LO unsigned int

Low-address of the DMI pointer.

INFO_GpuDmiRevoked

GPU lost DMI over address range [%{ADDR_LO},%{ADDR_HI}]. Fields:

ADDR_HI unsigned int

High-address of the DMI pointer.

ADDR_LO unsigned int

Low-address of the DMI pointer.

INFO_GpuStatsMemoryAccess

Statistics about memory access performance, emitted every 1000000 accesses. Fields:

READ_ACCESSSES_WITHOUT_DMI unsigned int

Total number of read accesses performed without DMI.

READ_ACCESSSES_WITH_DMI unsigned int

Total number of read accesses performed with DMI.

WRITE_ACCESSSES_WITHOUT_DMI unsigned int

Total number of write accesses performed without DMI.

WRITE_ACCESSSES_WITH_DMI unsigned int

Total number of write accesses performed with DMI.

INFO_Irq

IRQ signal received ({IRQ_NAME}) state {IRQ_STATE:(SET|CLEAR)}. Fields:

IRQ_NAME string

The IRQ name.

IRQ_STATE bool

The IRQ state.

INFO_ReadRegister

GPU Register Offset: {REG_OFFSET} Value: {VALUE} RegName: {REG_NAME}. Fields:

REG_NAME string

The name of the register read.

REG_OFFSET unsigned int

The address of the register read.

VALUE unsigned int

Read Value from the register.

INFO_Reset

GPU Reset {STATE:(START|END)}. Fields:

STATE bool

Reset state is start or end.

INFO_WriteRegister

GPU Register Offset: {REG_OFFSET} Updated: From value {VALUE} To value {UPDATED_VALUE} RegName: {REG_NAME}. Fields:

REG_NAME string

The name of the register updated.

REG_OFFSET unsigned int

The address of the register updated.

UPDATED_VALUE unsigned int

New Value in the register.

VALUE unsigned int

Old Value in the register.

WARN_AccessToUnimplementedRegister

GPU Register %{READ:(WRITE|READ)} to address: %{ADDR} FAILED as the register is unimplemented. Fields:

ADDR unsigned int

Address of the accessed register.

READ bool

Transaction type is read or write.

WARN_ReadToWriteOnlyRegister

GPU Register Read to address: %{ADDR} FAILED as the register is write-only RegName: %{REG_NAME}. Fields:

ADDR unsigned int

Address of the accessed register.

REG_NAME string

The name of the register updated.

WARN_WriteToReadOnlyRegister

GPU Register Write to address: %{ADDR} FAILED as the register is read-only RegName: %{REG_NAME}. Fields:

ADDR unsigned int

Address of the accessed register.

REG_NAME string

The name of the register updated.

2.136 Mali_G710

This section describes the trace sources.

INFO_AccessInReset

GPU Register %{READ:(WRITE|READ)} to address: %{ADDR} FAILED as GPU is being reset. Fields:

ADDR unsigned int

Address of the accessed register.

READ bool

Transaction type is read or write.

INFO_GpuDmiAcquired

GPU acquired DMI over address range [%{ADDR_LO},%{ADDR_HI}]. Fields:

ADDR_HI unsigned int

High-address of the DMI pointer.

ADDR_LO unsigned int

Low-address of the DMI pointer.

INFO_GpuDmiFailed

GPU failed to acquire DMI over address range [%{ADDR_LO},%{ADDR_HI}]. Fields:

ADDR_HI unsigned int

High-address of the DMI pointer.

ADDR_LO unsigned int

Low-address of the DMI pointer.

INFO_GpuDmiRevoked

GPU lost DMI over address range [%{ADDR_LO},%{ADDR_HI}]. Fields:

ADDR_HI unsigned int

High-address of the DMI pointer.

ADDR_LO unsigned int

Low-address of the DMI pointer.

INFO_GpuStatsMemoryAccess

Statistics about memory access performance, emitted every 1000000 accesses. Fields:

READ_ACCESSES_WITHOUT_DMI unsigned int

Total number of read accesses performed without DMI.

READ_ACCESSES_WITH_DMI unsigned int

Total number of read accesses performed with DMI.

WRITE_ACCESSES_WITHOUT_DMI unsigned int

Total number of write accesses performed without DMI.

WRITE_ACCESSES_WITH_DMI unsigned int

Total number of write accesses performed with DMI.

INFO_Irq

IRQ signal received (%{IRQ_NAME}) state %{IRQ_STATE:(SET|CLEAR)}. Fields:

IRQ_NAME string

The IRQ name.

IRQ_STATE bool

The IRQ state.

INFO_ReadRegister

GPU Register Offset: %{REG_OFFSET} Value: %{VALUE} RegName: %{REG_NAME}. Fields:

REG_NAME string

The name of the register read.

REG_OFFSET unsigned int

The address of the register read.

VALUE unsigned int

Read Value from the register.

INFO_Reset

GPU Reset %{STATE:(START|END)}. Fields:

STATE bool

Reset state is start or end.

INFO_WriteRegister

GPU Register Offset: %{REG_OFFSET} Updated: From value %{VALUE} To value %{UPDATED_VALUE} RegName: %{REG_NAME}. Fields:

REG_NAME string

The name of the register updated.

REG_OFFSET unsigned int

The address of the register updated.

UPDATED_VALUE unsigned int

New Value in the register.

VALUE unsigned int

Old Value in the register.

WARN_AccessToUnimplementedRegister

GPU Register %{READ:(WRITE|READ)} to address: %{ADDR} FAILED as the register is unimplemented. Fields:

ADDR unsigned int

Address of the accessed register.

READ bool

Transaction type is read or write.

WARN_ReadToWriteOnlyRegister

GPU Register Read to address: %{ADDR} FAILED as the register is write-only RegName: %{REG_NAME}. Fields:

ADDR unsigned int

Address of the accessed register.

REG_NAME string

The name of the register updated.

WARN_WriteToReadOnlyRegister

GPU Register Write to address: %{ADDR} FAILED as the register is read-only RegName: %{REG_NAME}. Fields:

ADDR unsigned int

Address of the accessed register.

REG_NAME string

The name of the register updated.

2.137 Mali_G715

This section describes the trace sources.

INFO_AccessInReset

GPU Register %{READ:(WRITE|READ)} to address: %{ADDR} FAILED as GPU is being reset. Fields:

ADDR unsigned int

Address of the accessed register.

READ bool

Transaction type is read or write.

INFO_GpuDmiAcquired

GPU acquired DMI over address range [%{ADDR_LO},%{ADDR_HI}]. Fields:

ADDR_HI unsigned int

High-address of the DMI pointer.

ADDR_LO unsigned int

Low-address of the DMI pointer.

INFO_GpuDmiFailed

GPU failed to acquire DMI over address range [%{ADDR_LO},%{ADDR_HI}]. Fields:

ADDR_HI unsigned int

High-address of the DMI pointer.

ADDR_LO unsigned int

Low-address of the DMI pointer.

INFO_GpuDmiRevoked

GPU lost DMI over address range [%{ADDR_LO},%{ADDR_HI}]. Fields:

ADDR_HI unsigned int

High-address of the DMI pointer.

ADDR_LO unsigned int

Low-address of the DMI pointer.

INFO_GpuStatsMemoryAccess

Statistics about memory access performance, emitted every 1000000 accesses. Fields:

READ_ACCESSSES_WITHOUT_DMI unsigned int

Total number of read accesses performed without DMI.

READ_ACCESSSES_WITH_DMI unsigned int

Total number of read accesses performed with DMI.

WRITE_ACCESSSES_WITHOUT_DMI unsigned int

Total number of write accesses performed without DMI.

WRITE_ACCESSSES_WITH_DMI unsigned int

Total number of write accesses performed with DMI.

INFO_Irq

IRQ signal received ({IRQ_NAME}) state {IRQ_STATE:(SET|CLEAR)}. Fields:

IRQ_NAME string

The IRQ name.

IRQ_STATE bool

The IRQ state.

INFO_ReadRegister

GPU Register Offset: {REG_OFFSET} Value: {VALUE} RegName: {REG_NAME}. Fields:

REG_NAME string

The name of the register read.

REG_OFFSET unsigned int

The address of the register read.

VALUE unsigned int

Read Value from the register.

INFO_Reset

GPU Reset {STATE:(START|END)}. Fields:

STATE bool

Reset state is start or end.

INFO_WriteRegister

GPU Register Offset: {REG_OFFSET} Updated: From value {VALUE} To value {UPDATED_VALUE} RegName: {REG_NAME}. Fields:

REG_NAME string

The name of the register updated.

REG_OFFSET unsigned int

The address of the register updated.

UPDATED_VALUE unsigned int

New Value in the register.

VALUE unsigned int

Old Value in the register.

WARN_AccessToUnimplementedRegister

GPU Register %{READ:(WRITE|READ)} to address: %{ADDR} FAILED as the register is unimplemented. Fields:

ADDR unsigned int

Address of the accessed register.

READ bool

Transaction type is read or write.

WARN_ReadToWriteOnlyRegister

GPU Register Read to address: %{ADDR} FAILED as the register is write-only RegName: %{REG_NAME}. Fields:

ADDR unsigned int

Address of the accessed register.

REG_NAME string

The name of the register updated.

WARN_WriteToReadOnlyRegister

GPU Register Write to address: %{ADDR} FAILED as the register is read-only RegName: %{REG_NAME}. Fields:

ADDR unsigned int

Address of the accessed register.

REG_NAME string

The name of the register updated.

2.138 Mali_G720

This section describes the trace sources.

INFO_AccessInReset

GPU Register %{READ:(WRITE|READ)} to address: %{ADDR} FAILED as GPU is being reset. Fields:

ADDR unsigned int

Address of the accessed register.

READ bool

Transaction type is read or write.

INFO_GpuDmiAcquired

GPU acquired DMI over address range [%{ADDR_LO},%{ADDR_HI}]. Fields:

ADDR_HI unsigned int

High-address of the DMI pointer.

ADDR_LO unsigned int

Low-address of the DMI pointer.

INFO_GpuDmiFailed

GPU failed to acquire DMI over address range [%{ADDR_LO},%{ADDR_HI}]. Fields:

ADDR_HI unsigned int

High-address of the DMI pointer.

ADDR_LO unsigned int

Low-address of the DMI pointer.

INFO_GpuDmiRevoked

GPU lost DMI over address range [%{ADDR_LO},%{ADDR_HI}]. Fields:

ADDR_HI unsigned int

High-address of the DMI pointer.

ADDR_LO unsigned int

Low-address of the DMI pointer.

INFO_GpuStatsMemoryAccess

Statistics about memory access performance, emitted every 1000000 accesses. Fields:

READ_ACCESSES_WITHOUT_DMI unsigned int

Total number of read accesses performed without DMI.

READ_ACCESSES_WITH_DMI unsigned int

Total number of read accesses performed with DMI.

WRITE_ACCESSES_WITHOUT_DMI unsigned int

Total number of write accesses performed without DMI.

WRITE_ACCESSES_WITH_DMI unsigned int

Total number of write accesses performed with DMI.

INFO_Irq

IRQ signal received (%{IRQ_NAME}) state %{IRQ_STATE:(SET|CLEAR)}. Fields:

IRQ_NAME string

The IRQ name.

IRQ_STATE bool

The IRQ state.

INFO_ReadRegister

GPU Register Offset: %{REG_OFFSET} Value: %{VALUE} RegName: %{REG_NAME}. Fields:

REG_NAME string

The name of the register read.

REG_OFFSET unsigned int

The address of the register read.

VALUE unsigned int

Read Value from the register.

INFO_Reset

GPU Reset %{STATE:(START|END)}. Fields:

STATE bool

Reset state is start or end.

INFO_WriteRegister

GPU Register Offset: %{REG_OFFSET} Updated: From value %{VALUE} To value %{UPDATED_VALUE} RegName: %{REG_NAME}. Fields:

REG_NAME string

The name of the register updated.

REG_OFFSET unsigned int

The address of the register updated.

UPDATED_VALUE unsigned int

New Value in the register.

VALUE unsigned int

Old Value in the register.

WARN_AccessToUnimplementedRegister

GPU Register %{READ:(WRITE|READ)} to address: %{ADDR} FAILED as the register is unimplemented. Fields:

ADDR unsigned int

Address of the accessed register.

READ bool

Transaction type is read or write.

WARN_ReadToWriteOnlyRegister

GPU Register Read to address: %{ADDR} FAILED as the register is write-only RegName: %{REG_NAME}. Fields:

ADDR unsigned int

Address of the accessed register.

REG_NAME string

The name of the register updated.

WARN_WriteToReadOnlyRegister

GPU Register Write to address: %{ADDR} FAILED as the register is read-only RegName: %{REG_NAME}. Fields:

ADDR unsigned int

Address of the accessed register.

REG_NAME string

The name of the register updated.

2.139 Mali_G725

This section describes the trace sources.

INFO_AccessInReset

GPU Register %{READ:(WRITE|READ)} to address: %{ADDR} FAILED as GPU is being reset. Fields:

ADDR unsigned int

Address of the accessed register.

READ bool

Transaction type is read or write.

INFO_GpuDmiAcquired

GPU acquired DMI over address range [%{ADDR_LO},%{ADDR_HI}]. Fields:

ADDR_HI unsigned int

High-address of the DMI pointer.

ADDR_LO unsigned int

Low-address of the DMI pointer.

INFO_GpuDmiFailed

GPU failed to acquire DMI over address range [%{ADDR_LO},%{ADDR_HI}]. Fields:

ADDR_HI unsigned int

High-address of the DMI pointer.

ADDR_LO unsigned int

Low-address of the DMI pointer.

INFO_GpuDmiRevoked

GPU lost DMI over address range [%{ADDR_LO},%{ADDR_HI}]. Fields:

ADDR_HI unsigned int

High-address of the DMI pointer.

ADDR_LO unsigned int

Low-address of the DMI pointer.

INFO_GpuStatsMemoryAccess

Statistics about memory access performance, emitted every 1000000 accesses. Fields:

READ_ACCESSSES_WITHOUT_DMI unsigned int

Total number of read accesses performed without DMI.

READ_ACCESSSES_WITH_DMI unsigned int

Total number of read accesses performed with DMI.

WRITE_ACCESSSES_WITHOUT_DMI unsigned int

Total number of write accesses performed without DMI.

WRITE_ACCESSSES_WITH_DMI unsigned int

Total number of write accesses performed with DMI.

INFO_Irq

IRQ signal received ({IRQ_NAME}) state {IRQ_STATE:(SET|CLEAR)}. Fields:

IRQ_NAME string

The IRQ name.

IRQ_STATE bool

The IRQ state.

INFO_ReadRegister

GPU Register Offset: {REG_OFFSET} Value: {VALUE} RegName: {REG_NAME}. Fields:

REG_NAME string

The name of the register read.

REG_OFFSET unsigned int

The address of the register read.

VALUE unsigned int

Read Value from the register.

INFO_Reset

GPU Reset {STATE:(START|END)}. Fields:

STATE bool

Reset state is start or end.

INFO_WriteRegister

GPU Register Offset: {REG_OFFSET} Updated: From value {VALUE} To value {UPDATED_VALUE} RegName: {REG_NAME}. Fields:

REG_NAME string

The name of the register updated.

REG_OFFSET unsigned int

The address of the register updated.

UPDATED_VALUE unsigned int

New Value in the register.

VALUE unsigned int

Old Value in the register.

WARN_AccessToUnimplementedRegister

GPU Register %{READ:(WRITE|READ)} to address: %{ADDR} FAILED as the register is unimplemented. Fields:

ADDR unsigned int

Address of the accessed register.

READ bool

Transaction type is read or write.

WARN_ReadToWriteOnlyRegister

GPU Register Read to address: %{ADDR} FAILED as the register is write-only RegName: %{REG_NAME}. Fields:

ADDR unsigned int

Address of the accessed register.

REG_NAME string

The name of the register updated.

WARN_WriteToReadOnlyRegister

GPU Register Write to address: %{ADDR} FAILED as the register is read-only RegName: %{REG_NAME}. Fields:

ADDR unsigned int

Address of the accessed register.

REG_NAME string

The name of the register updated.

2.140 Mali_G76

This section describes the trace sources.

INFO_AccessInReset

GPU Register %{READ:(WRITE|READ)} to address: %{ADDR} FAILED as GPU is being reset. Fields:

ADDR unsigned int

Address of the accessed register.

READ bool

Transaction type is read or write.

INFO_GpuDmiAcquired

GPU acquired DMI over address range [%{ADDR_LO},%{ADDR_HI}]. Fields:

ADDR_HI unsigned int

High-address of the DMI pointer.

ADDR_LO unsigned int

Low-address of the DMI pointer.

INFO_GpuDmiFailed

GPU failed to acquire DMI over address range [%{ADDR_LO},%{ADDR_HI}]. Fields:

ADDR_HI unsigned int

High-address of the DMI pointer.

ADDR_LO unsigned int

Low-address of the DMI pointer.

INFO_GpuDmiRevoked

GPU lost DMI over address range [%{ADDR_LO},%{ADDR_HI}]. Fields:

ADDR_HI unsigned int

High-address of the DMI pointer.

ADDR_LO unsigned int

Low-address of the DMI pointer.

INFO_GpuStatsMemoryAccess

Statistics about memory access performance, emitted every 1000000 accesses. Fields:

READ_ACCESSES_WITHOUT_DMI unsigned int

Total number of read accesses performed without DMI.

READ_ACCESSES_WITH_DMI unsigned int

Total number of read accesses performed with DMI.

WRITE_ACCESSES_WITHOUT_DMI unsigned int

Total number of write accesses performed without DMI.

WRITE_ACCESSES_WITH_DMI unsigned int

Total number of write accesses performed with DMI.

INFO_Irq

IRQ signal received (%{IRQ_NAME}) state %{IRQ_STATE:(SET|CLEAR)}. Fields:

IRQ_NAME string

The IRQ name.

IRQ_STATE bool

The IRQ state.

INFO_ReadRegister

GPU Register Offset: %{REG_OFFSET} Value: %{VALUE} RegName: %{REG_NAME}. Fields:

REG_NAME string

The name of the register read.

REG_OFFSET unsigned int

The address of the register read.

VALUE unsigned int

Read Value from the register.

INFO_Reset

GPU Reset %{STATE:(START|END)}. Fields:

STATE bool

Reset state is start or end.

INFO_WriteRegister

GPU Register Offset: %{REG_OFFSET} Updated: From value %{VALUE} To value %{UPDATED_VALUE} RegName: %{REG_NAME}. Fields:

REG_NAME string

The name of the register updated.

REG_OFFSET unsigned int

The address of the register updated.

UPDATED_VALUE unsigned int

New Value in the register.

VALUE unsigned int

Old Value in the register.

WARN_AccessToUnimplementedRegister

GPU Register %{READ:(WRITE|READ)} to address: %{ADDR} FAILED as the register is unimplemented. Fields:

ADDR unsigned int

Address of the accessed register.

READ bool

Transaction type is read or write.

WARN_ReadToWriteOnlyRegister

GPU Register Read to address: %{ADDR} FAILED as the register is write-only RegName: %{REG_NAME}. Fields:

ADDR unsigned int

Address of the accessed register.

REG_NAME string

The name of the register updated.

WARN_WriteToReadOnlyRegister

GPU Register Write to address: %{ADDR} FAILED as the register is read-only RegName: %{REG_NAME}. Fields:

ADDR unsigned int

Address of the accessed register.

REG_NAME string

The name of the register updated.

2.141 Mali_G78AE

This section describes the trace sources.

CloseWindow

Access Window Index: %{AW_ID} Partition Index: %{PT_ID} Owinging Group Index: %{GROUP_ID}. Fields:

AW_ID signed int

Closed access window index.

GROUP_ID signed int

Resource group that owns the window/partition.

PT_ID signed int

Partition index for which the window was closed.

MessageToGroup

Originating Window: \${AW_ID} Receiving Group \${GROUP_ID} Value 0: %{VAL0} Value 1: %{VAL1}. Fields:

AW_ID signed int

Originating Access Window.

GROUP_ID signed int

Destination Resource Group.

VAL0 unsigned int

Message Value 0.

VAL1 unsigned int

Message Value 1.

MessageToWindow

Originating Group: \${GROUP_ID} Receiving Window \${AW_ID} Value 0: %{VAL0} Value 1: %{VAL1}. Fields:

AW_ID signed int

Destination Access Window.

GROUP_ID signed int

Originating Resource Group.

VAL0 unsigned int

Message Value 0.

VAL1 unsigned int

Message Value 1.

OpenWindow

Access Window Index: %{AW_ID} Partition Index: %{PT_ID} Owning Group Index: %{GROUP_ID} Stream ID: \${SID} Protected Stream ID: %{PSID}. Fields:

AW_ID signed int

Opened access window index.

GROUP_ID signed int

Resource group that owns the window/partition.

PSID unsigned int

Protected StreamID of the window.

PT_ID signed int

Partition index for which the window was opened.

SID unsigned int

StreamID of the window.

PartitionIRQ

IRQ type: %{TYPE} State: %{STATE} Partition index: %{PARTITION} Access Window index: %{WINDOW}. Fields:

PARTITION unsigned int

Index of the partition that raised the interrupt.

STATE enum

Interrupt state.

TYPE enum

Type of the interrupt.

WINDOW unsigned int

Access Window opened for the partition.

PartitionManagerReset

Reset Type: %{TYPE}. Fields:

TYPE enum

Type of the reset.

PartitionResetSignal

Partition Index: %{PARTITION} Signal State: %{STATE}. Fields:

INDEX unsigned int

Partition Index.

STATE enum

Reset signal state.

ReadRegister

PTM Register Page: \${PAGE_TYPE} Page Index \${PAGE_ID} GPU Register Offset: \${REG_OFFSET} Value: \${VALUE}. Fields:

PAGE_ID unsigned int

Index of the PTM register page (0 if only one instance exists).

PAGE_TYPE string

Type of the PTM register page.

REG_OFFSET unsigned int

Offset into the register page.

VALUE unsigned int

Read Value from the register.

RemapDecision

Access Address: \${ADDR} Decision: \${DECISION}. Fields:

ADDR unsigned int

Address of memory access.

DECISION enum

What remap decision has been made.

RemapDecisionRevoked

Reason for revoke \${REASON}. Fields:

REASON enum

Reason for the revoke.

ResourceIRQ

IRQ type: \${TYPE} Index: \${INDEX} State: \${STATE} Status Register: \${STATUS}. Fields:

INDEX unsigned int

IRQ index.

STATE enum

Interrupt state.

STATUS unsigned int

IRQ status register.

TYPE enum

Type of the interrupt.

SystemIRQ

IRQ type: %{TYPE} State: %{STATE} Status Register 0: %{STATUS0} Status Register 1: %{STATUS1} Status Register 2: %{STATUS2}. Fields:

STATE enum

Interrupt state.

STATUS0 unsigned int

IRQ status register 0.

STATUS1 unsigned int

IRQ status register 1.

STATUS2 unsigned int

IRQ status register 2.

TYPE enum

Type of the interrupt.

WriteRegister

PTM Register Page: \${PAGE_TYPE} Page Index \${PAGE_ID} GPU Register Offset: %{REG_OFFSET} Value: %{VALUE}. Fields:

PAGE_ID unsigned int

Index of the PTM register page (0 if only one instance exists).

PAGE_TYPE string

Type of the PTM register page.

REG_OFFSET unsigned int

Offset into the register page.

VALUE unsigned int

Value written to the register.

2.142 Mali_T624

This section describes the trace sources.

INFO_AccessInReset

GPU Register %{READ:(WRITE|READ)} to address: %{ADDR} FAILED as GPU is being reset. Fields:

ADDR unsigned int

Address of the accessed register.

READ bool

Transaction type is read or write.

INFO_GpuDmiAcquired

GPU acquired DMI over address range [%{ADDR_LO},%{ADDR_HI}]. Fields:

ADDR_HI unsigned int

High-address of the DMI pointer.

ADDR_LO unsigned int

Low-address of the DMI pointer.

INFO_GpuDmiFailed

GPU failed to acquire DMI over address range [%{ADDR_LO},%{ADDR_HI}]. Fields:

ADDR_HI unsigned int

High-address of the DMI pointer.

ADDR_LO unsigned int

Low-address of the DMI pointer.

INFO_GpuDmiRevoked

GPU lost DMI over address range [%{ADDR_LO},%{ADDR_HI}]. Fields:

ADDR_HI unsigned int

High-address of the DMI pointer.

ADDR_LO unsigned int

Low-address of the DMI pointer.

INFO_GpuStatsMemoryAccess

Statistics about memory access performance, emitted every 1000000 accesses. Fields:

READ_ACCESSES_WITHOUT_DMI unsigned int

Total number of read accesses performed without DMI.

READ_ACCESSES_WITH_DMI unsigned int

Total number of read accesses performed with DMI.

WRITE_ACCESSES_WITHOUT_DMI unsigned int

Total number of write accesses performed without DMI.

WRITE_ACCESSES_WITH_DMI unsigned int

Total number of write accesses performed with DMI.

INFO_Irq

IRQ signal received (%{IRQ_NAME}) state %{IRQ_STATE:(SET|CLEAR)}. Fields:

IRQ_NAME string

The IRQ name.

IRQ_STATE bool

The IRQ state.

INFO_ReadRegister

GPU Register Offset: %{REG_OFFSET} Value: %{VALUE} RegName: %{REG_NAME}. Fields:

REG_NAME string

The name of the register read.

REG_OFFSET unsigned int

The address of the register read.

VALUE unsigned int

Read Value from the register.

INFO_Reset

GPU Reset %{STATE:(START|END)}. Fields:

STATE bool

Reset state is start or end.

INFO_WriteRegister

GPU Register Offset: %{REG_OFFSET} Updated: From value %{VALUE} To value %{UPDATED_VALUE} RegName: %{REG_NAME}. Fields:

REG_NAME string

The name of the register updated.

REG_OFFSET unsigned int

The address of the register updated.

UPDATED_VALUE unsigned int

New Value in the register.

VALUE unsigned int

Old Value in the register.

WARN_AccessToUnimplementedRegister

GPU Register %{READ:(WRITE|READ)} to address: %{ADDR} FAILED as the register is unimplemented. Fields:

ADDR unsigned int

Address of the accessed register.

READ bool

Transaction type is read or write.

WARN_ReadToWriteOnlyRegister

GPU Register Read to address: %{ADDR} FAILED as the register is write-only RegName: %{REG_NAME}. Fields:

ADDR unsigned int

Address of the accessed register.

REG_NAME string

The name of the register updated.

WARN_WriteToReadOnlyRegister

GPU Register Write to address: %{ADDR} FAILED as the register is read-only RegName: %{REG_NAME}. Fields:

ADDR unsigned int

Address of the accessed register.

REG_NAME string

The name of the register updated.

2.143 MessageHandlingUnitV2

This section describes the trace sources.

RECV_ArchMsg.Warning.MemoryMapped_AccessAborted

DISPLAY Receiver is asleep, access to message registers BLOCKED, address=%{VALUE}.

Fields:

OFFSET unsigned int

Offset of address within the MHU.

RECV_ArchMsg.Warning.MemoryMapped_ReadReserved

DISPLAY Reserved location attempting to be read. Fields:

OFFSET unsigned int

Offset of address within the MHU.

RECV_ArchMsg.Warning.MemoryMapped_ReadWriteOnlyReg

DISPLAY Write-only register %{REG_NAME} attempting to be read. Fields:

OFFSET unsigned int

Offset of address within the MHU.

REG_NAME string

Name of the register being read.

RECV_ArchMsg.Warning.MemoryMapped_WriteReadOnlyReg

DISPLAY Read-only register %{REG_NAME} attempting to be written with value %{VALUE}.

Fields:

OFFSET unsigned int

Offset of address within the MHU.

REG_NAME string

Name of the register being written.

VALUE unsigned int

Value written.

RECV_ArchMsg.Warning.MemoryMapped_WriteReserved

DISPLAY Reserved location attempting to be written with value %{VALUE}. Fields:

OFFSET unsigned int

Offset of address within the MHU.

VALUE unsigned int

Value written.

RECV_MemoryMapped_Read

Trace read from a memory-mapped register. Fields:

OFFSET unsigned int

Offset of address within the MHU.

REG_NAME string

Name of the register being read.

VALUE unsigned int

Value returned.

RECV_MemoryMapped_Write

Trace write to a memory-mapped register. Fields:

OFFSET unsigned int

Offset of address within the MHU.

REG_NAME string

Name of the register being written.

UPDATED_VALUE unsigned int

New value read back from the register; for write-only registers this will be equal to VALUE.

VALUE unsigned int

Value written.

SEND_ArchMsg.Warning.MemoryMapped_AccessAborted

DISPLAY Receiver is asleep, access to message registers BLOCKED, address=%{VALUE}.

Fields:

OFFSET unsigned int

Offset of address within the MHU.

SEND_ArchMsg.Warning.MemoryMapped_ReadReserved

DISPLAY Reserved location attempting to be read. Fields:

OFFSET unsigned int

Offset of address within the MHU.

SEND_ArchMsg.Warning.MemoryMapped_ReadWriteOnlyReg

DISPLAY Write-only register %{REG_NAME} attempting to be read. Fields:

OFFSET unsigned int

Offset of address within the MHU.

REG_NAME string

Name of the register being read.

SEND_ArchMsg.Warning.MemoryMapped_WriteReadOnlyReg

DISPLAY Read-only register %{REG_NAME} attempting to be written with value %{VALUE}. Fields:

OFFSET unsigned int

Offset of address within the MHU.

REG_NAME string

Name of the register being written.

VALUE unsigned int

Value written.

SEND_ArchMsg.Warning.MemoryMapped_WriteReserved

DISPLAY Reserved location attempting to be written with value %{VALUE}. Fields:

OFFSET unsigned int

Offset of address within the MHU.

VALUE unsigned int

Value written.

SEND_MemoryMapped_Read

Trace read from a memory-mapped register. Fields:

OFFSET unsigned int

Offset of address within the MHU.

REG_NAME string

Name of the register being read.

VALUE unsigned int

Value returned.

SEND_MemoryMapped_Write

Trace write to a memory-mapped register. Fields:

OFFSET unsigned int

Offset of address within the MHU.

REG_NAME string

Name of the register being written.

UPDATED_VALUE unsigned int

New value read back from the register; for write-only registers this will be equal to VALUE.

VALUE unsigned int

Value written.

2.144 MessageHandlingUnitV3

This section describes the trace sources.

MHU:InterruptCombinedMailbox

Trace interrupt to combined mailbox. Fields:

LEVEL string

level of interrupt signal.

MHU:InterruptCombinedPostbox

Trace interrupt to combined postbox. Fields:

LEVEL string

level of interrupt signal.

MHU:InterruptReceiverFastChannel

Trace interrupt to receiver fast channel. Fields:

CHANNEL unsigned int

channel of interrupt signal.

LEVEL string

level of interrupt signal.

MHU:InterruptReceiverFastChannelGroup

Trace interrupt to receiver fast group. Fields:

GROUP unsigned int

fast channel group of interrupt signal.

LEVEL string

level of interrupt signal.

MHU_RECV:MemoryMapped_Read

Trace read from a memory-mapped register. Fields:

OFFSET unsigned int

Offset of address within the MHU.

REG_NAME string

Name of the register being read.

VALUE unsigned int

Value returned.

MHU_RECV:MemoryMapped_Write

Trace write to a memory-mapped register. Fields:

OFFSET unsigned int

Offset of address within the MHU.

REG_NAME string

Name of the register being written.

UPDATED_VALUE unsigned int

New value read back from the register; for write-only registers this will be equal to VALUE.

VALUE unsigned int

Value written.

MHU_SEND:MemoryMapped_Read

Trace read from a memory-mapped register. Fields:

OFFSET unsigned int

Offset of address within the MHU.

REG_NAME string

Name of the register being read.

VALUE unsigned int

Value returned.

MHU_SEND:MemoryMapped_Write

Trace write to a memory-mapped register. Fields:

OFFSET unsigned int

Offset of address within the MHU.

REG_NAME string

Name of the register being written.

UPDATED_VALUE unsigned int

New value read back from the register; for write-only registers this will be equal to VALUE.

VALUE unsigned int

Value written.

MailboxRecvDoorbellMessage

Receive Doorbell Message at Mailbox. Fields:

CHANNEL unsigned int

Channel message received on.

VALUE unsigned int

Value received.

MailboxRecvFastChannel32Message

Receive Fast Channel 32 Message at Mailbox. Fields:

CHANNEL unsigned int

Channel message received on.

VALUE unsigned int

Value received.

MailboxRecvFastChannel64Message

Receive Fast Channel 64 Message at Mailbox. Fields:

CHANNEL unsigned int

Channel message received on.

VALUE unsigned int

Value received.

PostboxSendDoorbellMessage

Send Message from Doorbell Postbox. Fields:

CHANNEL unsigned int

Channel message sent on.

VALUE unsigned int

Value written.

PostboxSendFastChannel32Message

Send Message from Fast Channel 32 Postbox. Fields:

CHANNEL unsigned int

Channel message sent on.

VALUE unsigned int

Value written.

PostboxSendFastChannel64Message

Send Message from Fast Channel 64 Postbox. Fields:

CHANNEL unsigned int

Channel message sent on.

VALUE unsigned int

Value written.

2.145 MetaDataController

This section describes the trace sources.

ArchMsg.Error.error_tagged_access_in_no_metadata_region

Tagged data access in a region without metadata. Fields:

PA unsigned int

Physical address the data access is made into.

PAS enum

Physical Address Space.

COMPONENT_BUSY

This component is now in use. The request has to wait until the component is ready . Fields:

Component string

Name of the component in use.

READ_METADATA

Read metadata from end-point storage. Fields:

ACCESS_SIZE_IN_BYTES unsigned int

Size in Bytes.

ADDRESS unsigned int

Address to read.

METADATA unsigned int

Metadata content.

READ_OFFSET_IN_ADDRESS unsigned int

Offset in master's storage.

WRITE_METADATA

Write metadata to end-point storage. Fields:

ACCESS_SIZE_IN_BYTES unsigned int

Size in Bytes.

ADDRESS unsigned int

Address to write.

METADATA unsigned int

Metadata content.

WRITE_OFFSET_IN_ADDRESS unsigned int

Offset in master's storage.

2.146 NI700

This section describes the trace sources.

ArchMsg.Error.MemoryMapped_AddressInvalid

DISPLAY Register access attempted to %{ADDRESS} does not fall into %{REG_SPACE_BASE} - %{REG_SPACE_END}. This range is where registers may be found. Fields:

ADDRESS unsigned int

Address of Access that entered RegisterInterface to the Register Space in the Interconnect.

REG_SPACE_BASE unsigned int

Register Space Base Address.

REG_SPACE_END unsigned int

Register Space End Address.

ArchMsg.Warning.MemoryMapped_ReadDisallowed

DISPLAY Read of %{REG_NAME}{%{Reg_Ownership}} not allowed from PASpace=%{PASpace}; will be read-as-zero. Fields:

OFFSET unsigned int

Offset of address within the Interconnect.

PASpace string

Physical Address Space.

REG_NAME string

Name of the register being read.

Reg_Ownership string

Register Ownership.

ArchMsg.Warning.MemoryMapped_ReadReserved

DISPLAY Reserved offset %{OFFSET} attempting to be read. Fields:

OFFSET unsigned int

Offset of address within the Interconnect.

ArchMsg.Warning.MemoryMapped_ReadWriteOnlyReg

DISPLAY Write-only register %{REG_NAME} attempting to be read; will read-as-zero. Fields:

OFFSET unsigned int

Offset of address within the Interconnect.

REG_NAME string

Name of the register being read.

ArchMsg.Warning.MemoryMapped_UnalignedReadAccess

DISPLAY Read attempted to %{OFFSET} and not aligned to 32 or 64-bit. Fields:

OFFSET unsigned int

Offset of address within the Interconnect.

ArchMsg.Warning.MemoryMapped_UnalignedWriteAccess

DISPLAY Write attempted to %{OFFSET} and not aligned to 32 or 64-bit. Fields:

OFFSET unsigned int

Offset of address within the Interconnect.

ArchMsg.Warning.MemoryMapped_WriteDisallowed

DISPLAY Write to %{REG_NAME}{%{Reg_Ownership}}=%{VALUE} not allowed from PASpace=%{PASpace}; will be ignored. Fields:

OFFSET unsigned int

Offset of address within the Interconnect.

PASpace string

Physical Address Space.

REG_NAME string

Name of the register being read.

Reg_Ownership string

Register Ownership.

VALUE unsigned int

Value.

ArchMsg.Warning.MemoryMapped_WriteReadOnlyReg

DISPLAY Read-only register %{REG_NAME} attempting to be written with value %{VALUE}. Fields:

OFFSET unsigned int

Offset of address within the Interconnect.

REG_NAME string

Name of the register being written.

VALUE unsigned int

Value written.

ArchMsg.Warning.MemoryMapped_WriteReserved

DISPLAY Reserved offset %{OFFSET} attempting to be written with value %{VALUE}. Fields:

OFFSET unsigned int

Offset of address within the Interconnect.

VALUE unsigned int

Value written.

IDM_Access_Control_Transaction_Abort

DISPLAY Access to memory location %{ADDR} being aborted at isolated %{DOMAIN} %{PORT}. Fields:

ADDR unsigned int

Address of memory location being accessed.

DOMAIN enum

The domain of the xxNI isolated by IDM access control.

PORT unsigned int

The index of the xxNI isolated by IDM access control.

IDM_Reset_Transaction_Abort

DISPLAY Access to memory location %{ADDR} being aborted at reset %{DOMAIN} %{PORT}. Fields:

ADDR unsigned int

Address of memory location being accessed.

DOMAIN enum

The domain of the xxNI under IDM reset.

PORT unsigned int

The index of the xxNI under IDM reset.

MPAM_Override

DISPLAY Overriding MPAM_PMG=%{MPAM_PMG} MPAM_PARTID=%{MPAM_PARTID} MPAM_NS=%{MPAM_NS} of access to memory location %{ADDR} from %{DOMAIN} %{PORT}. Fields:

ADDR unsigned int

Address of memory location being accessed.

DOMAIN enum

The source domain of the xSNI whose transaction attributes are being overridden.

MPAM_NS unsigned int

Overriding value for the MPAM_NS transaction attribute.

MPAM_PARTID unsigned int

Overriding value for the MPAM_PARTID transaction attribute.

MPAM_PMG unsigned int

Overriding value for the MPAM_PMG transaction attribute.

PORT unsigned int

The index of the xSNI whose transaction attributes are being overridden.

MemoryMapped_Read

Trace read from a memory-mapped register. Fields:

OFFSET unsigned int

Offset of address within the Interconnect.

PASpace string

Physical Address Space.

REG_NAME string

Name of the register being read.

VALUE unsigned int

Value returned.

MemoryMapped_Write

Trace write to a memory-mapped register. Fields:

OFFSET unsigned int

Offset of address within the Interconnect.

PASpace string

Physical Address Space.

REG_NAME string

Name of the register being written.

UPDATED_VALUE unsigned int

New value read back from the register; for write-only registers this will be equal to VALUE.

VALUE unsigned int

Value written.

Routing

DISPLAY Access to memory location %{ADDR} is routed to port %{PORT} in %{DOMAIN}. Fields:

ADDR unsigned int

Address of memory location being accessed.

DOMAIN enum

Downstream domain to which access is routed.

PORT unsigned int

Downstream index to which access is routed.

Transaction_Abort

DISPLAY The transaction accessing memory location %{ADDR} from %{PORT} in %{DOMAIN} was aborted. Fields:

ADDR unsigned int

Address of memory location being accessed.

DOMAIN enum

Upstream domain where the access is coming from.

PORT unsigned int

Upstream index of the port where the access is coming from.

2.147 NI710AE

This section describes the trace sources.

ArchMsg.Error.MemoryMapped_AddressInvalid

DISPLAY Register access attempted to %{ADDRESS} does not fall into %{REG_SPACE_BASE} - %{REG_SPACE_END}. This range is where registers may be found. Fields:

ADDRESS unsigned int

Address of Access that entered RegisterInterface to the Register Space in the Interconnect.

REG_SPACE_BASE unsigned int

Register Space Base Address.

REG_SPACE_END unsigned int

Register Space End Address.

ArchMsg.Warning.MemoryMapped_ReadDisallowed

DISPLAY Read of %{REG_NAME}{%{Reg_Ownership}} not allowed from PASpace=%{PASpace}; will be read-as-zero. Fields:

OFFSET unsigned int

Offset of address within the Interconnect.

PASpace string

Physical Address Space.

REG_NAME string

Name of the register being read.

Reg_Ownership string

Register Ownership.

ArchMsg.Warning.MemoryMapped_ReadReserved

DISPLAY Reserved offset %{OFFSET} attempting to be read. Fields:

OFFSET unsigned int

Offset of address within the Interconnect.

ArchMsg.Warning.MemoryMapped_ReadWriteOnlyReg

DISPLAY Write-only register %{REG_NAME} attempting to be read; will read-as-zero. Fields:

OFFSET unsigned int

Offset of address within the Interconnect.

REG_NAME string

Name of the register being read.

ArchMsg.Warning.MemoryMapped_UnalignedReadAccess

DISPLAY Read attempted to %{OFFSET} and not aligned to 32 or 64-bit. Fields:

OFFSET unsigned int

Offset of address within the Interconnect.

ArchMsg.Warning.MemoryMapped_UnalignedWriteAccess

DISPLAY Write attempted to %{OFFSET} and not aligned to 32 or 64-bit. Fields:

OFFSET unsigned int

Offset of address within the Interconnect.

ArchMsg.Warning.MemoryMapped_WriteDisallowed

DISPLAY Write to %{REG_NAME}{%{Reg_Ownership}}=%{VALUE} not allowed from PASpace=%{PASpace}; will be ignored. Fields:

OFFSET unsigned int

Offset of address within the Interconnect.

PASpace string

Physical Address Space.

REG_NAME string

Name of the register being read.

Reg_Ownership string

Register Ownership.

VALUE unsigned int

Value.

ArchMsg.Warning.MemoryMapped_WriteReadOnlyReg

DISPLAY Read-only register %{REG_NAME} attempting to be written with value %{VALUE}. Fields:

OFFSET unsigned int

Offset of address within the Interconnect.

REG_NAME string

Name of the register being written.

VALUE unsigned int

Value written.

ArchMsg.Warning.MemoryMapped_WriteReserved

DISPLAY Reserved offset %{OFFSET} attempting to be written with value %{VALUE}. Fields:

OFFSET unsigned int

Offset of address within the Interconnect.

VALUE unsigned int

Value written.

Disallowed xMNI APU txn

DISPLAY Transaction originating from %{xSNI} to address %{addr} targeting %{xMNI} failed due to insufficient xMNI APU permissions. Fields:

addr unsigned int

Address of memory location.

xMNI string

Downstream xMNI.

xSNI string

Upstream xSNI.

Disallowed xSNI APU txn

DISPLAY Transaction originating from %{xSNI} to address %{addr} failed due to insufficient xSNI APU permissions. Fields:

addr unsigned int

Address of memory location.

xSNI string

Upstream xSNI.

FMU Error Packet Dropped

DISPLAY Error Packet Dropped: Node ID 0x%{node_id:x}, Node type 0x%{node_type:x} SMEN 0x%{smen:x}, SM 0x%{errsm:x}. Fields:

errsm unsigned int

Error Safety Mechanism bitmask.

node_id unsigned int

Node identifier.

node_type unsigned int

Node type identifier.

smen unsigned int

Enabled Safety Mechanism bitmask.

IDM_Access_Control_Transaction_Abort

DISPLAY Access to memory location %{ADDR} being aborted at isolated %{DOMAIN} %{PORT}. Fields:

ADDR unsigned int

Address of memory location being accessed.

DOMAIN enum

The domain of the xxNI isolated by IDM access control.

PORT unsigned int

The index of the xxNI isolated by IDM access control.

IDM_Reset_Transaction_Abort

DISPLAY Access to memory location %{ADDR} being aborted at reset %{DOMAIN} %{PORT}. Fields:

ADDR unsigned int

Address of memory location being accessed.

DOMAIN enum

The domain of the xxNI under IDM reset.

PORT unsigned int

The index of the xxNI under IDM reset.

MPAM_Override

DISPLAY Overriding MPAM_PMG=%{MPAM_PMG} MPAM_PARTID=%{MPAM_PARTID} MPAM_NS=%{MPAM_NS} of access to memory location %{ADDR} from %{DOMAIN} %{PORT}. Fields:

ADDR unsigned int

Address of memory location being accessed.

DOMAIN enum

The source domain of the xSNI whose transaction attributes are being overridden.

MPAM_NS unsigned int

Overriding value for the MPAM_NS transaction attribute.

MPAM_PARTID unsigned int

Overriding value for the MPAM_PARTID transaction attribute.

MPAM_PMG unsigned int

Overriding value for the MPAM_PMG transaction attribute.

PORT unsigned int

The index of the xSNI whose transaction attributes are being overridden.

MemoryMapped_Read

Trace read from a memory-mapped register. Fields:

OFFSET unsigned int

Offset of address within the Interconnect.

PASpace string

Physical Address Space.

REG_NAME string

Name of the register being read.

VALUE unsigned int

Value returned.

MemoryMapped_Write

Trace write to a memory-mapped register. Fields:

OFFSET unsigned int

Offset of address within the Interconnect.

PASpace string

Physical Address Space.

REG_NAME string

Name of the register being written.

UPDATED_VALUE unsigned int

New value read back from the register; for write-only registers this will be equal to VALUE.

VALUE unsigned int

Value written.

Routing

DISPLAY Access to memory location %{ADDR} is routed to port %{PORT} in %{DOMAIN}. Fields:

ADDR unsigned int

Address of memory location being accessed.

DOMAIN enum

Downstream domain to which access is routed.

PORT unsigned int

Downstream index to which access is routed.

Transaction_Abort

DISPLAY The transaction accessing memory location %{ADDR} from %{PORT} in %{DOMAIN} was aborted. Fields:

ADDR unsigned int

Address of memory location being accessed.

DOMAIN enum

Upstream domain where the access is coming from.

PORT unsigned int

Upstream index of the port where the access is coming from.

2.148 NOC_S3

This section describes the trace sources.

ArchMsg.Error.MemoryMapped_AddressInvalid

DISPLAY Register access attempted to %{ADDRESS} does not fall into %{REG_SPACE_BASE} - %{REG_SPACE_END}. This range is where registers may be found. Fields:

ADDRESS unsigned int

Address of Access that entered RegisterInterface to the Register Space in the Interconnect.

REG_SPACE_BASE unsigned int

Register Space Base Address.

REG_SPACE_END unsigned int

Register Space End Address.

ArchMsg.Warning.MemoryMapped_ReadDisallowed

DISPLAY Read of %{REG_NAME}{%{Reg_Ownership}} not allowed from PASpace=%{PASpace}; will be read-as-zero. Fields:

OFFSET unsigned int

Offset of address within the Interconnect.

PASpace string

Physical Address Space.

REG_NAME string

Name of the register being read.

Reg_Ownership string

Register Ownership.

ArchMsg.Warning.MemoryMapped_ReadReserved

DISPLAY Reserved offset %{OFFSET} attempting to be read. Fields:

OFFSET unsigned int

Offset of address within the Interconnect.

ArchMsg.Warning.MemoryMapped_ReadWriteOnlyReg

DISPLAY Write-only register %{REG_NAME} attempting to be read; will read-as-zero. Fields:

OFFSET unsigned int

Offset of address within the Interconnect.

REG_NAME string

Name of the register being read.

ArchMsg.Warning.MemoryMapped_UnalignedReadAccess

DISPLAY Read attempted to %{OFFSET} and not aligned to 32 or 64-bit. Fields:

OFFSET unsigned int

Offset of address within the Interconnect.

ArchMsg.Warning.MemoryMapped_UnalignedWriteAccess

DISPLAY Write attempted to %{OFFSET} and not aligned to 32 or 64-bit. Fields:

OFFSET unsigned int

Offset of address within the Interconnect.

ArchMsg.Warning.MemoryMapped_WriteDisallowed

DISPLAY Write to %{REG_NAME}{%{Reg_Ownership}}=%{VALUE} not allowed from PASpace=%{PASpace}; will be ignored. Fields:

OFFSET unsigned int

Offset of address within the Interconnect.

PASpace string

Physical Address Space.

REG_NAME string

Name of the register being read.

Reg_Ownership string

Register Ownership.

VALUE unsigned int

Value.

ArchMsg.Warning.MemoryMapped_WriteReadOnlyReg

DISPLAY Read-only register %{REG_NAME} attempting to be written with value %{VALUE}.

Fields:

OFFSET unsigned int

Offset of address within the Interconnect.

REG_NAME string

Name of the register being written.

VALUE unsigned int

Value written.

ArchMsg.Warning.MemoryMapped_WriteReserved

DISPLAY Reserved offset %{OFFSET} attempting to be written with value %{VALUE}. Fields:

OFFSET unsigned int

Offset of address within the Interconnect.

VALUE unsigned int

Value written.

Default Target Used

DISPLAY is %{ADDR} is being routed to the default target port %{PORT} in %{DOMAIN}.

Fields:

ADDR unsigned int

Address of memory location being accessed.

DOMAIN enum

Downstream domain to which access is routed.

PORT unsigned int

Downstream index to which access is routed.

Disallowed xMNI APU txn

DISPLAY Transaction originating from %{xSNI} to address %{addr} targeting %{xMNI} failed due to insufficient xMNI APU permissions. Fields:

addr unsigned int

Address of memory location.

xMNI string

Downstream xMNI.

xSNI string

Upstream xSNI.

Disallowed xSNI APU txn

DISPLAY Transaction originating from %{xSNI} to address %{addr} failed due to insufficient xSNI APU permissions. Fields:

addr unsigned int

Address of memory location.

xSNI string

Upstream xSNI.

IDM_Access_Control_Transaction_Abort

DISPLAY Access to memory location %{ADDR} being aborted at isolated %{DOMAIN} %{PORT}. Fields:

ADDR unsigned int

Address of memory location being accessed.

DOMAIN enum

The domain of the xxNI isolated by IDM access control.

PORT unsigned int

The index of the xxNI isolated by IDM access control.

IDM_Reset_Transaction_Abort

DISPLAY Access to memory location %{ADDR} being aborted at reset %{DOMAIN} %{PORT}. Fields:

ADDR unsigned int

Address of memory location being accessed.

DOMAIN enum

The domain of the xxNI under IDM reset.

PORT unsigned int

The index of the xxNI under IDM reset.

MPAM_Override

DISPLAY Overriding MPAM_PMG=%{MPAM_PMG} MPAM_PARTID=%{MPAM_PARTID} MPAM_NS=%{MPAM_NS} of access to memory location %{ADDR} from %{DOMAIN} %{PORT}. Fields:

ADDR unsigned int

Address of memory location being accessed.

DOMAIN enum

The source domain of the xSNI whose transaction attributes are being overridden.

MPAM_NS unsigned int

Overriding value for the MPAM_NS transaction attribute.

MPAM_PARTID unsigned int

Overriding value for the MPAM_PARTID transaction attribute.

MPAM_PMG unsigned int

Overriding value for the MPAM_PMG transaction attribute.

PORT unsigned int

The index of the xSNI whose transaction attributes are being overridden.

MemoryMapped_Read

Trace read from a memory-mapped register. Fields:

OFFSET unsigned int

Offset of address within the Interconnect.

PASpace string

Physical Address Space.

REG_NAME string

Name of the register being read.

VALUE unsigned int

Value returned.

MemoryMapped_Write

Trace write to a memory-mapped register. Fields:

OFFSET unsigned int

Offset of address within the Interconnect.

PASpace string

Physical Address Space.

REG_NAME string

Name of the register being written.

UPDATED_VALUE unsigned int

New value read back from the register; for write-only registers this will be equal to VALUE.

VALUE unsigned int

Value written.

Programmable_Address_Map

DISPLAY Programmable Address Map of %{SNI} is %{MMAP}. Fields:

MMAP string

Memory map.

SNI string

Upstream SNI.

Routing

DISPLAY Access to memory location %{ADDR} is routed to port %{PORT} in %{DOMAIN}. Fields:

ADDR unsigned int

Address of memory location being accessed.

DOMAIN enum

Downstream domain to which access is routed.

PORT unsigned int

Downstream index to which access is routed.

Transaction_Abort

DISPLAY The transaction accessing memory location %{ADDR} from %{PORT} in %{DOMAIN} was aborted. Fields:

ADDR unsigned int

Address of memory location being accessed.

DOMAIN enum

Upstream domain where the access is coming from.

PORT unsigned int

Upstream index of the port where the access is coming from.

2.149 PL011_Uart

This section describes the trace sources.

pl011_character_in

A character was received. Fields:

char unsigned int

The character received.

fifo_overflow bool

If true, then the character was lost as the fifo was full.

tick unsigned int

The count of ticks from simulation start that the UART has received.

pl011_character_out

A character was written to the UART to output. Fields:

char unsigned int

The output character.

fifo_overflow bool

If true, then the character was lost as the fifo was full.

tick unsigned int

The count of ticks from simulation start that the UART has received.

pl011_line_buffered_out

The buffered output of character_out. The buffer is flushed when any control character is received. The control characters do not form part of the buffer. The buffer's size is limited to 255 characters and truncation is indicated by an appended '...' . Fields:

buffer string

The line buffer.

tick unsigned int

The count of ticks from simulation start that the UART has received at the point at which it receives the control character that flushes the buffer.

2.150 PL310_L2CC

This section describes the trace sources.

CACHE_READ_HIT

A line read hit occurred. Fields:

addr unsigned int

Address of transaction.

cache_line_index unsigned int

Cache line index.

ns bool

Non-secure.

CACHE_READ_MISS

A line read miss occurred. Fields:

addr unsigned int

Address of transaction.

ns bool

Non-secure.

CACHE_WRITE_HIT

A line write hit occurred. Fields:

addr unsigned int

Address of transaction.

cache_line_index unsigned int

Cache line index.

ns bool

Non-secure.

CACHE_WRITE_MISS

A line write miss occurred. Fields:

addr unsigned int

Address of transaction.

ns bool

Non-secure.

cache_line_eviction

The cache is starting to evict a line to downstream. Fields:

addr unsigned int

Address.

cache_line_index unsigned int

Cache line index.

data unsigned int

Data being evicted.

memory_attributes enum

Memory attributes.

ns bool

Non-secure.

cache_line_refill

The cache refilled a line from downstream. Fields:

addr unsigned int

Address.

cache_line_index unsigned int

Cache line index.

data unsigned int

Data or empty on error.

memory_attributes enum

Memory attributes.

ns bool

Non-secure.

cache_line_refill_start

The cache is starting to refill a line from downstream. Fields:

addr unsigned int

Address.

cache_line_index unsigned int

Cache line index.

memory_attributes enum

Memory attributes.

ns bool

Non-secure.

interrupt_status

The raw interrupt status. Fields:

mask_of_new_bit unsigned int

A mask of which bit was changed.

new_value unsigned int

New interrupt status value.

register_read

A register read occurred. Fields:

data unsigned int

Data read.

name string

The register name.

ns bool

Access is non-secure.

offset unsigned int

Offset of register in file.

ok bool

True if a valid register.

register_write

A register write occurred. Fields:

data unsigned int

Data written.

name string

The register name.

ns bool

Access is non-secure.

offset unsigned int

Offset of register in file.

ok bool

True if ok.

2.151 PPUv0

This section describes the trace sources.

PPU_AccessToUnimplementedRegister

DISPLAY PPU Register %{READ:(WRITE|READ)} to address: %{ADDR} FAILED as the register is unimplemented. Fields:

ADDR unsigned int

Address of the accessed register.

READ bool

Transaction type is read or write.

PPU_ReadToWriteOnlyRegister

DISPLAY PPU Register Read to address: %{ADDR} FAILED as the register is write-only. Fields:

ADDR unsigned int

Address of the accessed register.

PPU_RegRead

DISPLAY PPU Register Offset: %{REG_OFFSET} Value: %{VALUE} RegName: %{REG_NAME}. Fields:

REG_NAME string

The name of the register read.

REG_OFFSET unsigned int

The address of the register read.

VALUE unsigned int

Read Value from the register.

PPU_RegWrite

DISPLAY PPU Register Offset: %{REG_OFFSET} Updated: From value %{VALUE} To value %{UPDATED_VALUE} RegName: %{REG_NAME}. Fields:

REG_NAME string

The name of the register updated.

REG_OFFSET unsigned int

The address of the register updated.

UPDATED_VALUE unsigned int

New Value in the register.

VALUE unsigned int

Old Value in the register.

PPU_WriteToReadOnlyRegister

DISPLAY PPU Register Write to address: %{ADDR} FAILED as the register is read-only.
Fields:

ADDR unsigned int

Address of the accessed register.

2.152 PPUv1

This section describes the trace sources.

PPU_AccessToUnimplementedRegister

DISPLAY PPU Register %{READ:(WRITE|READ)} to address: %{ADDR} FAILED as the register is unimplemented. Fields:

ADDR unsigned int

Address of the accessed register.

READ bool

Transaction type is read or write.

PPU_ReadToWriteOnlyRegister

DISPLAY PPU Register Read to address: %{ADDR} FAILED as the register is write-only.
Fields:

ADDR unsigned int

Address of the accessed register.

PPU_RegRead

DISPLAY PPU Register Offset: %{REG_OFFSET} Value: %{VALUE} RegName: %{REG_NAME}.
Fields:

REG_NAME string

The name of the register read.

REG_OFFSET unsigned int

The address of the register read.

VALUE unsigned int

Read Value from the register.

PPU_RegWrite

DISPLAY PPU Register Offset: %{REG_OFFSET} Updated: From value %{VALUE} To value %{UPDATED_VALUE} RegName: %{REG_NAME}. Fields:

REG_NAME string

The name of the register updated.

REG_OFFSET unsigned int

The address of the register updated.

UPDATED_VALUE unsigned int

New Value in the register.

VALUE unsigned int

Old Value in the register.

PPU_WriteToReadOnlyRegister

DISPLAY PPU Register Write to address: %{ADDR} FAILED as the register is read-only.
Fields:

ADDR unsigned int

Address of the accessed register.

key_registers_value_when_power_state_change

trace the key register value when power state may change. Fields:

PPU_DISR unsigned int

.

PPU_IISR unsigned int

.

PPU_ISR unsigned int

.

PPU_PWPR unsigned int

.

PPU_PWSR unsigned int

.

Source string

.

lock_status

The current PPU OFF lock status is %{STATUS}. Fields:

STATUS enum

The status of PPU OFF lock.

pactive_port_status

Detected PACTIVE port signal change %{PORT_STATUS} with value %{SIG_VALUE:x}. Fields:

PORT_STATUS enum

The status of PPU PACTIVE port.

SIG_VALUE unsigned int

The value of PACTIVE signal.

receive_wakerequest

trace the wakerequest event. Fields:

Wakerequest string

PPU is receiving Wakerequest.

transition_status

The current PPU transition status is %{STATUS}. Fields:

STATUS enum

The status of PPU transition.

wakerequest_failed

The wakerequest failed and PPU_ISR is %{PPU_ISR}, PPU_IISR is %{PPU_IISR}. Fields:

PPU_IISR unsigned int

The value of PPU_IISR when wakerequest is failed.

PPU_ISR unsigned int

The value of PPU_ISR when wakerequest is failed.

2.153 PVBUS2AMBAPV

This section describes the trace sources.

READ_ACCESS

Logger. Fields:

ACCESS_SIZE unsigned int

Log2 of access size (i.e. 0=byte, 1=halfword, 2=word, 3=doubleword ...).

ACE enum

Encodes the ACE operation.

ATTR unsigned int

Transaction Attributes: [12] Instruction fetch [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

DATA unsigned int

The data transferred.

EXTENDED_ID unsigned int

The Extended ID.

LATENCY unsigned int

Time downstream (ticks).

MANAGER_ID unsigned int

The AXI Manager ID.

NUMBER_OF_BEATS unsigned int

The number of data transfers (beats) in this burst.

PADDR unsigned int

Physical address of access.

RESPONSE enum

Whether the transaction was successful, or an error occurred.

USER_FLAGS unsigned int

Core specific additional signals.

READ_ACCESS_START

Logger. Fields:

ACCESS_SIZE unsigned int

Log2 of access size (i.e. 0=byte, 1=halfword, 2=word, 3=doubleword ...).

ACE enum

Encodes the ACE operation.

ATTR unsigned int

Transaction Attributes: [12] Instruction fetch [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

DATA unsigned int

The data transferred.

EXTENDED_ID unsigned int

The Extended ID.

MANAGER_ID unsigned int

The AXI Manager ID.

NUMBER_OF_BEATS unsigned int

The number of data transfers (beats) in this burst.

PADDR unsigned int

Physical address of access.

USER_FLAGS unsigned int

Core specific additional signals.

WRITE_ACCESS

Logger. Fields:

ACCESS_SIZE unsigned int

Log2 of access size (i.e. 0=byte, 1=halfword, 2=word, 3=doubleword ...).

ACE enum

Encodes the ACE operation.

ATTR unsigned int

Transaction Attributes: [12] Instruction fetch [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

DATA unsigned int

The data transferred.

EXTENDED_ID unsigned int

The Extended ID.

LATENCY unsigned int

Time downstream (ticks).

MANAGER_ID unsigned int

The AXI Manager ID.

NUMBER_OF_BEATS unsigned int

The number of data transfers (beats) in this burst.

PADDR unsigned int

Physical address of access.

RESPONSE enum

Whether the transaction was successful, or an error occurred.

USER_FLAGS unsigned int

Core specific additional signals.

WRITE_ACCESS_START

Logger. Fields:

ACCESS_SIZE unsigned int

Log2 of access size (i.e. 0=byte, 1=halfword, 2=word, 3=doubleword ...).

ACE enum

Encodes the ACE operation.

ATTR unsigned int

Transaction Attributes: [12] Instruction fetch [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

DATA unsigned int

The data transferred.

EXTENDED_ID unsigned int

The Extended ID.

MANAGER_ID unsigned int

The AXI Manager ID.

NUMBER_OF_BEATS unsigned int

The number of data transfers (beats) in this burst.

PADDR unsigned int

Physical address of access.

USER_FLAGS unsigned int

Core specific additional signals.

2.154 PVBUS2AMBAPVACE

This section describes the trace sources.

ACE_DVM

ACE dvm trace message. Fields:

MSG string

DVM Message.

ACE_SNOOP

ACE snoop trace data. Fields:

ADDR_END_INCL unsigned int

Physical address of access - end inclusive.

ADDR_START unsigned int

Physical address of access - start.

IS_DATATRANS bool

Is a data transfer?.

IS_ERROR bool

Is an error?.

IS_PASSDIRTY bool

Is pass dirty?.

IS_SHARED bool

Is shared?.

IS_WASUNIQUE bool

Was unique?.

NS unsigned int

Is a non secure transaction?.

PAS enum

Physical Address Space of the transaction.

Snoop enum

Encodes the ACE operation: 0 ReadNoSnoop, 1 ReadOnce, 2 ReadClean, 3 ReadShared, 4 ReadNotSharedDirty, 5 ReadUnique, 6 CleanShared, 7 CleanInvalid, 8 MakeInvalid, 9 CleanUnique, 10 MakeUnique, 11 WriteNoSnoop, 12 PseudoWriteOnce, 13 WriteUnique, 14 WriteLineUnique, 15 WriteBack, 16 WriteClean, 17 Evict, 18 Default, 19 MemoryBarrier, 20 SynchronisationBarrier, 21 NoOperation, 22 DebugRead, 23 DebugWrite.

READ_ACCESS

Logger. Fields:

ACCESS_SIZE unsigned int

Log2 of access size (i.e. 0=byte, 1=halfword, 2=word, 3=doubleword ...).

ACE enum

Encodes the ACE operation.

ATTR unsigned int

Transaction Attributes: [12] Instruction fetch [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

DATA unsigned int

The data transferred.

EXTENDED_ID unsigned int

The Extended ID.

LATENCY unsigned int

Time downstream (ticks).

MANAGER_ID unsigned int

The AXI Manager ID.

NUMBER_OF_BEATS unsigned int

The number of data transfers (beats) in this burst.

PADDR unsigned int

Physical address of access.

RESPONSE enum

Whether the transaction was successful, or an error occurred.

USER_FLAGS unsigned int

Core specific additional signals.

READ_ACCESS_START

Logger. Fields:

ACCESS_SIZE unsigned int

Log2 of access size (i.e. 0=byte, 1=halfword, 2=word, 3=doubleword ...).

ACE enum

Encodes the ACE operation.

ATTR unsigned int

Transaction Attributes: [12] Instruction fetch [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

DATA unsigned int

The data transferred.

EXTENDED_ID unsigned int

The Extended ID.

MANAGER_ID unsigned int

The AXI Manager ID.

NUMBER_OF_BEATS unsigned int

The number of data transfers (beats) in this burst.

PADDR unsigned int

Physical address of access.

USER_FLAGS unsigned int

Core specific additional signals.

WRITE_ACCESS

Logger. Fields:

ACCESS_SIZE unsigned int

Log2 of access size (i.e. 0=byte, 1=halfword, 2=word, 3=doubleword ...).

ACE enum

Encodes the ACE operation.

ATTR unsigned int

Transaction Attributes: [12] Instruction fetch [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

DATA unsigned int

The data transferred.

EXTENDED_ID unsigned int

The Extended ID.

LATENCY unsigned int

Time downstream (ticks).

MANAGER_ID unsigned int

The AXI Manager ID.

NUMBER_OF_BEATS unsigned int

The number of data transfers (beats) in this burst.

PADDR unsigned int

Physical address of access.

RESPONSE enum

Whether the transaction was successful, or an error occurred.

USER_FLAGS unsigned int

Core specific additional signals.

WRITE_ACCESS_START

Logger. Fields:

ACCESS_SIZE unsigned int

Log2 of access size (i.e. 0=byte, 1=halfword, 2=word, 3=doubleword ...).

ACE enum

Encodes the ACE operation.

ATTR unsigned int

Transaction Attributes: [12] Instruction fetch [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

DATA unsigned int

The data transferred.

EXTENDED_ID unsigned int

The Extended ID.

MANAGER_ID unsigned int

The AXI Manager ID.

NUMBER_OF_BEATS unsigned int

The number of data transfers (beats) in this burst.

PADDR unsigned int

Physical address of access.

USER_FLAGS unsigned int

Core specific additional signals.

2.155 PVBusBridge

This section describes the trace sources.

BRIDGE_DMI

DMI information from PVBus to AMBA-PV bridge. Fields:

ADDR unsigned int

Address of the operation.

Access string

R = Read, W = Write, r = Prefetch only read, w = Prefetch only write, I = Invalidate.

Op string

D = Downstream call, N = New range received, I = Invalidate from downstream, M = invalidate from upstream monitor.

R_END unsigned int

Range end.

R_PTR unsigned int

Range pointer.

R_START unsigned int

Range start.

2.156 PVBusExclusiveMonitor

This section describes the trace sources.

GLOBAL_MONITOR

Global monitor activity. Fields:

MANAGER_ID unsigned int

Indicates the manager associated with this monitor.

NS enum

Secure state of the region monitored.

PADDR unsigned int

Physical address of the region monitored.

SIZE unsigned int

Size of the region monitored.

STATE enum

State of the monitor (Open/Exclusive).

TRIGGER enum

Reason the monitor changed state.

WIDTH unsigned int

Width in bytes of the access which created the exclusive region.

MONITOR_TRANSACTION

Transactions observed by the monitor. Fields:

DATA unsigned int

The data transferred by the transaction.

MANAGER_ID unsigned int

Identifies the manager of this transaction.

NS enum

Secure state of the transaction.

PADDR unsigned int

Physical address of the transaction.

RESPONSE enum

The transaction result.

TYPE enum

Type of transaction.

WIDTH unsigned int

Width in bytes of the access.

2.157 PVBusGICv3Comms

This section describes the trace sources.

ArchMsg.Warning.GICv3CommsPVBus_PacketTooBig

DISPLAY Received incoming PVBus packet that is too large to be sent over GICv3Comms connection (length = %{LENGTH}, data[0] (packet type) = 0x%{DATA_0:x}, index/manager ID = %{INDEX}). Fields:

DATA_0 unsigned int

First byte of the packet (normally data type).

INDEX unsigned int

Index of GICv3Comms connection that the message should have been sent to (based on AXI manager ID).

LENGTH unsigned int

Number bytes in data packet.

ArchMsg.Warning.GICv3CommsPVBus_UnknownDestination

DISPLAY Received incoming PVBus packet that indicates that it should be sent to an unconnected destination (index/manager ID = %{INDEX}, length = %{LENGTH}, data[0] (packet type) = 0x%{DATA_0:x}). Fields:

DATA unsigned int

The message data.

DATA_0 unsigned int

First byte of the packet (normally data type).

INDEX unsigned int

Index of GICv3Comms connection that the message should have been sent to (based on AXI manager ID).

LENGTH unsigned int

Number bytes in data packet.

GICv3CommsPVBUS_FailedMessage

Processed message from connected GICv3Comms port and sent out, but PVBUS transaction failed. Fields:

DATA unsigned int

The message data.

INDEX unsigned int

Index of GICv3Comms connection that message came in on.

LENGTH unsigned int

Message length in bytes.

GICv3CommsPVBUS_ProcessedMessage

Processed message from connected GICv3Comms port. Fields:

DATA unsigned int

The message data.

INDEX unsigned int

Index of GICv3Comms connection that message came in on.

LENGTH unsigned int

Message length in bytes.

GICv3CommsPVBUS_ReceiveAXIManagerID

DISPLAY Received AXI manager ID %{MANAGER_ID:x} for connection %{INDEX}. Fields:

INDEX unsigned int

Index of GICv3Comms connection that message came in on.

MANAGER_ID unsigned int

Manager ID value recieved.

GICv3CommsPVBUS_SentViaGICv3Comms

Packet received via PVBUS and sent via GICv3 Comms to connected distributor/CPU. Fields:

DATA unsigned int

The message data.

INDEX unsigned int

Index of GICv3Comms connection that the message should have been sent to (based on AXI manager ID).

LENGTH unsigned int

Number bytes in data packet.

2.158 PVBusLogger

This section describes the trace sources.

READ_ACCESS

Trace access at this point in the PVBus hierarchy. Fields:

ACCESS_SIZE unsigned int

Log2 of access size (i.e. 0=byte, 1=halfword, 2=word, 3=doubleword ...).

ACE enum

Encodes the ACE operation.

ATTR unsigned int

Transaction Attributes: [12] Instruction fetch [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

DATA unsigned int

The data transferred.

EXTENDED_ID unsigned int

The Extended ID.

LATENCY unsigned int

Time downstream (ticks).

MANAGER_ID unsigned int

The AXI Manager ID.

NUMBER_OF_BEATS unsigned int

The number of data transfers (beats) in this burst.

PADDR unsigned int

Physical address of access.

RESPONSE enum

Whether the transaction was successful, or an error occurred.

USER_FLAGS unsigned int

Core specific additional signals.

READ_ACCESS_START

Trace access at this point in the PVBUS hierarchy. Fields:

ACCESS_SIZE unsigned int

Log2 of access size (i.e. 0=byte, 1=halfword, 2=word, 3=doubleword ...).

ACE enum

Encodes the ACE operation.

ATTR unsigned int

Transaction Attributes: [12] Instruction fetch [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

DATA unsigned int

The data transferred.

EXTENDED_ID unsigned int

The Extended ID.

MANAGER_ID unsigned int

The AXI Manager ID.

NUMBER_OF_BEATS unsigned int

The number of data transfers (beats) in this burst.

PADDR unsigned int

Physical address of access.

USER_FLAGS unsigned int

Core specific additional signals.

WRITE_ACCESS

Trace access at this point in the PVBUS hierarchy. Fields:

ACCESS_SIZE unsigned int

Log2 of access size (i.e. 0=byte, 1=halfword, 2=word, 3=doubleword ...).

ACE enum

Encodes the ACE operation.

ATTR unsigned int

Transaction Attributes: [12] Instruction fetch [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

DATA unsigned int

The data transferred.

EXTENDED_ID unsigned int

The Extended ID.

LATENCY unsigned int

Time downstream (ticks).

MANAGER_ID unsigned int

The AXI Manager ID.

NUMBER_OF_BEATS unsigned int

The number of data transfers (beats) in this burst.

PADDR unsigned int

Physical address of access.

RESPONSE enum

Whether the transaction was successful, or an error occurred.

USER_FLAGS unsigned int

Core specific additional signals.

WRITE_ACCESS_START

Trace access at this point in the PVBUS hierarchy. Fields:

ACCESS_SIZE unsigned int

Log2 of access size (i.e. 0=byte, 1=halfword, 2=word, 3=doubleword ...).

ACE enum

Encodes the ACE operation.

ATTR unsigned int

Transaction Attributes: [12] Instruction fetch [11] Non-secure, [10] Privileged, [9:8] shareability domain (0=nsh, 1=ish, 2=osh, 3=system), [7:4] outer memory attributes, [3:0] inner memory attributes ([7]/[3] Allocate (Other Allocate when read), [6]/[2] Allocate (Other Allocate when write), [5]/[1] Modifiable, [4]/[0] Bufferable). All other bits are implementation defined.

DATA unsigned int

The data transferred.

EXTENDED_ID unsigned int

The Extended ID.

MANAGER_ID unsigned int

The AXI Manager ID.

NUMBER_OF_BEATS unsigned int

The number of data transfers (beats) in this burst.

PADDR unsigned int

Physical address of access.

USER_FLAGS unsigned int

Core specific additional signals.

2.159 PVBusMapper

This section describes the trace sources.

ArchMsg.Error.BusActiveDuringReset

Transactions recieved at the bus slave port whilst reset was asserted.

ArchMsg.Warning.BusDeadlockDetected

A potential bus deadlock has been detected. Fields:

INTERCONNECT_RESET bool

Downstream bus interconnect reset.

INTERMEDIARY_RESET bool

Bus mapper intermediary reset.

SLAVE_RESET bool

Downstream bus slave reset.

2.160 PVBusMaster

This section describes the trace sources.

ArchMsg.Warning.BusDeadlockDetected

A potential bus deadlock has been detected. Fields:

INTERCONNECT_RESET bool

Bus interconnect reset.

MASTER_RESET bool

Bus master reset.

SLAVE_RESET bool

Bus slave reset.

2.161 PVBusSlave

This section describes the trace sources.

ATOMIC_SLAVE_ACCESS

Information on the atomic operation at slave side. Fields:

ACCESS_RESULT enum

Result of atomic access.

COMPARE_VALUE unsigned int

Value to be compared with memory by CAS.

LOAD_VALUE unsigned int

Value to be loaded from memory as a result of atomic operation.

MANAGER unsigned int

ID of manager which initiated the transaction.

NS bool

Transaction is Non-Secure.

OPERAND_VALUE unsigned int

Incoming operand value provided by core.

OPERATION unsigned int

Type of atomic operation.

PADDR unsigned int

Physical address.

STORE_VALUE unsigned int

Value to be stored to memory as a result of atomic operation.

ArchMsg.Warning.uninitialized_data_read

Unwritten data read warning for ACCESSMODE_TRACKED_MEMORY. DISPLAY At %{PADDR:x}, %{WIDTH:d}bit read contains uninitialized bytes in position %{MASK:x} Uninitialized bytes in position %{MASK_COMPLETE:x}. Fields:

MASK unsigned int

Set bits indicate uninitialized data in bits [63:0].

MASK2 unsigned int

Set bits indicate uninitialized data in bits [127:64].

MASK_COMPLETE unsigned int

Set bits indicate uninitialized data in bits [(WIDTH - 1):0].

PADDR unsigned int

Physical address.

WIDTH unsigned int

Bit width of access.

ArchMsg.Warning.unmonitored_exclusive_access

Exclusive access to address not covered by an exclusive monitor. DISPLAY Exclusive %{NS:(|non-}secure %{DIRECTION:(write to|read from)} physical address %{PADDR:x} not covered by an exclusive monitor. Fields:

DIRECTION enum

Access direction (read/write).

NS bool

Transaction is Non-Secure.

PADDR unsigned int

Physical address.

DMI_BUS_SLAVE

Information on DMI memory operations at slave side. Fields:

ADDRESS unsigned int

Address of DMI operation.

DMI_OPERATION enum

DMI Operation.

SIZE unsigned int

Size of DMI operation.

STORAGE unsigned int

Storage pointer.

2.162 PVCache

This section describes the trace sources.

ALLOC_LINEFILL

The system allocated a complete line fill into the RAMs. This happens when the cache has to read the data for a line fill from potentially upstream or downstream. Fields:

ENTRY_INDEX unsigned int

The entry index the line has been loaded into.

MANAGER_ID unsigned int

The manager ID for the associated transaction.

MEMORY_ATTRIBUTES unsigned int

The memory attributes used for the allocation. bits[3:0] are the inner ACACHE, bits[7:4] are the outer ACACHE, bits[9:8] are ADOMAIN, bits[12:10] are APROT, the rest are undocumented.

NS_ADDR unsigned int

The security world and address of the first byte of the line, the MSB encodes the NS bit.

PAS enum

The Physical Address Space.

ALLOC_WRITE

The system allocated a complete line write into the RAMs. This happens when the cache receives a complete cache line write. It need not read from anywhere. Fields:

ENTRY_INDEX unsigned int

The entry index the line has been written into.

MEMORY_ATTRIBUTES unsigned int

The memory attributes used for the allocation. bits[3:0] are the inner ACACHE, bits[7:4] are the outer ACACHE, bits[9:8] are ADOMAIN, bits[12:10] are APROT, the rest are undocumented.

NS_ADDR unsigned int

The security world and address of the first byte of the line, the MSB encodes the NS bit.

PAS enum

The Physical Address Space.

ATOMIC_SLAVE_ACCESS

Information on the atomic operation at slave side. Fields:

ACCESS_RESULT enum

Result of atomic access.

COMPARE_VALUE unsigned int

Value to be compared with memory by CAS.

LOAD_VALUE unsigned int

Value to be loaded from memory as a result of atomic operation.

MANAGER unsigned int

ID of manager which initiated the transaction.

NS bool

Transaction is Non-Secure.

OPERAND_VALUE unsigned int

Incoming operand value provided by core.

OPERATION enum

Type of atomic operation.

PADDR unsigned int

Physical address.

PAS enum

The Physical Address Space.

STORE_VALUE unsigned int

Value to be stored to memory as a result of atomic operation.

ArchMsg.Error.cached_attributes_mismatch

Attributes differ between page and line in cache from same page. DISPLAY Mismatch in attributes for page including address `%{NS_ADDR[63]:(s|ns)}-%{NS_ADDR[62:0]:x} %{LINES:d}` cache lines are allocated with attributes: inner-`%{LINEATTR[3:0]:(device-|device-|noncacheable-non-bufferable|noncacheable|wRmb|wRmB|WT-cacheable|WB-cacheable|Rwmb|RwmB|WT-cacheable|WB-cacheable|RWmb|RWmB|WT-cacheable|WB-cacheable)}%{LINEATTR[3:0.14:13]:(nGnRnE|nGRnE?|GnRnE?|GRnE?|nGnRE|nGRE|GnRE?|GRE|)}%{LINEATTR[15]:(|-transient)}` outer-`%{LINEATTR[7:4]:(device-|`

device-|noncacheable-non-bufferable|noncacheable|wRmb|wRmB|WT-cacheable|WB-cacheable|Rwmb|RwmB|WT-cacheable|WB-cacheable|RWmb|RWmB|WT-cacheable|WB-cacheable)%{LINEATTR[7:4.14:13]:(nGnRnE|nGRnE?|GnRnE?|GRnE?|nGnRE|nGRE|GnRE?|GRE))}%{LINEATTR[16]:(|-transient)} shareability: %{LINEATTR[9:8]:(nsh|ish|osh|sys)}%{LINEATTR[21.24]:(|TranslatedAccess|NonStallable|TransFaultFlow)}%{LINEATTR[22]:(|NonAddressBasedRouted)}%{LINEATTR[25]:(|HasMetaData)}%{LINEATTR[26]:(|ReservedImpDef)}%{LINEATTR[27]:(|-StreamTransaction)}%{LINEATTR[28]:(|-nse)}%{LINEATTR[29]:(|-nse2)}%{LINEATTR[31]:(|-HasMutatingMetaDataOp)} but transaction attributes are: inner-%{TXATTR[3:0]:(device-|device-|noncacheable-non-bufferable|noncacheable|wRmb|wRmB|WT-cacheable|WB-cacheable|Rwmb|RwmB|WT-cacheable|WB-cacheable|RWmb|RWmB|WT-cacheable|WB-cacheable)}%{TXATTR[3:0.14:13]:(nGnRnE|nGRnE?|GnRnE?|GRnE?|nGnRE|nGRE|GnRE?|GRE))}%{TXATTR[15]:(|-transient)} outer-%{TXATTR[7:4]:(device-|device-|noncacheable-non-bufferable|noncacheable|wRmb|wRmB|WT-cacheable|WB-cacheable|Rwmb|RwmB|WT-cacheable|WB-cacheable|RWmb|RWmB|WT-cacheable|WB-cacheable)}%{TXATTR[7:4.14:13]:(nGnRnE|nGRnE?|GnRnE?|GRnE?|nGnRE|nGRE|GnRE?|GRE))}%{TXATTR[16]:(|-transient)} shareability: %{TXATTR[9:8]:(nsh|ish|osh|sys)}%{TXATTR[21.24]:(|TranslatedAccess|NonStallable|TransFaultFlow)}%{TXATTR[22]:(|NonAddressBasedRouted)}%{TXATTR[25]:(|HasMetaData)}%{TXATTR[26]:(|ReservedImpDef)}%{TXATTR[27]:(|-StreamTransaction)}%{TXATTR[28]:(|-nse)}%{TXATTR[29]:(|-nse2)}%{TXATTR[31]:(|-HasMutatingMetaDataOp)}. PRIMARY KEY NS_ADDR. Fields:

LINEATTR unsigned int

Attributes of the cache allocated lines.

LINES unsigned int

Number of cache lines already allocated.

NS_ADDR unsigned int

The security world and address of the first byte of the page, the MSB encodes the NS bit.

PAS enum

The Physical Address Space.

TXATTR unsigned int

Attributes of the transaction.

ArchMsg.Error.cached_attributes_mismatch#lineinfo

Information about allocated lines in page. DISPLAY %{TAG[1]:(u|U)}%{TAG[2]:(d|D)}-%{TAG[63:61]:(s|rt|ns|rl|sa|na6|nsp|na7)}-%{PADDR}-%{TAG[59:58]:(nsh|ish|osh|sys)}%{TAG[56]:(|-iHittable)}%{TAG[57]:(|-oHittable)}. Fields:

PADDR unsigned int

address of this cache line.

TAG unsigned int

tag data of this cache line.

ArchMsg.Warning.ns_s_dirty_hit

Cache hit at S and NS versions of the same physical address DISPLAY At PAddr %{PADDR} Line index %{INDEX0} is for %{TAG0[1]:(u|U)}%{TAG0[2]:(d|D)}-%{TAG0[63:61]:(s|

rt|ns|rl|sa|na6|nsp|na7}}-%{PADDR}-%{TAG0[59:58]:(nsh|ish|osh|sys}}%{TAG0[56]:(|-iHittable}}%{TAG0[57]:(|-oHittable}} Line index %{|INDEX1} is for %{|TAG1[1]:(u|U}}%{TAG1[2]:(d|D}}-%{TAG1[63:61]:(s|rt|ns|rl|sa|na6|nsp|na7}}-%{PADDR}-%{TAG1[59:58]:(nsh|ish|osh|sys}}%{TAG1[56]:(|-iHittable}}%{TAG1[57]:(|-oHittable}} i.e. same address but different security regimes and one or more are dirty This is not an error if different memory is backing the lines, but otherwise can cause unpredictable cache incoherency. PRIMARY KEY PADDR. Fields:

INDEX0 unsigned int

The cache line index of the affected cache line.

INDEX1 unsigned int

The cache line index of the other regime cache line.

PADDR unsigned int

The address of the affected cache line.

TAG0 unsigned int

The tag data of the affected cache line.

TAG1 unsigned int

The tag data of the other regime cache line.

CACHE_READ_HIT

Read access cache hit. Note that this trace source causes a large slowdown in the simulation.

Fields:

IS_PRELOAD enum

Is the access a preload.

IS_SHARED enum

Is the access shared.

LATENCY unsigned int

Cache latency in ticks per access.

MANAGER_ID unsigned int

Indicates the manager ID associated with this monitor.

CACHE_READ_MISS

Read access cache miss and cache miss latency. Fields:

IS_PRELOAD enum

Is the access a preload.

IS_SHARED enum

Is the access shared.

LATENCY unsigned int

Cache miss latency in ticks per access.

MANAGER_ID unsigned int

Indicates the manager ID associated with this monitor.

CACHE_WRITE_HIT

Write access cache hit. Note that this trace source causes a large slowdown in the simulation. Fields:

IS_PRELOAD enum

Is the access a preload.

IS_SHARED enum

Is the access shared.

LATENCY unsigned int

Cache latency in ticks per access.

MANAGER_ID unsigned int

Indicates the manager ID associated with this monitor.

CACHE_WRITE_MISS

Write access cache miss and cache miss latency. Fields:

IS_PRELOAD enum

Is the access a preload.

IS_SHARED enum

Is the access shared.

LATENCY unsigned int

Cache miss latency in ticks per access.

MANAGER_ID unsigned int

Indicates the manager ID associated with this monitor.

DVM_Message

DISPLAY DVM_msg (%{DVM_MESSAGE}). Fields:

DVM_MESSAGE string

The DVM Message.

ENTRY_BECOMES_INVALID

An entry is now invalid. The cause could be any reason. Fields:

ENTRY_INDEX unsigned int

The entry index that has become invalid.

ENTRY_SET_DIRTY

Entry change, clean/dirty. Fields:

ENTRY_INDEX unsigned int

The entry index that has become clean or dirty.

STATUS unsigned int

True if the line was made dirty.

ERROR_MIXED_ATTRIBUTES_LINE

Mismatched line and transaction attributes. Fields:

ENTRY_INDEX unsigned int

The entry index that has collided.

NS_ADDR unsigned int

The security world and address of the first byte of the line, the MSB encodes the NS bit.

PAS enum

The Physical Address Space.

ERROR_MIXED_ATTRIBUTES_PAGE

Attributes differ between page and lines in cache from the same page. Fields:

NS_ADDR unsigned int

The security world and address of the first byte of the page, the MSB encodes the NS bit.

PAS enum

The Physical Address Space.

EVICTION

A line was evicted from the cache. Fields:

ADDRESS unsigned int

Base address of the victim cache line.

CLEAN bool

If true the cache retained the cache line after writing the dirty data.

ENTRY_INDEX unsigned int

The entry index of the victim.

NON_SECURE bool

If true the victim was NonSecure.

PAS enum

Physical Address Space.

IMPRECISE_ABORT

An operation produced an imprecise abort. Usually this is a WriteBack/WriteClean operation receiving an error response. Fields:

ADDRESS unsigned int

Address of the transaction that imprecise aborted.

DECERR bool

True if this is a DECERR response.

DOWNSTREAM_PORT_INDEX unsigned int

The downstream port index that received the imprecise abort.

NS bool

The NS state of the transaction (true is non-secure).

PAS enum

The Physical Address Space.

MAINTENANCE_CLEAN_SET_WAY

A cache maintenance by set-way operation occurred, possibly the 'clean' part of a clean-and-invalidate operation. Fields:

ENTRY_INDEX unsigned int

The entry index the set-way operation has mapped to.

IS_NON_SECURE bool

If the invalidate operation is non-secure (and so only affects non-secure lines), or secure (and so affects both).

PAS enum

Physical Address Space.

STATE enum

Line transformation incurred.

MAINTENANCE_INVALIDATE_ALL

A cache maintenance invalidate-all operation occurred. Fields:

IS_NON_SECURE bool

True if the invalidate operation is non-secure or realm, otherwise false: * Non-secure invalidates only non-secure lines * Secure invalidates secure and non-secure lines * Realm invalidates realm and non-secure lines * Root invalidates lines from any world.

PAS enum

The Physical Address Space.

MAINTENANCE_INV_SET_WAY

A cache maintenance by set-way operation occurred, possibly the 'invalidate' part of a clean-and-invalidate operation. Fields:

ENTRY_INDEX unsigned int

The entry index the set-way operation has mapped to.

IS_NON_SECURE bool

If the invalidate operation is non-secure (and so only affects non-secure lines), or secure (and so affects both).

PAS enum

The Physical Address Space.

STATE enum

Line transformation incurred.

change_dvm_disabled_ports

This reports a bitmap of which ports are disabled to receive DVM messages. Fields:

new_bitmap unsigned int

The new bitmap of disabled ports.

new_bitmap_list unsigned int

A list of port numbers that were disabled.

old_bitmap unsigned int

The old bitmap of disabled ports.

old_bitmap_list unsigned int

A list of port numbers that were disabled.

change_upstream_disabled_ports

This reports a bitmap of which upstream ports of the cache are *disabled* and so no snoop requests are sent to them. Fields:

new_bitmap unsigned int

The bitmap of the first 32 upstream ports that are disabled.

new_bitmap_list unsigned int

A list of port numbers that were disabled.

old_bitmap unsigned int

The bitmap of the first 32 upstream ports that are disabled.

old_bitmap_list unsigned int

A list of port numbers that were disabled.

entry_after_far_atomic

A cache line has had a far atomic applied to it. Fields:

atomic_operation enum

The operation that operated on this line.

be bool

The far atomic's big-endian flag.

data unsigned int

The data in the cache line.

entry_begin_address unsigned int

The start address of the entry.

entry_index unsigned int

The entry index in the cache.

incoming_data unsigned int

The incoming data from the far atomic. For CAS operations then this is compare-value and then the replacement-value.

manager_id unsigned int

The manager ID for the associated transaction.

ns enum

The security state of the line.

pas enum

The Physical Address Space.

prior_data unsigned int

The data that will be returned as the 'prior' data of the far atomic.

trans_begin_address unsigned int

The start address of the transaction.

entry_after_read

A cache line has been read. Fields:

ace_operation enum

The ACE operation that operated on this line.

data unsigned int

The data in the cache line.

dirty bool

Is the line ACE-dirty.

entry_begin_address unsigned int

The start address of the entry.

entry_index unsigned int

The entry index in the cache.

ns enum

The security state of the line.

pas enum

The Physical Address Space.

snoop bool

This transaction is a snoop transaction.

trans_begin_address unsigned int

The start address of the transaction.

unique bool

Is the line ACE-unique.

entry_after_write

A cache line has been written. Fields:

ace_operation enum

The ACE operation that operated on this line.

data unsigned int

The data in the cache line.

entry_begin_address unsigned int

The start address of the entry.

entry_index unsigned int

The entry index in the cache.

ns enum

The security state of the line.

pas enum

The Physical Address Space.

trans_begin_address unsigned int

The start address of the transaction.

2.163 PVCache64

This section describes the trace sources.

ALLOC_LINEFILL

The system allocated a complete line fill into the RAMs. This happens when the cache has to read the data for a line fill from potentially upstream or downstream. Fields:

ENTRY_INDEX unsigned int

The entry index the line has been loaded into.

MANAGER_ID unsigned int

The manager ID for the associated transaction.

MEMORY_ATTRIBUTES unsigned int

The memory attributes used for the allocation. bits[3:0] are the inner ACACHE, bits[7:4] are the outer ACACHE, bits[9:8] are ADOMAIN, bits[12:10] are APROT, the rest are undocumented.

NS_ADDR unsigned int

The security world and address of the first byte of the line, the MSB encodes the NS bit.

PAS enum

The Physical Address Space.

ALLOC_WRITE

The system allocated a complete line write into the RAMs. This happens when the cache receives a complete cache line write. It need not read from anywhere. Fields:

ENTRY_INDEX unsigned int

The entry index the line has been written into.

MEMORY_ATTRIBUTES unsigned int

The memory attributes used for the allocation. bits[3:0] are the inner ACACHE, bits[7:4] are the outer ACACHE, bits[9:8] are ADOMAIN, bits[12:10] are APROT, the rest are undocumented.

NS_ADDR unsigned int

The security world and address of the first byte of the line, the MSB encodes the NS bit.

PAS enum

The Physical Address Space.

ATOMIC_SLAVE_ACCESS

Information on the atomic operation at slave side. Fields:

ACCESS_RESULT enum

Result of atomic access.

COMPARE_VALUE unsigned int

Value to be compared with memory by CAS.

LOAD_VALUE unsigned int

Value to be loaded from memory as a result of atomic operation.

MANAGER unsigned int

ID of manager which initiated the transaction.

NS bool

Transaction is Non-Secure.

OPERAND_VALUE unsigned int

Incoming operand value provided by core.

OPERATION enum

Type of atomic operation.

PADDR unsigned int

Physical address.

PAS enum

The Physical Address Space.

STORE_VALUE unsigned int

Value to be stored to memory as a result of atomic operation.

ArchMsg.Error.cached_attributes_mismatch

Attributes differ between page and line in cache from same page. DISPLAY Mismatch in attributes for page including address `%{NS_ADDR[63]:(s|ns)}-%{NS_ADDR[62:0]:x} %{LINES:d}` cache lines are allocated with attributes: inner-`%{LINEATTR[3:0]:(device-|device-|noncacheable-non-bufferable|noncacheable|wRmb|wRmB|WT-cacheable|WB-cacheable|Rwmb|RwmB|WT-cacheable|WB-cacheable|RWmb|RWmB|WT-cacheable|WB-cacheable)}%{LINEATTR[3:0.14:13]:(nGnRnE|nGRnE?|GnRnE?|GRnE?|nGnRE|nGRE|GnRE?|GRE|)}%{LINEATTR[15]:(|-transient)}` outer-`%{LINEATTR[7:4]:(device-|`

device-|noncacheable-non-bufferable|noncacheable|wRmb|wRmB|WT-cacheable|WB-cacheable|Rwmb|RwmB|WT-cacheable|WB-cacheable|RWmb|RWmB|WT-cacheable|WB-cacheable)%{LINEATTR[7:4.14:13]:(nGnRnE|nGRnE?|GnRnE?|GRnE?|nGnRE|nGRE|GnRE?|GRE|)%{LINEATTR[16]:(|-transient)} shareability: %{LINEATTR[9:8]:(nsh|ish|osh|sys)}%{LINEATTR[21.24]:(|TranslatedAccess|NonStallable|TransFaultFlow)}%{LINEATTR[22]:(|NonAddressBasedRouted)}%{LINEATTR[25]:(|HasMetaData)}%{LINEATTR[26]:(|ReservedImpDef)}%{LINEATTR[27]:(|-StreamTransaction)}%{LINEATTR[28]:(|-nse)}%{LINEATTR[29]:(|-nse2)}%{LINEATTR[31]:(|-HasMutatingMetaDataOp)} but transaction attributes are: inner-%{TXATTR[3:0]:(device-|device-|noncacheable-non-bufferable|noncacheable|wRmb|wRmB|WT-cacheable|WB-cacheable|Rwmb|RwmB|WT-cacheable|WB-cacheable|RWmb|RWmB|WT-cacheable|WB-cacheable)}%{TXATTR[3:0.14:13]:(nGnRnE|nGRnE?|GnRnE?|GRnE?|nGnRE|nGRE|GnRE?|GRE|)%{TXATTR[15]:(|-transient)} outer-%{TXATTR[7:4]:(device-|device-|noncacheable-non-bufferable|noncacheable|wRmb|wRmB|WT-cacheable|WB-cacheable|Rwmb|RwmB|WT-cacheable|WB-cacheable|RWmb|RWmB|WT-cacheable|WB-cacheable)}%{TXATTR[7:4.14:13]:(nGnRnE|nGRnE?|GnRnE?|GRnE?|nGnRE|nGRE|GnRE?|GRE|)%{TXATTR[16]:(|-transient)} shareability: %{TXATTR[9:8]:(nsh|ish|osh|sys)}%{TXATTR[21.24]:(|TranslatedAccess|NonStallable|TransFaultFlow)}%{TXATTR[22]:(|NonAddressBasedRouted)}%{TXATTR[25]:(|HasMetaData)}%{TXATTR[26]:(|ReservedImpDef)}%{TXATTR[27]:(|-StreamTransaction)}%{TXATTR[28]:(|-nse)}%{TXATTR[29]:(|-nse2)}%{TXATTR[31]:(|-HasMutatingMetaDataOp)}. PRIMARY KEY NS_ADDR. Fields:

LINEATTR unsigned int

Attributes of the cache allocated lines.

LINES unsigned int

Number of cache lines already allocated.

NS_ADDR unsigned int

The security world and address of the first byte of the page, the MSB encodes the NS bit.

PAS enum

The Physical Address Space.

TXATTR unsigned int

Attributes of the transaction.

ArchMsg.Error.cached_attributes_mismatch#lineinfo

Information about allocated lines in page. DISPLAY %{TAG[1]:(u|U)}%{TAG[2]:(d|D)}-%{TAG[63:61]:(s|rt|ns|rl|sa|na6|nsp|na7)}-%{PADDR}-%{TAG[59:58]:(nsh|ish|osh|sys)}%{TAG[56]:(|-iHittable)}%{TAG[57]:(|-oHittable)}. Fields:

PADDR unsigned int

address of this cache line.

TAG unsigned int

tag data of this cache line.

ArchMsg.Warning.ns_s_dirty_hit

Cache hit at S and NS versions of the same physical address DISPLAY At PAddr %{PADDR} Line index %{INDEX0} is for %{TAG0[1]:(u|U)}%{TAG0[2]:(d|D)}-%{TAG0[63:61]:(s|

rt|ns|rl|sa|na6|nsp|na7}}-%{PADDR}-%{TAG0[59:58]:(nsh|ish|osh|sys}}%{TAG0[56]:(|-iHittable}}%{TAG0[57]:(|-oHittable}} Line index %{|INDEX1} is for %{|TAG1[1]:(u|U)}%{|TAG1[2]:(d|D)}-%{TAG1[63:61]:(s|rt|ns|rl|sa|na6|nsp|na7}}-%{PADDR}-%{TAG1[59:58]:(nsh|ish|osh|sys}}%{TAG1[56]:(|-iHittable}}%{TAG1[57]:(|-oHittable}} i.e. same address but different security regimes and one or more are dirty This is not an error if different memory is backing the lines, but otherwise can cause unpredictable cache incoherency. PRIMARY KEY PADDR. Fields:

INDEX0 unsigned int

The cache line index of the affected cache line.

INDEX1 unsigned int

The cache line index of the other regime cache line.

PADDR unsigned int

The address of the affected cache line.

TAG0 unsigned int

The tag data of the affected cache line.

TAG1 unsigned int

The tag data of the other regime cache line.

CACHE_READ_HIT

Read access cache hit. Note that this trace source causes a large slowdown in the simulation. Fields:

IS_PRELOAD enum

Is the access a preload.

IS_SHARED enum

Is the access shared.

LATENCY unsigned int

Cache latency in ticks per access.

MANAGER_ID unsigned int

Indicates the manager ID associated with this monitor.

CACHE_READ_MISS

Read access cache miss and cache miss latency. Fields:

IS_PRELOAD enum

Is the access a preload.

IS_SHARED enum

Is the access shared.

LATENCY unsigned int

Cache miss latency in ticks per access.

MANAGER_ID unsigned int

Indicates the manager ID associated with this monitor.

CACHE_WRITE_HIT

Write access cache hit. Note that this trace source causes a large slowdown in the simulation. Fields:

IS_PRELOAD enum

Is the access a preload.

IS_SHARED enum

Is the access shared.

LATENCY unsigned int

Cache latency in ticks per access.

MANAGER_ID unsigned int

Indicates the manager ID associated with this monitor.

CACHE_WRITE_MISS

Write access cache miss and cache miss latency. Fields:

IS_PRELOAD enum

Is the access a preload.

IS_SHARED enum

Is the access shared.

LATENCY unsigned int

Cache miss latency in ticks per access.

MANAGER_ID unsigned int

Indicates the manager ID associated with this monitor.

DVM_Message

DISPLAY DVM_msg (%{DVM_MESSAGE}). Fields:

DVM_MESSAGE string

The DVM Message.

ENTRY_BECOMES_INVALID

An entry is now invalid. The cause could be any reason. Fields:

ENTRY_INDEX unsigned int

The entry index that has become invalid.

ENTRY_SET_DIRTY

Entry change, clean/dirty. Fields:

ENTRY_INDEX unsigned int

The entry index that has become clean or dirty.

STATUS unsigned int

True if the line was made dirty.

ERROR_MIXED_ATTRIBUTES_LINE

Mismatched line and transaction attributes. Fields:

ENTRY_INDEX unsigned int

The entry index that has collided.

NS_ADDR unsigned int

The security world and address of the first byte of the line, the MSB encodes the NS bit.

PAS enum

The Physical Address Space.

ERROR_MIXED_ATTRIBUTES_PAGE

Attributes differ between page and lines in cache from the same page. Fields:

NS_ADDR unsigned int

The security world and address of the first byte of the page, the MSB encodes the NS bit.

PAS enum

The Physical Address Space.

EVICTION

A line was evicted from the cache. Fields:

ADDRESS unsigned int

Base address of the victim cache line.

CLEAN bool

If true the cache retained the cache line after writing the dirty data.

ENTRY_INDEX unsigned int

The entry index of the victim.

NON_SECURE bool

If true the victim was NonSecure.

PAS enum

Physical Address Space.

IMPRECISE_ABORT

An operation produced an imprecise abort. Usually this is a WriteBack/WriteClean operation receiving an error response. Fields:

ADDRESS unsigned int

Address of the transaction that imprecise aborted.

DECERR bool

True if this is a DECERR response.

DOWNSTREAM_PORT_INDEX unsigned int

The downstream port index that received the imprecise abort.

NS bool

The NS state of the transaction (true is non-secure).

PAS enum

The Physical Address Space.

MAINTENANCE_CLEAN_SET_WAY

A cache maintenance by set-way operation occurred, possibly the 'clean' part of a clean-and-invalidate operation. Fields:

ENTRY_INDEX unsigned int

The entry index the set-way operation has mapped to.

IS_NON_SECURE bool

If the invalidate operation is non-secure (and so only affects non-secure lines), or secure (and so affects both).

PAS enum

Physical Address Space.

STATE enum

Line transformation incurred.

MAINTENANCE_INVALIDATE_ALL

A cache maintenance invalidate-all operation occurred. Fields:

IS_NON_SECURE bool

True if the invalidate operation is non-secure or realm, otherwise false: * Non-secure invalidates only non-secure lines * Secure invalidates secure and non-secure lines * Realm invalidates realm and non-secure lines * Root invalidates lines from any world.

PAS enum

The Physical Address Space.

MAINTENANCE_INV_SET_WAY

A cache maintenance by set-way operation occurred, possibly the 'invalidate' part of a clean-and-invalidate operation. Fields:

ENTRY_INDEX unsigned int

The entry index the set-way operation has mapped to.

IS_NON_SECURE bool

If the invalidate operation is non-secure (and so only affects non-secure lines), or secure (and so affects both).

PAS enum

The Physical Address Space.

STATE enum

Line transformation incurred.

change_dvm_disabled_ports

This reports a bitmap of which ports are disabled to receive DVM messages. Fields:

new_bitmap unsigned int

The new bitmap of disabled ports.

new_bitmap_list unsigned int

A list of port numbers that were disabled.

old_bitmap unsigned int

The old bitmap of disabled ports.

old_bitmap_list unsigned int

A list of port numbers that were disabled.

change_upstream_disabled_ports

This reports a bitmap of which upstream ports of the cache are *disabled* and so no snoop requests are sent to them. Fields:

new_bitmap unsigned int

The bitmap of the first 32 upstream ports that are disabled.

new_bitmap_list unsigned int

A list of port numbers that were disabled.

old_bitmap unsigned int

The bitmap of the first 32 upstream ports that are disabled.

old_bitmap_list unsigned int

A list of port numbers that were disabled.

entry_after_far_atomic

A cache line has had a far atomic applied to it. Fields:

atomic_operation enum

The operation that operated on this line.

be bool

The far atomic's big-endian flag.

data unsigned int

The data in the cache line.

entry_begin_address unsigned int

The start address of the entry.

entry_index unsigned int

The entry index in the cache.

incoming_data unsigned int

The incoming data from the far atomic. For CAS operations then this is compare-value and then the replacement-value.

manager_id unsigned int

The manager ID for the associated transaction.

ns enum

The security state of the line.

pas enum

The Physical Address Space.

prior_data unsigned int

The data that will be returned as the 'prior' data of the far atomic.

trans_begin_address unsigned int

The start address of the transaction.

entry_after_read

A cache line has been read. Fields:

ace_operation enum

The ACE operation that operated on this line.

data unsigned int

The data in the cache line.

dirty bool

Is the line ACE-dirty.

entry_begin_address unsigned int

The start address of the entry.

entry_index unsigned int

The entry index in the cache.

ns enum

The security state of the line.

pas enum

The Physical Address Space.

snoop bool

This transaction is a snoop transaction.

trans_begin_address unsigned int

The start address of the transaction.

unique bool

Is the line ACE-unique.

entry_after_write

A cache line has been written. Fields:

ace_operation enum

The ACE operation that operated on this line.

data unsigned int

The data in the cache line.

entry_begin_address unsigned int

The start address of the entry.

entry_index unsigned int

The entry index in the cache.

ns enum

The security state of the line.

pas enum

The Physical Address Space.

trans_begin_address unsigned int

The start address of the transaction.

2.164 PVMemoryProtectionEngine

This section describes the trace sources.

EC_MATCH

Access is made with owning encryption context (PAS, MECID). Fields:

IS_READ bool

Whether read or write access.

MECID unsigned int

MECID of the page.

PAGE_BASE unsigned int

Base address of page.

PAS enum

Physical address space of the page.

PAGE_OWNERSHIP_UPDATE

Encryption context (PAS, MECID) ownership update of 4KiB page is updated. Fields:

NEW_MECID unsigned int

New MECID of the page.

NEW_PAS enum

New Physical address space.

OLD_MECID unsigned int

Old MECID of the page.

OLD_PAS enum

Old Physical address space.

PAGE_BASE unsigned int

Base address of page.

PAGE_STATUS

Status of 4KiB page. Fields:

PAGE_BASE unsigned int

Base address of page.

Status enum

Status of page.

PAS_MISMATCH

Access is made with non-owning encryption context (PAS, MECID). Fields:

IS_READ bool

Whether read or write access.

NEW_MECID unsigned int

New MECID of the page.

NEW_PAS enum

New Physical address space.

OLD_MECID unsigned int

Old MECID of the page.

OLD_PAS enum

Old Physical address space.

PAGE_BASE unsigned int

Base address of page.

2.165 PVWriteBuffer

This section describes the trace sources.

ATOMIC_SLAVE_ACCESS

Information on the atomic operation at slave side. Fields:

ACCESS_RESULT enum

Result of atomic access.

COMPARE_VALUE unsigned int

Value to be compared with memory by CAS.

LOAD_VALUE unsigned int

Value to be loaded from memory as a result of atomic operation.

MANAGER unsigned int

ID of manager which initiated the transaction.

NS bool

Normal(true), Secure(false).

OPERAND_VALUE unsigned int

Incoming operand value provided by core.

OPERATION unsigned int

Type of atomic operation.

PADDR unsigned int

Physical address.

PAS enum

The physical address space.

STORE_VALUE unsigned int

Value to be stored to memory as a result of atomic operation.

WRITE_BUFFER_CONTENTS

Contents of a Line in WriteBuffer before/after updating them. Fields:

ADDRESS unsigned int

Address.

BEFORE_UPDATE bool

Before update.

DATA unsigned int

Data content.

LINE_INDEX unsigned int

Line index.

METADATA unsigned int

MetaData value.

NS bool

Normal(true), Secure(false).

PAS enum

The physical address space.

WRITE_BUFFER_HIT

An access hits in WriteBuffer. Fields:

ADDRESS unsigned int

Address.

IS_IMMEDIATELY_SERVED_BY_WB bool

true: This access is serviced in WriteBuffer? false: Sent downstream.

IS_READ bool

Is Read.

NS bool

Normal(true), Secure(false).

PAS enum

The physical address space.

WRITE_BUFFER_READ_DATA_DOWNSTREAM

Read one line of data from downstream. Fields:

ADDRESS unsigned int

Address.

DATA unsigned int

Data content.

NS bool

Normal(true), Secure(false).

PAS enum

The physical address space.

WRITE_BUFFER_READ_METADATA_DOWNSTREAM

Read one MetadataElement_t from downstream. Fields:

ADDRESS unsigned int

Address.

METADATA unsigned int

MetaData read from downstream.

NS bool

Normal(true), Secure(false).

PAS enum

The physical address space.

WRITE_BUFFER_WARNING_RESET_WHEN_DIRTY

Reset received whilst dirty lines exist in the write buffer.

WRITE_BUFFER_WARNING_RESET_WHEN_IN_TRANSACTIONAL_STATE

Reset received whilst the write buffer contains TME Transactional state.

2.166 SMMUv3AEM

This section describes the trace sources.

ArchMsg.Error.error

These messages are about activity occurring on the SMMU that is considered an error. Messages will only come out here if parameter `all_error_messages_through_trace` is true. `DISPLAY %{\output}`. Fields:

output string

The stream output.

ArchMsg.Error.fetch_from_memory_type_not_supporting_httu

A descriptor fetch from an HTTU-enabled translation regime to an unsupported memory type was made. Whilst the fetch itself may succeed, if an update to the descriptor was attempted then it would fail. Fields:

address unsigned int

The address of the descriptor fetch.

desc_inner enum

Inner cacheability of descriptor.

desc_outer enum

Outer cacheability of descriptor.

desc_prot enum

The instruction/data, bus-ns/s and privileged/user marking.

desc_sh enum

Shareability of descriptor.

ssd enum

The SSD of the transaction.

ssd_ns bool

Is the Security State Determination of the transaction non-secure.

stage unsigned int

The stage at which we had a problem.

streamid unsigned int

The StreamID of the transaction.

substreamid unsigned int

The SubstreamID of the transaction (or ~0u if no substreamid).

trans_id unsigned int

Transaction id.

ArchMsg.Error.ns_cmdq

There is an error associated with the command queue. Fields:

cmd_id unsigned int

Command ID of the command in error.

cons unsigned int

CONS of the command.

what string

What is wrong.

ArchMsg.Error.priq_streamid_truncated

The SMMU received a PCIe PRI request with a StreamID that was larger than that which the SMMU has been configured for. The StreamID that appears in the PRIQ entry will be truncated. Fields:

actual_streamid unsigned int

The actual StreamID that this request has.

sidsize unsigned int

The bit width of the SMMU for StreamIDs, as indicated by SMMU_IDR1.SIDSIZE.

trans_id unsigned int

The transaction ID that identifies this request.

trunc_streamid unsigned int

The truncated StreamID that will appear in the PRIQ entry.

ArchMsg.Error.priq_substreamid_truncated

The SMMU received a PCIe PRI request with a PASID prefix (SubstreamID) that was larger than that which the SMMU has been configured for. The SubstreamID that appears in the PRIQ entry will be truncated. Fields:

actual_substreamid unsigned int

The actual SubstreamID that this request has.

ssidsize unsigned int

The bit width of the SMMU for SubstreamIDs, as indicated by SMMU_IDR1.SSIDSIZE.

trans_id unsigned int

The transaction ID that identifies this request.

trunc_substreamid unsigned int

The truncated SubstreamID that will appear in the PRIQ entry.

ArchMsg.Error.s_cmdq

There is an error associated with the command queue. Fields:

cmd_id unsigned int

Command ID of the command in error.

cons unsigned int

CONS of the command.

what string

What is wrong.

ArchMsg.Error.tlb_entries_overlap

A TLB entry was attempted to be inserted into the TLB and was determined that it overlaps an existing entry. This check is not perfect but will catch simple errors. Fields:

do_f_tlb_conflict bool

Chosen to perform an F_TLB_CONFLICT.

end_address_of_new_entry unsigned int

End address of new entry.

end_address_of_old_entry unsigned int

End address of old entry.

index_of_new_entry unsigned int

Index of new entry.

index_of_old_entry unsigned int

Index of old entry.

start_address_of_new_entry unsigned int

Start address of new entry.

start_address_of_old_entry unsigned int

Start address of old entry.

why enum

Why it is an error for these two entries to overlap.

ArchMsg.Error.tlb_entry_not_invalidated_due_to_ril

An entry in the cache was not invalidated even though in the right address range because of the RIL fields of the command the entry do not match. Fields:

cmd_num unsigned int

The NUM field of the RIL part of the command.

cmd_ril_tg enum

The RIL_TG field of the RIL part of the command.

cmd_ril_ttl unsigned int

The TTL field of the RIL part of the command, zero means any level. 0x80 means match level 0 (and is from a DVM message).

cmd_scale unsigned int

The SCALE field of the RIL part of the command.

entry_id unsigned int

The entry id that is being invalidated.

tlb_entry string

The TLB entry.

ArchMsg.Error.vatos_sel_vmid_out_of_range

The SMMU_(S_)VATOS_SEL.VMID field was programmed with a VMID that was too wide for this implementation (SMMU_IDR0.VMID16 == 0). DISPLAY %**{ssd_ns:(s-|ns-)}**VMID:%**{vmid}** is out of range. Fields:

ssd_ns bool

The security state of the VATOS interface.

vmid unsigned int

The VMID programmed.

ArchMsg.Info.info

These are information messages about what is happening in the SMMU. DISPLAY %{output}.
Fields:

output string

The stream output.

ArchMsg.Warning.CMD_RESUME_no_transaction_resumed

A CMD_RESUME was issued that matched no transaction. Fields:

stag unsigned int

STAG in the CMD_RESUME.

streamid unsigned int

StreamID in the CMD_RESUME.

streamid_ns bool

The StreamID was for the non-secure world.

ArchMsg.Warning.atc_inv_strange

Something was odd about the CMD_ATC_INV. DISPLAY CMD_ATC_INV strange as: %{why}.
Fields:

cmd_id unsigned int

Command id.

ssd_of_cmdq enum

The SSD of the CMDQ.

why string

Why the CMD_ATC_INV was strange.

ArchMsg.Warning.bad_axi_stream_msi_addr_to_match_s

The signal was not driven at reset by the external system, but was driven after we first started using the default value we had assumed. The external system, if it is going to drive the signal, must do so at first reset so that both sides of the connection can agree on the value of this signal. Fields:

default_value unsigned int

The default value of the signal we have been assuming.

value unsigned int

The value of the signal.

ArchMsg.Warning.bad_conf_reset_of_SMMU_GBPA_ABORT

The signal was not driven at reset by the external system, but was driven after we first started using the default value we had assumed. The external system, if it is going to drive the signal, must do so at first reset so that both sides of the connection can agree on the value of this signal. Fields:

default_value bool

The default value of the signal we have been assuming.

value bool

The value of the signal.

ArchMsg.Warning.bad_conf_reset_of_SMMU_S_GBPA_ABORT

The signal was not driven at reset by the external system, but was driven after we first started using the default value we had assumed. The external system, if it is going to drive the signal, must do so at first reset so that both sides of the connection can agree on the value of this signal. Fields:

default_value bool

The default value of the signal we have been assuming.

value bool

The value of the signal.

ArchMsg.Warning.bad_conf_system_supports_btm

The signal was not driven at reset by the external system, but was driven after we first started using the default value we had assumed. The external system, if it is going to drive the signal, must do so at first reset so that both sides of the connection can agree on the value of this signal. Fields:

default_value bool

The default value of the signal we have been assuming.

value bool

The value of the signal.

ArchMsg.Warning.bad_conf_system_supports_cohacc

The signal was not driven at reset by the external system, but was driven after we first started using the default value we had assumed. The external system, if it is going to drive the signal, must do so at first reset so that both sides of the connection can agree on the value of this signal. Fields:

default_value bool

The default value of the signal we have been assuming.

value bool

The value of the signal.

ArchMsg.Warning.bad_conf_system_supports_httu

The signal was not driven at reset by the external system, but was driven after we first started using the default value we had assumed. The external system, if it is going to drive the signal, must do so at first reset so that both sides of the connection can agree on the value of this signal. Fields:

default_value bool

The default value of the signal we have been assuming.

value bool

The value of the signal.

ArchMsg.Warning.bad_conf_system_supports_sev

The signal was not driven at reset by the external system, but was driven after we first started using the default value we had assumed. The external system, if it is going to drive the signal, must do so at first reset so that both sides of the connection can agree on the value of this signal. Fields:

default_value bool

The default value of the signal we have been assuming.

value bool

The value of the signal.

ArchMsg.Warning.bad_reset_in

The signal was not driven at reset by the external system, but was driven after we first started using the default value we had assumed. The external system, if it is going to drive the signal, must do so at first reset so that both sides of the connection can agree on the value of this signal. Fields:

default_value bool

The default value of the signal we have been assuming.

value bool

The value of the signal.

ArchMsg.Warning.contig_bit_gives_too_large_region_for_TxSZ

If the contig bit was used then the size of the contig region would be larger than that indicated by TxSZ. Fields:

desc_kind enum

Descriptor kind.

el enum

Exception level.

input_address unsigned int

Input address.

log2_contig_region_size unsigned int

Log2(contig region size in bytes).

output_address unsigned int

Output address.

stage_and_level unsigned int

The stage in bits [7:4] and the level in [3:0]. Level -1 is 0xF.

vmid unsigned int

VMID if appropriate.

ArchMsg.Warning.contig_bit_has_inconsistent_input_and_output_address

If the contig bit was used then the some bits of the output address held in the descriptor and the input address must match. Fields:

desc_kind enum

Descriptor kind.

el enum

Exception level.

input_address unsigned int

Input address.

log2_contig_region_size unsigned int

Log2(contig region size in bytes).

match_mask unsigned int

Bits that must match.

output_address unsigned int

Output address.

stage_and_level unsigned int

The stage in bits [7:4] and the level in [3:0]. Level -1 is 0xF.

vmid unsigned int

VMID if appropriate.

ArchMsg.Warning.fetch_from_memory_type_not_supporting_httu

A descriptor fetch from an HTTU-enabled translation regime to an unsupported memory type was made. Whilst the fetch itself may succeed, if an update to the descriptor was attempted then it would fail. Fields:

address unsigned int

The address of the descriptor fetch.

desc_inner enum

Inner cacheability of descriptor.

desc_outer enum

Outer cacheability of descriptor.

desc_prot enum

The instruction/data, bus-ns/s and privileged/user marking.

desc_sh enum

Shareability of descriptor.

ssd enum

The SSD of the transaction.

ssd_ns bool

Is the Security State Determination of the transaction non-secure.

stage unsigned int

The stage at which we had a problem.

streamid unsigned int

The StreamID of the transaction.

substreamid unsigned int

The SubstreamID of the transaction (or ~0u if no substreamid).

trans_id unsigned int

Transaction id.

ArchMsg.Warning.msi_address_truncated

An MSI was generated, but the address was silently truncated due to the limited downstream address bus width. Fields:

pmcg_index_and_counter unsigned int

If this is a PMCG interrupt, then the top 16 bits are the PMCG index, the lower 16 bits are counter index.

ras_error_group_id_node_id_and_record_index_in_node unsigned int

If this is a RAS interrupt then this is the error group record id in bits[23:16], the node_id in [15:8] and the record index in the node in [7:0].

truncated_address unsigned int

The truncated address.

untruncated_address unsigned int

The untruncated address.

which enum

Which MSI this is.

ArchMsg.Warning.msi_lost

An MSI was attempted to be sent, but couldn't be sent. Fields:

id unsigned int

ID of this interrupt transaction.

kind enum

What kind of interrupt.

why enum

Why this interrupt was denied.

ArchMsg.Warning.pmcg_non_secure_world

A commentary on potential mis-programming of the PMCG when transitioning the PMCG between security states. Fields:

commentary string

The commentary.

ArchMsg.Warning.pmcg_programming_violates_security

A commentary on potential mis-programming of the PMCG when transitioning the PMCG between different security states. This is under the assumption that an agent from another security state could be writing to the PMCG. Fields:

commentary string

The commentary.

ArchMsg.Warning.priq_auto_response_failed_to_find_STE

The PRIQ was going to generate an auto-response, but failed to find an STE and so is returning a Failure message to the EndPoint which should disable the PRI interface of the EndPoint. Fields:

trans_id unsigned int

The ID of the PPR that is being lost.

why enum

Why the PPR is being lost.

ArchMsg.Warning.priq_overflow_bad_acking

Indicates that an overflow condition was acknowledged by writing to:- SMMU_PRIQ_CONS.OVACKFLG but an overflow condition did not exist. Fields:

new_ovackflg bool

The new value of the SMMU_PRIQ_CONS.OVACKFLG.

ArchMsg.Warning.priq_smmuen_forces_effective_priqen_low

If SMMUEN == 0, then the effective value of PRIQEN is 0. This warning is triggered when PRIQEN == 1 && SMMUEN == 0; which may not be what was intended. The PRIQ cannot be active if SMMUEN == 0.

ArchMsg.Warning.sev_lost

A SEV was lost because it isn't supported according to SMMU_IDRO.SEV. DISPLAY SEV was lost because: %{why}. Fields:

why enum

Why the SEV was generated.

ArchMsg.Warning.suspicious_overlapping_entries

Two DPT TLB entries are overlapping but they differ in ways that are potentially a SW error. Fields:

new_AC enum

The value of the 'AC' field that controls access to this region.

new_FWB bool

The region is FWB.

new_VMID unsigned int

The VMID, if any, associated with this region.

new_entry_index unsigned int

The entry index of the new DPT TLB entry.

new_entry_is_vmsa_formed bool

The new entry was inserted as a result of VMSA information.

new_input_end_incl_address unsigned int

The end inclusive address of the input range that this matches.

new_input_start_address unsigned int

The start address of the input range that this matches.

new_output_pas enum

The output PAS of this DPT region.

new_ssd enum

The SSD of the streams this region captures.

new_writeable bool

True if this region is writeable.

old_AC enum

The value of the 'AC' field that controls access to this region.

old_FWB bool

The region is FWB.

old_VMID unsigned int

The VMID, if any, associated with this region.

old_entry_index unsigned int

The entry index of the old DPT TLB entry.

old_entry_is_vmsa_formed bool

The old entry was inserted as a result of VMSA information.

old_input_end_incl_address unsigned int

The end inclusive address of the input range that this matches.

old_input_start_address unsigned int

The start address of the input range that this matches.

old_output_pas enum

The output PAS of this DPT region.

old_ssd enum

The SSD of the streams this region captures.

old_writeable bool

True if this region is writeable.

pas_differ bool

The output PAS of the two entries differ.

vmids_differ bool

The VMIDs are used by at least one of the AC schemes and are different.

vmsa_formed_writeable_while_DPT_entry_is_not bool

The VMSA-formed entry is writeable but the DPT Entry says it is not writeable.

ArchMsg.Warning.warning

These messages are about unusual (but not necessarily incorrect) activity occurring on the SMMU. DISPLAY %{output}. Fields:

output string

The stream output.

ArchMsg.Warning.warning_effective_EOPD_differs_from_what_would_be_cached_in_TLB

Effective value of EOPD differs from what would be cached in the TLB DISPLAY transaction (%{transaction_id}), sid (%{sid}), ssid (%{ssid}), ssd (%{ssd}), effective EOPD (%{effective_EOPD}), cached EOPD (%{cached_EOPD}). Fields:

cached_EOPD bool

The EOPD value that would be cached in the TLB.

effective_EOPD bool

The effective value of EOPD.

ssd enum

SSD.

streamid unsigned int

StreamID or ~0ull if NoStreamID.

substreamid unsigned int

SubstreamID or ~0u if SSValid == false.

transaction_id unsigned int

The transaction ID.

DPTTLB.tlb_commentary

A commentary from the cache about what is going on. Fields:

output string

The stream output.

DPTTLB.tlb_entry_invalidated

An entry in the cache has been invalidated. Fields:

entry_id unsigned int

The entry id that is being invalidated.

reason enum

Why the entry is being invalidated.

EVENTQ_config

Configuration of interrupt updated. Fields:

address unsigned int

Address of the MSI.

data unsigned int

Data payload of the MSI.

irqen bool

Appropriate bit of SMMU_s_IRQ_CTRL controlling this interrupt.

memattr enum

Memory type.

ns bool

PAS of bus attribute is non-secure.

pas enum

PAS of the MSI.

ra bool

read-allocate attribute.

sh enum

Shareability.

smmu_idr0_msi bool

Are MSIs supported for this security world.

tr bool

transient attribute.

wa bool

write-allocate attributes.

EVENTQ_irqen

The IRQEN has been changed. Fields:

new_value bool

The new value of IRQEN.

EVENTQ_irqen_ack

The IRQEN change has been acked. Fields:

new_value bool

The new value of IRQEN.

GERROR_config

Configuration of interrupt updated. Fields:

address unsigned int

Address of the MSI.

data unsigned int

Data payload of the MSI.

irqen bool

Appropriate bit of SMMU_s_IRQ_CTRL controlling this interrupt.

memattr enum

Memory type.

ns bool

PAS of bus attribute is non-secure.

pas enum

PAS of the MSI.

ra bool

read-allocate attribute.

sh enum

Shareability.

smmu_idr0_msi bool

Are MSIs supported for this security world.

tr bool

transient attribute.

wa bool

write-allocate attributes.

GERROR_irqen

The IRQEN has been changed. Fields:

new_value bool

The new value of IRQEN.

GERROR_irqen_ack

The IRQEN change has been acked. Fields:

new_value bool

The new value of IRQEN.

HACDBS_PROCESSING_COMPLETE_config

Configuration of interrupt updated. Fields:

address unsigned int

Address of the MSI.

data unsigned int

Data payload of the MSI.

irqen bool

Appropriate bit of SMMU_s_IRQ_CTRL controlling this interrupt.

memattr enum

Memory type.

ns bool

PAS of bus attribute is non-secure.

pas enum

PAS of the MSI.

ra bool
read-allocate attribute.

sh enum
Shareability.

smmu_idr0_msi bool
Are MSIs supported for this security world.

tr bool
transient attribute.

wa bool
write-allocate attributes.

HACDBS_PROCESSING_COMPLETE_irqen
The IRQEN has been changed. Fields:

new_value bool
The new value of IRQEN.

HACDBS_PROCESSING_COMPLETE_irqen_ack
The IRQEN change has been acked. Fields:

new_value bool
The new value of IRQEN.

HDBSS_TABLE_FULL_config
Configuration of interrupt updated. Fields:

address unsigned int
Address of the MSI.

data unsigned int
Data payload of the MSI.

irqen bool
Appropriate bit of SMMU_s_IRQ_CTRL controlling this interrupt.

memattr enum
Memory type.

ns bool
PAS of bus attribute is non-secure.

pas enum
PAS of the MSI.

ra bool
read-allocate attribute.

sh enum
Shareability.

smmu_idr0_msi bool

Are MSIs supported for this security world.

tr bool

transient attribute.

wa bool

write-allocate attributes.

HDBSS_TABLE_FULL_irqen

The IRQEN has been changed. Fields:

new_value bool

The new value of IRQEN.

HDBSS_TABLE_FULL_irqen_ack

The IRQEN change has been acked. Fields:

new_value bool

The new value of IRQEN.

PRIQ_config

Configuration of interrupt updated. Fields:

address unsigned int

Address of the MSI.

data unsigned int

Data payload of the MSI.

irqen bool

Appropriate bit of SMMU_s_IRQ_CTRL controlling this interrupt.

memattr enum

Memory type.

ns bool

PAS of bus attribute is non-secure.

pas enum

PAS of the MSI.

ra bool

read-allocate attribute.

sh enum

Shareability.

smmu_idr0_msi bool

Are MSIs supported for this security world.

tr bool

transient attribute.

wa bool

write-allocate attributes.

PRIQ_irqen

The IRQEN has been changed. Fields:

new_value bool

The new value of IRQEN.

PRIQ_irqen_ack

The IRQEN change has been acked. Fields:

new_value bool

The new value of IRQEN.

SMMU_CR0ACK_SMMUEN_hazarded_by_priq

The SMMU_r_CR0ACK.SMMUEN cannot acknowledge the change to SMMUEN because there are outstanding PRIQ writes.

SMMU_CR0ACK_SMMUEN_update

The acknowledge to SMMU_CR0.SMMUEN was performed. Fields:

new_value bool

The new value of the register.

SMMU_CR0_SMMUEN_old_set_complete

A set of transactions associated with the old value of SMMU_CR0.SMMUEN completed. Fields:

last bool

This is the last set completing.

old_value bool

The old value of the set completing.

SMMU_CR0_SMMUEN_write

A write to SMMU_CR0.SMMUEN occurred. Fields:

new_value bool

The new value of the register.

old_value bool

The old value of the register.

SMMU_GBPA_old_set_complete

A set of transactions associated with the old value of SMMU_GBPA completed. Fields:

last bool

This is the last set completing.

old_value unsigned int

The old value of the set completing.

SMMU_GBPA_update

The Update flag to SMMU_GBPA was lowered. Fields:

new_value unsigned int

The new value of the register.

SMMU_GBPA_write

A write to SMMU_GBPA occurred. Fields:

new_value unsigned int

The new value of the register.

old_value unsigned int

The old value of the register.

SMMU_S_CR0ACK_SMMUEN_update

The acknowledge to SMMU_S_CR0.SMMUEN was performed. Fields:

new_value bool

The new value of the register.

SMMU_S_CR0_SMMUEN_old_set_complete

A set of transactions associated with the old value of SMMU_S_CR0.SMMUEN completed. Fields:

last bool

This is the last set completing.

old_value bool

The old value of the set completing.

SMMU_S_CR0_SMMUEN_write

A write to SMMU_S_CR0.SMMUEN occurred. Fields:

new_value bool

The new value of the register.

old_value bool

The old value of the register.

SMMU_S_GBPA_old_set_complete

A set of transactions associated with the old value of SMMU_S_GBPA completed. Fields:

last bool

This is the last set completing.

old_value unsigned int

The old value of the set completing.

SMMU_S_GBPA_update

The Update flag to SMMU_S_GBPA was lowered. Fields:

new_value unsigned int

The new value of the register.

SMMU_S_GBPA_write

A write to SMMU_S_GBPA occurred. Fields:

new_value unsigned int

The new value of the register.

old_value unsigned int

The old value of the register.

S_EVENTQ_config

Configuration of interrupt updated. Fields:

address unsigned int

Address of the MSI.

data unsigned int

Data payload of the MSI.

irgen bool

Appropriate bit of SMMU_s_IRQ_CTRL controlling this interrupt.

memattr enum

Memory type.

ns bool

PAS of bus attribute is non-secure.

pas enum

PAS of the MSI.

ra bool

read-allocate attribute.

sh enum

Shareability.

smmu_idr0_msi bool

Are MSIs supported for this security world.

tr bool

transient attribute.

wa bool

write-allocate attributes.

S_EVENTQ_irgen

The IRQEN has been changed. Fields:

new_value bool

The new value of IRQEN.

S_EVENTQ_irqen_ack

The IRQEN change has been acked. Fields:

new_value bool

The new value of IRQEN.

S_GERROR_config

Configuration of interrupt updated. Fields:

address unsigned int

Address of the MSI.

data unsigned int

Data payload of the MSI.

irqen bool

Appropriate bit of SMMU_s_IRQ_CTRL controlling this interrupt.

memattr enum

Memory type.

ns bool

PAS of bus attribute is non-secure.

pas enum

PAS of the MSI.

ra bool

read-allocate attribute.

sh enum

Shareability.

smmu_idr0_msi bool

Are MSIs supported for this security world.

tr bool

transient attribute.

wa bool

write-allocate attributes.

S_GERROR_irqen

The IRQEN has been changed. Fields:

new_value bool

The new value of IRQEN.

S_GERROR_irqen_ack

The IRQEN change has been acked. Fields:

new_value bool

The new value of IRQEN.

S_HACDBS_PROCESSING_COMPLETE_config

Configuration of interrupt updated. Fields:

address unsigned int

Address of the MSI.

data unsigned int

Data payload of the MSI.

irqen bool

Appropriate bit of SMMU_s_IRQ_CTRL controlling this interrupt.

memattr enum

Memory type.

ns bool

PAS of bus attribute is non-secure.

pas enum

PAS of the MSI.

ra bool

read-allocate attribute.

sh enum

Shareability.

smmu_idr0_msi bool

Are MSIs supported for this security world.

tr bool

transient attribute.

wa bool

write-allocate attributes.

S_HACDBS_PROCESSING_COMPLETE_irqen

The IRQEN has been changed. Fields:

new_value bool

The new value of IRQEN.

S_HACDBS_PROCESSING_COMPLETE_irqen_ack

The IRQEN change has been acked. Fields:

new_value bool

The new value of IRQEN.

S_HDBSS_TABLE_FULL_config

Configuration of interrupt updated. Fields:

address unsigned int

Address of the MSI.

data unsigned int

Data payload of the MSI.

irqen bool

Appropriate bit of SMMU_s_IRQ_CTRL controlling this interrupt.

memattr enum

Memory type.

ns bool

PAS of bus attribute is non-secure.

pas enum

PAS of the MSI.

ra bool

read-allocate attribute.

sh enum

Shareability.

smmu_idr0_msi bool

Are MSIs supported for this security world.

tr bool

transient attribute.

wa bool

write-allocate attributes.

S_HDBSS_TABLE_FULL_irqen

The IRQEN has been changed. Fields:

new_value bool

The new value of IRQEN.

S_HDBSS_TABLE_FULL_irqen_ack

The IRQEN change has been acked. Fields:

new_value bool

The new value of IRQEN.

TLB.tlb_commentary

A commentary from the cache about what is going on. Fields:

output string

The stream output.

TLB.tlb_entry_invalidated

An entry in the cache has been invalidated. Fields:

entry_id unsigned int

The entry id that is being invalidated.

reason enum

Why the entry is being invalidated.

atc_inv_nop

The CMD_ATC_INV command is ignored as a **NOP**. This may be emitted multiple times if the CMD_ATC_INV is being ignored for multiple reasons. Fields:

address unsigned int

Untranslated Address to invalidate.

cmd_id unsigned int

Command id.

cons unsigned int

CONS of the command.

global bool

Global flag.

log2_size_in_bytes unsigned int

Log 2 of the size in bytes of the region to invalidate.

ssd_of_queue enum

The SSD of the CMDQ.

ssd_of_streamid enum

The SSD of the StreamID.

streamid unsigned int

StreamID in the invalidate command.

substreamid unsigned int

SubstreamID or ~0u if SSValid == false.

why enum

Why was NOPped.

atos_complete_fault

The ATOS operation completed with a fault. Fields:

effective_st1translate bool

Because of the settings and/or the ATOS type then the effective st1translate can be different.

fault_faddr unsigned int

The fault FADDR.

fault_faultcode enum

The fault code.

fault_reason enum

The fault reason.

ssd_ns bool

This is a non-secure ATOS operation.

st1translate bool

The StreamID/SubstreamID combination should have been translated by stage 1.

st2translate bool

The translation suffered a stage 2 translation.

state enum

The final transaction state of the ATOS operation.

vatos_vmid unsigned int

The VMID of the VATOS operation, or ~0u if not a VATOS operation.

atos_complete_fault_inv_req

The ATOS operation completed, faulted and generated an INV_REQ response. Fields:

ssd_ns bool

This is a non-secure ATOS operation.

vatos_vmid unsigned int

The VMID of the VATOS operation, or ~0u if not a VATOS operation.

why enum

Why the ATOS operation generated an INV_REQ.

atos_complete_success

The ATOS operation completed successfully. Fields:

base_addr unsigned int

The actual base address of region.

effective_st1translate bool

Because of the settings and/or the ATOS type then the effective st1translate can be different.

par_addr unsigned int

The PAR.ADDR field.

par_mair unsigned int

The memory attributes encoded as a MAIR.

par_ns bool

The PAR.NS field, for an SSD-ns request then this will always be 0.

par_sh enum

Shareability.

par_size bool

The PAR.Size field.

size_in_bytes unsigned int

The actual size in bytes of the region.

ssd_ns bool

This is a non-secure ATOS operation.

st1translate bool

The StreamID/SubstreamID combination should have been translated by stage 1.

st2translate bool

The translation suffered a stage 2 translation.

state enum

The final transaction state of the ATOS operation.

vatos_vmid unsigned int

The VMID of the VATOS operation, or ~0u if not a VATOS operation.

atos_run_set

The SMMU_s_GATOS_CTRL.RUN field was set to start the ATOS operation. Fields:

ssd_ns bool

This is a non-secure ATOS operation.

vatos bool

This is a VATOS operation.

atos_starting

The ATOS operation is starting. Fields:

addr unsigned int

The input address to the ATOS operation.

httui bool

Inhibit HTTU update.

ind bool

Instruction Data.

pnu bool

Privileged not User.

rnw bool

Read not Write.

ssd_ns bool

This is a non-secure ATOS operation.

ssec bool

If this is a secure ATOS operation then this is if it is secure or not.

streamid unsigned int

The StreamID requested.

substreamid unsigned int

The SubstreamID requested, or ~0u if no SubstreamID.

type enum

The requested ATOS type.

vatos_vmid unsigned int

The VMID of the VATOS operation, or ~0u if not a VATOS operation.

axi_stream_msi_addr_to_match_s

Address to use to send SMMU originated MSIs directly to the GIC. This pin is sampled on the negedge of the reset pin and has to be driven beforehand for the value to take effect. Fields:

value unsigned int

The value of the signal.

cd_cc.CD_commentary

A commentary from the cache about what is going on. Fields:

output string

The stream output.

cd_cc.CD_entry_invalidated

An entry in the cache has been invalidated. Fields:

entry_id unsigned int

The entry id that is being invalidated.

reason enum

Why the entry is being invalidated.

cd_entry_allocated

An CD entry has been allocated. Fields:

AssuredTranslation bool

The CD (and any L1CD) was fetched from an AssuredOnly stage 2 page and so might be capable of creating AssuredTranslations.

cd string

A textual description of the CD.

entry_id unsigned int

The id of the cache entry. Top bit is set if is a debug entry.

ns enum

For the non-secure world.

ssd enum

The SSD of the entry.

streamid unsigned int

The StreamID of the entry this will match.

substreamid unsigned int

The SubstreamID of the entry this will match. This may be zero for transactions without a SubstreamID.

conf_reset_of_SMMU_GBPA_ABORT

The pin is driven. This is the reset value of SMMU_GBPA.ABORT. This pin is sampled on the negedge of the reset pin and has to be driven beforehand for the value to take effect. Fields:

value bool

The value of the signal.

conf_reset_of_SMMU_S_GBPA_ABORT

The pin is driven. This is the reset value of SMMU_S_GBPA.ABORT. This pin is sampled on the negedge of the reset pin and has to be driven beforehand for the value to take effect. Fields:

value bool

The value of the signal.

conf_system_supports_btm

The pin is driven. This indicates the system supports BTM. This pin is sampled on the negedge of the reset pin and has to be driven beforehand for the value to take effect. Fields:

value bool

The value of the signal.

conf_system_supports_cohacc

The pin is driven. This indicates the system supports COHACC. This pin is sampled on the negedge of the reset pin and has to be driven beforehand for the value to take effect. Fields:

value bool

The value of the signal.

conf_system_supports_httu

The pin is driven. This indicates the system supports HTTU. This pin is sampled on the negedge of the reset pin and has to be driven beforehand for the value to take effect. Fields:

value bool

The value of the signal.

conf_system_supports_sev

The pin is driven. This indicates the system supports SEV. This pin is sampled on the negedge of the reset pin and has to be driven beforehand for the value to take effect. Fields:

value bool

The value of the signal.

dpttlb_entry_allocated

A DPT TLB entry has been allocated. Fields:

AC enum

The value of the 'AC' field that controls access to this region.

FWB bool

The region is FWB.

VMID unsigned int

The VMID, if any, associated with this region.

index unsigned int

Index of the TLB entry.

input_end_incl_address unsigned int

The end inclusive address of the input range that this matches.

input_start_address unsigned int

The start address of the input range that this matches.

output_pas enum

The output PAS of this DPT region.

ssd enum

The SSD of the streams this region captures.

trans_id unsigned int

The trans_id of the transaction that caused this allocation.

vmsa_formed bool

The entry was formed from VMSA information rather than from a DPT walk.

writeable bool

True if this region is writeable.

dpttlb_invalidate_intersects_but_does_not_cover_entry_range

ENCODED_SIZE < size of the region covered by the DPT entry so invalidation is not architecturally guaranteed. No invalidation is performed. Fields:

dpttlb_entry_id unsigned int

ID of the TLB entry.

entry_end_incl_address unsigned int

Last address covered by the TLB entry.

entry_start_address unsigned int

First address covered by the TLB entry.

invalidate_end_incl_address unsigned int

Last address covered by the invalidation range.

invalidate_start_address unsigned int

First address covered by the invalidation range.

ssd enum

The security state of the TLB entry.

dpttlb_overlapping_entries

Two DPT TLB entries are overlapping. Fields:

new_AC enum

The value of the 'AC' field that controls access to this region.

new_FWB bool

The region is FWB.

new_VMID unsigned int

The VMID, if any, associated with this region.

new_entry_index unsigned int

The entry index of the new DPT TLB entry.

new_entry_is_vmsa_formed bool

The new entry was inserted as a result of VMSA information.

new_input_end_incl_address unsigned int

The end inclusive address of the input range that this matches.

new_input_start_address unsigned int

The start address of the input range that this matches.

new_output_pas enum

The output PAS of this DPT region.

new_ssd enum

The SSD of the streams this region captures.

new_writeable bool

True if this region is writeable.

old_AC enum

The value of the 'AC' field that controls access to this region.

old_FWB bool

The region is FWB.

old_VMID unsigned int

The VMID, if any, associated with this region.

old_entry_index unsigned int

The entry index of the old DPT TLB entry.

old_entry_is_vmsa_formed bool

The old entry was inserted as a result of VMSA information.

old_input_end_incl_address unsigned int

The end inclusive address of the input range that this matches.

old_input_start_address unsigned int

The start address of the input range that this matches.

old_output_pas enum

The output PAS of this DPT region.

old_ssd enum

The SSD of the streams this region captures.

old_writeable bool

True if this region is writeable.

dvm_tlbinvalidate_complete

The DVM TLB Invalidate message completed. Fields:

id unsigned int

The unique id of this DVM message.

ok bool

The DVM message was OK.

dvm_tlbinvalidate_received

A DVM message for a TLB Invalidate has been received. Fields:

address unsigned int

The VA or IPA to use if match_address.

asid unsigned int

The ASID to match if match_asid.

by_ipa bool

The operation is for an IPA operation if match_address.

id unsigned int

The unique id of this DVM message.

ignored enum

The DVM message was ignored.

last_level bool

The operation is for last level if supported.

match_address bool

Match the address field.

match_asid bool

Match the asid field.

match_vmid bool

Match the vmid field.

num unsigned int

If a range operation, the NUM field. If a single-address operation this is 0.

prot enum

The protection level for which this TLB Invalidate will operate on.

security_world enum

The security world that this will apply to.

smmu_scale unsigned int

If a range operation, then the SCALE field with the meaning in the SMMU architecture which is different to the PE architecture. If a single-address operation this is 0.

stage1_only bool

The operation is for stage 1 only if supported.

tg enum

If a single-address or address-range operation, then the Translation Granule hint. Address-range operations always supply a Translation Granule.

translation_table_level enum

The leaf level of the translation table.

vmid unsigned int

The VMID to match if match_vmid.

found_tlb_entry_has_different_aset

Architecturally, a particular ASID either should be ASET0 or ASET1. However, we have managed to find a TLB entry that has a different ASET than that which we were searching for. This indicates a programming error. You should examine all contexts with this particular ASID/VMID and ensure they are consistent. Fields:

aset unsigned int

The ASET of this entry.

asid unsigned int

ASID if appropriate.

index unsigned int

Index of TLB entry.

input_start_address unsigned int

The start address of the input range that this matches.

scheme enum

The tagging scheme used.

ssd enum

The SSD of the tagging scheme.

ssd_ns enum

Security State.

vmid unsigned int

VMID if appropriate.

httu_update_abandoned_update

The HTTU update of a descriptor in memory was potentially possible, but it was behind an update that failed to apply cleanly. Fields:

AF enum

Whether an AF update should/could occur.

DBM enum

Whether a DBM update should/could occur.

address unsigned int

Address of HTTU update.

httu_inner enum

Inner cacheability of descriptor to update.

httu_outer enum

Outer cacheability of descriptor to update.

httu_prot enum

The instruction/data, bus-ns/s and privileged/user marking.

httu_sh enum

Shareability of descriptor to update.

mpam_pmg_and_partid unsigned int

Bits[23:16] are the MPAM PMG. Bits[15:0] are the physical PARTID.

mpam_sp enum

The MPAM_SP of the transaction.

original_descriptor unsigned int

The original descriptor that the model observed.

stage_and_level unsigned int

The stage (bits[7:4]) and level (bits[3:0]). A level of 0xF means -1.

trans_id unsigned int

The transaction id that this HTTU update is associated with.

what enum

What this descriptor represents.

httu_update_end_update

The attempted update of a descriptor in memory has occurred. Fields:

is_big_endian bool

The descriptor is big-endian in memory.

original_descriptor unsigned int

The original descriptor value.

result enum

The result of the attempt to update.

trans_id unsigned int

The transaction id that this HTTU update is associated with.

try_to_change_to_descriptor unsigned int

The new descriptor value that tried to replace the original.

value_that_was_in_memory unsigned int

The value that the compare-and-swap operation returned as the value that was in memory.

httu_update_not_done

A discretionary HTTU update could occur and the implementation choose not to do it. A mandatory_do_if_linked_performed means that this is a stage 2 leaf descriptor that maps a subsequent stage 1 leaf descriptor whose update is discretionary and if that subsequent discretionary update is going to occur then this update becomes mandatory. Fields:

AF enum

Whether an AF update should/could occur.

DBM enum

Whether a DBM update should/could occur.

address unsigned int

Address of HTTU update.

httu_inner enum

Inner cacheability of descriptor to update.

httu_outer enum

Outer cacheability of descriptor to update.

httu_prot enum

The instruction/data, bus-ns/s and privileged/user marking.

httu_sh enum

Shareability of descriptor to update.

mpam_pmg_and_partid unsigned int

Bits[23:16] are the MPAM PMG. Bits[15:0] are the physical PARTID.

mpam_sp enum

The MPAM_SP of the transaction.

original_descriptor unsigned int

The original descriptor that the model observed.

stage_and_level unsigned int

The stage (bits[7:4]) and level (bits[3:0]). A level of 0xF means -1.

trans_id unsigned int

The transaction id that this HTTU update is associated with.

what enum

What this descriptor represents.

httu_update_start_update

An HTTU update could occur and the implementation chose to try it. A mandatory_do_if_linked_performed means that this is a stage 2 leaf descriptor that maps a subsequent stage 1 leaf descriptor whose update is discretionary and if that subsequent discretionary update is going to occur then this update becomes mandatory. Fields:

AF enum

Whether an AF update should/could occur.

DBM enum

Whether a DBM update should/could occur.

address unsigned int

Address of HTTU update.

httu_inner enum

Inner cacheability of descriptor to update.

httu_outer enum

Outer cacheability of descriptor to update.

httu_prot enum

The instruction/data, bus-ns/s and privileged/user marking.

httu_sh enum

Shareability of descriptor to update.

is_big_endian bool

The descriptor will be written to memory as big-endian.

mecid unsigned int

The masked MECID used for the update transaction, or ~0u if not appropriate.

mpam_pmg_and_partid unsigned int

Bits[23:16] are the MPAM PMG. Bits[15:0] are the physical PARTID.

mpam_sp enum

The MPAM_SP of the transaction.

original_descriptor unsigned int

The original descriptor that the model observed.

stage_and_level unsigned int

The stage (bits[7:4]) and level (bits[3:0]). A level of 0xF means -1.

trans_id unsigned int

The transaction id that this HTTU update is associated with.

try_to_change_to_descriptor unsigned int

The new descriptor value that is going to try replace the original.

what enum

What this descriptor represents.

will_do_AF bool

What the implementation chose to do for the AF update.

will_do_DBM bool

What the implementation chose to do for the DBM update.

integration_mode_end_ras_level_interrupt_restored

RAS level sensitive interrupt restored due to integration mode ending. Fields:

is_tcu bool

Is TCU integration mode.

ras_interrupt enum

RAS interrupt being restored.

tbu_index unsigned int

TBU index. ~0u if it's TCU.

value bool

Level of the interrupt signal.

integration_mode_pmcg_interrupt_lost

PMCG interrupt lost due to being in integration mode. Fields:

is_tcu enum

Is TCU integration mode.

pmcg_interrupt enum

PMCG interrupt being dropped.

tbu_index unsigned int

TBU index. ~0u if it's TCU.

integration_mode_ras_interrupt_lost

RAS interrupt lost due to being in integration mode. Fields:

is_tcu enum

Is TCU integration mode.

ras_interrupt enum

RAS interrupt being dropped.

tbu_index unsigned int

TBU index. ~0u if it's TCU.

integration_mode_ras_level_interrupt_lost

RAS level sensitive interrupt lost due to being in integration mode. Fields:

is_tcu bool

Is TCU integration mode.

ras_interrupt enum

RAS interrupt being dropped.

tbu_index unsigned int

TBU index. ~0u if it's TCU.

value bool

Level of the interrupt signal.

integration_mode_start_ras_level_interrupt_cleared

RAS level sensitive interrupt cleared due to integration mode starting. Fields:

is_tcu bool

Is TCU integration mode.

ras_interrupt enum

RAS interrupt being cleared.

tbu_index unsigned int

TBU index. ~0u if it's TCU.

integration_mode_tcu_evento_lost

Evento lost due to being in integration mode.

integration_mode_tcu_interrupt_lost

Interrupt lost due to being in integration mode. Fields:

interrupt enum

Interrupt that is being dropped.

interrupt_returned

An interrupt/MSI returned from downstream. Fields:

InD_NS_PnU enum

Instruction/Data, Non-secure/Secure/Root/Realm, Privileged/User. Bit[1] = PAS[0], and bit[3] = PAS[1].

address unsigned int

Address of the MSI.

data unsigned int

Data payload of the MSI.

dcmdqcp_and_index unsigned int

If this not a DCMDQ MSI CMD_SYNC, then this is 0xFFFFFFFF. If it is then the DCMDQ QCP index is in bits[31:16] and the index in the QCP is in [7:0].

gpc_denied_msi bool

True if the MSI was denied as it failed its GPC checks. Thus the field 'ok' will be false.

id unsigned int

ID of this interrupt transaction.

inner_cache enum

The actual attributes of the access.

kind enum

What kind of interrupt.

mecid unsigned int

The MECID of the MSI, if applicable.

mpam_ns bool

For a non-RME system, this is true if the MPAM space is Non-secure. For an RME system, this is MPAM_SP[0].

mpam_pmg_and_partid unsigned int

Bits[23:16] are the MPAM PMG. Bits[15:0] are the physical PARTID.

mpam_sp enum

The MPAM_SP security state of the MPAM partition. 0 – secure, 1 – non-secure, 2 – root, 3 – realm.

msi_valid bool

MSI will attempt to be sent.

ok bool

Did the access return OK or an abort?.

outer_cache enum

The actual attributes of the access.

pas enum

The PAS of the MSI, if applicable.

pmcg_index_and_counter unsigned int

If this is a PMCG interrupt, then the top 16 bits are the PMCG index, the lower 16 bits are counter index.

ras_error_group_id_node_id_and_record_index_in_node unsigned int

If this is a RAS interrupt then this is the error group record id in bits[23:16], the node_id in [15:8] and the record index in the node in [7:0].

sh enum

Shareability.

interrupt_sent

An interrupt is raised. If it sends an MSI then this is *after* any device-dependent transform on the architectural attributes and so may differ from what is programmed. Fields:

InD_NS_PnU enum

Instruction/Data, Non-secure/Secure/Root/Realm, Privileged/User. Bit[1] = PAS[0], and bit[3] = PAS[1].

address unsigned int

Address of the MSI.

data unsigned int

Data payload of the MSI.

dcmdq_device_id unsigned int

If this is a DCMDQ then this is the DeviceID of the MSI write. Otherwise, 0xFFFF'ffff.

dcmdq_qcp_and_index unsigned int

If this not a DCMDQ MSI CMD_SYNC, then this is 0xFFFF'ffff. If it is then the DCMDQ QCP index is in bits[31:16] and the index in the QCP is in [7:0].

id unsigned int

ID of this interrupt transaction.

inner_cache enum

The actual attributes of the access.

kind enum

What kind of interrupt.

mecid unsigned int

The MECID of the MSI, if applicable.

mpam_ns bool

For a non-RME system, this is true if the MPAM space is Non-secure. For an RME system, this is MPAM_SP[0].

mpam_pmg_and_partid unsigned int

Bits[23:16] are the MPAM PMG. Bits[15:0] are the physical PARTID.

mpam_sp enum

The MPAM_SP security state of the MPAM partition. 0 – secure, 1 – non-secure, 2 – root, 3 – realm.

msi_valid bool

MSI will attempt to be sent.

outer_cache enum

The actual attributes of the access.

pas enum

The PAS of the MSI, if applicable.

pmcg_index_and_counter unsigned int

If this is a PMCG interrupt, then the top 16 bits are the PMCG index, the lower 16 bits are counter index.

qSID unsigned int

If this is a DCMDQ then this is the qSID of the MSI write. Otherwise, 0xFFFF'ffff.

ras_error_group_id_node_id_and_record_index_in_node unsigned int

If this is a RAS interrupt then this is the error group record id in bits[23:16], the node_id in [15:8] and the record index in the node in [7:0].

sh enum

Shareability.

l1cd_cc.L1CD_commentary

A commentary from the cache about what is going on. Fields:

output string

The stream output.

l1cd_cc.L1CD_entry_invalidated

An entry in the cache has been invalidated. Fields:

entry_id unsigned int

The entry id that is being invalidated.

reason enum

Why the entry is being invalidated.

l1cd_entry_allocated

An L1 CD entry has been allocated. Fields:

AssuredTranslation bool

The L1CD was fetched from an AssuredOnly stage 2 page and so might be capable of creating AssuredTranslations.

entry_id unsigned int

The id of the cache entry. Top bit is set if is a debug entry.

normalised_substreamid unsigned int

The first SubstreamID of the range of SubstreamIDs that this L1CD entry will match.

ns enum

For the non-secure world.

pa_12 unsigned int

The PA of the L2 CD table. This is L2Ptr << 12.

ssd enum

The SSD of the entry.

streamid unsigned int

The StreamID this CD is for.

valid bool

Is the entry valid.

l1ste_cc.L1STE_commentary

A commentary from the cache about what is going on. Fields:

output string

The stream output.

l1ste_cc.L1STE_entry_invalidated

An entry in the cache has been invalidated. Fields:

entry_id unsigned int

The entry id that is being invalidated.

reason enum

Why the entry is being invalidated.

l1ste_entry_allocated

An L1 STE entry has been allocated. Fields:

entry_id unsigned int

The id of the cache entry. Top bit is set if is a debug entry.

first_streamid_of_range unsigned int

The first StreamID of the range of StreamIDs that this L1STE entry will match.

ns enum

For the non-secure world.

num_entries_in_l2 unsigned int

The number of entries in the L2 table. This is $2^{*(\text{Span}-1)}$ or 0 if invalid.

pa_l2 unsigned int

The PA of the L2 ST table. This is $\text{L2Ptr} \ll 6$ and aligned to the size of the table.

ssd enum

The SSD of the entry.

level_interrupt_sent

A level interrupt changed state. Fields:

kind enum

What kind of interrupt.

ras_error_group_id_node_id_and_record_index_in_node unsigned int

If this is a level RAS interrupt then this is the error group record id in bits[23:16], the node_id in [15:8] and the record index in the node in [7:0].

set_high bool

Level interrupt state.

ns_cmd_sync_completed_irq

“Non-secure” CMD_SYNC completed its IRQ action. Fields:

cmd_id unsigned int

The ID of the command being issued.

cons_incl_wrap unsigned int

CONS pointer of CMD_SYNC.

is_synthetic bool

This is a synthetic sync. The cons_incl_wrap is 0xFFFF'ffff.

ns_cmd_sync_completed_no_action

Non-secure CMD_SYNC completed, there was no CMD_SYNC action (SIG_NONE). Fields:

cmd_id unsigned int

The ID of the command being issued.

cons_incl_wrap unsigned int

CONS pointer of CMD_SYNC.

is_synthetic bool

This is a synthetic sync. The cons_incl_wrap is 0xFFFF'ffff.

ns_cmd_sync_completed_sev

Non-secure CMD_SYNC completed its action of SIG_SEV. Fields:

cmd_id unsigned int

The ID of the command being issued.

cons_incl_wrap unsigned int

CONS pointer of CMD_SYNC.

is_synthetic bool

This is a synthetic sync. The cons_incl_wrap is 0xFFFF'ffff.

ns_cmd_sync_completed_with_error

Non-secure CMD_SYNC completed, however, there was an error associated with the completion of the CMD_SYNC. Fields:

error enum

The error associated with the CMD_SYNC.

cmd_id unsigned int

The ID of the command being issued.

cons_incl_wrap unsigned int

CONS pointer of CMD_SYNC.

is_synthetic bool

This is a synthetic sync. The cons_incl_wrap is 0xFFFF'ffff.

ns_cmd_sync_issued

Non-secure CMD_SYNC has now completed issuing and we now know if something is hazarding the CMD_SYNC from completing. Fields:

cmd_id unsigned int

The ID of the command being issued.

cons_incl_wrap unsigned int

CONS pointer of CMD_SYNC.

hazarded bool

The CMD_SYNC is currently hazarded from completing.

is_synthetic bool

This is a synthetic sync. The cons_incl_wrap is 0xFFFF'ffff.

ns_cmd_sync_issuing

Non-secure CMD_SYNC is starting to issue, we are going to start accounting for which things have to be part of the prior set of things that have to complete before the CMD_SYNC can complete. Fields:

cmd_id unsigned int

The ID of the command being issued.

cons_incl_wrap unsigned int

CONS pointer of CMD_SYNC.

is_synthetic bool

This is a synthetic sync. The cons_incl_wrap is 0xFFFF'ffff.

ns_cmd_sync_starting_completion_action_irq

Non-secure CMD_SYNC starting IRQ completion action. Fields:

cmd_id unsigned int

The ID of the command being issued.

cons_incl_wrap unsigned int

CONS pointer of CMD_SYNC.

is_synthetic bool

This is a synthetic sync. The cons_incl_wrap is 0xFFFF'ffff.

msiaddress unsigned int

The MSI address to use (or 0 if not sending an MSI).

msidata unsigned int

The MSI data to send.

ns_cmd_sync_starting_completion_action_sev

Non-secure CMD_SYNC starting its completion action of SIG_SEV. Fields:

cmd_id unsigned int

The ID of the command being issued.

cons_incl_wrap unsigned int

CONS pointer of CMD_SYNC.

is_synthetic bool

This is a synthetic sync. The cons_incl_wrap is 0xFFFF'ffff.

ns_cmdq_cmd_sync_error

The CMD_SYNC can fault if an ATC Invalidate timeout occurs. The timeout will be attached to the next CMD_SYNC on the queue on which the invalidate was launched. Fields:

cerror enum

The CERROR to attach.

cmd_queue unsigned int

The top bits indicate the CMDQ id. These bits are attached to all cmd_ids generated for this queue. The bits [63:44] contain the queue ID. [63:62] 0: s-ssd, 1: reserved, 2: ns-ssd, 3: rl-ssd [61] is ECMDQ [59:52] QCP index [51:44] ECMDQ index in QCP.

herror enum

The HERROR to attach, or HERROR_NONE if we are attaching a CERROR.

ns_cmdq_cmd_sync_error_delivered

The CMD_SYNC can fault if an ATC Invalidate timeout occurs. This is the CMD_SYNC delivering that fault. Fields:

cerror enum

The CERROR to attach.

cmd_id unsigned int

Command ID of the CMD_SYNC.

ns_cmdq_ecmdq_enack

Trace the changes to ENACK. Fields:

cerror enum

The error being exposed, if any.

cmd_queue unsigned int

The top bits indicate the CMDQ id. These bits are attached to all cmd_ids generated for this queue. The bits [63:44] contain the queue ID. [63:62] 0: s-ssd, 1: reserved, 2: ns-ssd. 3: rl-ssd [61] is ECMDQ [59:52] QCP index [51:44] ECMDQ index in QCP.

enack bool

The value of ENACK.

forced_error bool

The queue was enabled with ERR != ERRACK and so the implementation chooses to trigger the error reporting mechanism.

ns_cmdq_issue

A command is actually being executed. Fields:

cmd_id unsigned int

Command ID of the command being executed.

cons unsigned int

CONS of the command.

what string

What is being executed.

ns_cmdq_pointers_state

The command queue pointers. Fields:

cmd_queue unsigned int

The top bits indicate the CMDQ id. These bits are attached to all cmd_ids generated for this queue. The bits [63:44] contain the queue ID. [63:62] 0: s-ssd, 1: reserved, 2: ns-ssd. 3: rl-ssd [61] is ECMDQ [59:52] QCP index [51:44] ECMDQ index in QCP.

disable_fetch bool

The model has decided that as you programmed PROD/CONS inconsistently that it would not fetch as it has no confidence as to which are commands and which are not.

q_issuable_and_issued_cons unsigned int

The CONS pointer of the region that is issuable and/or issued (including wrap).

q_issuable_and_issued_prod unsigned int

The PROD pointer of the region that is issuable and/or issued (including wrap).

q_programmer_cons unsigned int

The CONS pointer as seen by the programmer (including wrap).

q_programmer_prod unsigned int

The PROD pointer as seen by the programmer (including wrap).

q_unfetched_cons unsigned int

The CONS pointer of the region unfetched (including wrap).

q_unfetched_prod unsigned int

The PROD pointer of the region unfetched (including wrap).

ns_cmdq_state

A description of what the CMDQ can do now. Fields:

can_cerror bool

There is a CERROR that can be recognised.

can_cmd_sync bool

There is a CMD_SYNC that can be recognised.

can_fetch bool

The SMMU can fetch.

can_herror bool

There is an HERROR that can be recognised.

can_issue bool

There are fetched commands that are eligible to be issued.

can_update_cons bool

The programmer view of CMDQ_CONS can be moved to say some have been consumed.

cmd_queue unsigned int

The top bits indicate the CMDQ id. These bits are attached to all cmd_ids generated for this queue. The bits [63:44] contain the queue ID. [63:62] 0: s-ssd, 1: reserved, 2: ns-ssd, 3: rl-ssd [61] is ECMDQ [59:52] QCP index [51:44] ECMDQ index in QCP.

cmd_sync enum

Why a CMD_SYNC cannot be issued.

cmdqen bool

CMDQEN, when enabled then the CMDQ can fetch commands.

cmdqen_1_to_0_unacknowledged bool

CMDQEN has been set to 0, but we have yet to acknowledge it.

current-real-cerror enum

The current CERROR that would be visible to the programmer if there was an error.

fetch-disabled_due_to_programmer_error bool

The PROD/CONS pointer was inconsistent and the model decided to stop processing commands.

fetch-number_of_unfetched_commands unsigned int

The number of commands that have yet to be fetched from memory, or ~0u if CMDQEN == 0.

fetch-state enum

The state of the fetch state machine.

fetch-too_many_commands_outstanding_so_not_fetching_any_more bool

If there are too many issued commands, or pending to be issued then we won't fetch any more.

issue-number_of_issuable_commands unsigned int

The number of commands that have been fetched but not yet issued.

issue-number_of_unissuable_commands unsigned int

This might include commands behind an illegal command, or behind a CMD_SYNC. Under some circumstances this can include a command representing an external abort.

issue-state enum

The internal issued state machine state.

outstanding_fetches_since_reset unsigned int

The number of outstanding fetches.

synthetic_sync_can_be_completed bool

The synthetic sync that is outstanding can be completed.

waiting_for_synthetic_sync bool

There is a synthetic sync that is outstanding.

ns_eventq_adding_to_cmd_sync_prior_set

The event created by the specified transaction id must become visible to the programmer before the CMD_SYNC can complete. Fields:

trans_id unsigned int

The transaction id that caused the event.

ns_eventq_auto_retry_stalled_transaction

A transaction that was stalled but not yet reported on the Event Queue was retried. The retry can happen if the transaction didn't write because it was de-duplicated behind a reported stalling transaction and that stalling transaction was CMD_RESUMEd, or if the queue was unwritable and the queue is now writable. Fields:

trans_id unsigned int

Transaction id of the auto-retried transaction.

why enum

Why the transaction retried.

ns_eventq_cmd_sync_unhazarded

The CMD_SYNC has been unhazarded as the prior set is empty and as much of it as required is visible in the programmers' view. Fields:

prod_incl_wrap unsigned int

The prod index including wrap bit that was required to become visible.

prog_prod_incl_wrap unsigned int

The current programmer-visible prod including wrap bit.

ns_eventq_eventqen_and_eventqenack

Trace the values of EVENTQEN and EVENTQENACK as they change. Fields:

about enum

Some more information about the queue state.

cons_incl_wrap unsigned int

The CONS pointer (including wrap bits).

eventqen bool

EVENTQEN.

eventqenack bool

EVENTQENACK.

prod_incl_wrap unsigned int

The PROD pointer (including wrap bits).

ns_eventq_external_abort

A particular eventq record got an external abort. The event will subsequently appear in the ns/s_eventq_lost_event_records trace. Fields:

prod unsigned int

The prod pointer (excluding wrap bit) of the event record that aborted.

trans_id unsigned int

The transaction id that produced the event record that aborted.

ns_eventq_lost_event_records

Event records that are lost and never record appear on this trace. Fields:

dword0 unsigned int

Double word 0 of event.

dword1 unsigned int

Double word 1 of event.

dword2 unsigned int

Double word 2 of event.

dword3 unsigned int

Double word 3 of event.

event enum

Event code that was thrown away.

prod unsigned int

The PROD index of the event if appropriate or ~0u otherwise.

reason enum

The reason the record was lost.

trans_id unsigned int

The transaction id that this corresponds to.

ns_eventq_overflow_acknowledged

When events are lost then they appear on the eventq_lost_event_records_* trace. This will activate the overflow flag in the corresponding SMMU_s_EVENTQ_PROD register. Software can then acknowledge that flag using the overflow acknowledge flag in SMMU_s_EVENTQ_CONS. Fields:

bad bool

If true then the programmer has attempted to acknowledge an overflow condition that didn't exist.

count_of_events_lost_due_to_overflow unsigned int

This is the number of events that were lost because the event queue was full. This does not include any lost because of an EVENTQ_ABT_ERR.

ns_eventq_pending_event_records

Event records that are pending to be dealt with by the eventq state machine. A specific event can be traced multiple times if it becomes part of the prior set of transactions of a CMD_SYNC that has to complete before the CMD_SYNC is allowed to be completed. Fields:

dword0 unsigned int

Double word 0 of event.

dword1 unsigned int

Double word 1 of event.

dword2 unsigned int

Double word 2 of event.

dword3 unsigned int

Double word 3 of event.

event enum

Event code.

is_prior_set bool

The transaction forms part of the prior set of a CMD_SYNC that have to become visible (or thrown away if necessary) before the CMD_SYNC can complete.

trans_id unsigned int

The transaction id that this corresponds to.

ns_eventq_prod_cons_updated

This records when the programmer visible PROD/CONS pointer is updated. More can be visible in memory than are advertised in the programmer-visible PROD pointer. This also traces the latest PROD that could have been advertised. Fields:

mem_prod unsigned int

The PROD pointer value for all records that are visible in the memory.

num_events_available unsigned int

The number of events available to the programmer.

prog_cons unsigned int

The programmer-visible CONS pointer.

prog_prod unsigned int

The programmer-visible PROD pointer.

what enum

What changed?.

ns_eventq_raising_eventq_abt_err

EVENTQ_ABT_ERR can now be raised.

ns_eventq_removing_from_cmd_sync_prior_set

The event captured by a CMD_SYNC and created by the specified transaction id is now in memory. Fields:

prod unsigned int

The prod index (no wrap bit) that must become visible to the programmer to see this record. If not head_of_line then the record will not become visible and this field is meaningless.

reason enum

The reason the event was removed.

trans_id unsigned int

The transaction id that caused the event.

ns_eventq_stashing_unreported_stalled_transaction_for_auto_retry

A stall event record would have been lost, instead it is stashed away to auto-retry when it can. Fields:

dword0 unsigned int

Double word 0 of event.

dword1 unsigned int

Double word 1 of event.

dword2 unsigned int

Double word 2 of event.

dword3 unsigned int

Double word 3 of event.

event enum

Event code that was thrown away.

reason enum

The reason the record was lost.

trans_id unsigned int

The transaction id that this corresponds to.

ns_eventq_write_event

An event is committed to being written. Fields:

dword0 unsigned int

Double word 0 of event.

dword1 unsigned int

Double word 1 of event.

dword2 unsigned int

Double word 2 of event.

dword3 unsigned int

Double word 3 of event.

event enum

Event code that was thrown away.

prod unsigned int

The PROD pointer (without wrap) that this event will be written to.

stall_event bool

Describes an event for a stalled transaction.

trans_id unsigned int

Transaction ID that generated this event.

ns_eventq_write_event_visible_in_memory

An event is now visible in memory. Fields:

abort bool

The write aborted.

dword0 unsigned int

Double word 0 of event.

dword1 unsigned int

Double word 1 of event.

dword2 unsigned int

Double word 2 of event.

dword3 unsigned int

Double word 3 of event.

event enum

Event code that was thrown away.

prod unsigned int

The PROD pointer (without wrap) that this event will be written to.

stall_event bool

Describes an event for a stalled transaction.

trans_id unsigned int

Transaction ID that generated this event.

ns_gerror_inverted

A GERROR bit was attempted to be inverted to record a fault. Fields:

already_different_to_gerrorn bool

True if the bit was already different to the corresponding bit in GERRORN and so it was not inverted.

gerrorn unsigned int

GERRORN register value.

interrupt_potentially_sent bool

An interrupt is potentially sent, depending on IRQEN.
GERROR(N).MSI_GERROR_ABT_ERR does not send an interrupt.

new_gerror unsigned int

GERROR register value after any inversion.

which enum

Which bit was attempted to be inverted.

ns_gerrorn_acknowledge

A GERROR was acknowledged by SW. Fields:

which enum

Which GERROR was acknowledged.

pmcg_irq_config

The interrupt configuration of the Performance Monitor Counter Group (PMCG) changed.
Fields:

address unsigned int

Address of the MSI.

data unsigned int

Data payload of the MSI.

irqen bool

SMMU_PMCG_CTRL.IRQEN.

memattr enum

Memory type.

mpam_ns bool

The NS state of the MPAM PARTID and MPAM PMG.

mpam_pmg_and_partid unsigned int

Bits[23:16] are the MPAM PMG. Bits[15:0] are the physical PARTID.

msi_supported bool

Are MSIs supported for this security world.

ns bool

Non-secure bus attribute.

number_of_interrupts_in_flight unsigned int

The number of interrupts that have been committed to be produced or in flight.

pmcg_index unsigned int

Index of the PMCG.

sh enum

Shareability.

smmu_pmcg_gmpam_Update bool

The SMMU_PMC_GMPAM.Update flag. Only when this is zero are writes predictable.

ssd_ns bool

True if this PMCG controlled by the non-secure world.

pmcg_irqen

A trace of SMMU_PMC_GIRQ_CTRL.IRQEN. Fields:

new_value bool

The new value of IRQEN.

pmcg_index unsigned int

Index of the PMCG.

ssd_ns bool

True if this PMCG controlled by the non-secure world.

pmcg_irqen_ack

A trace of SMMU_PMC_GIRQLACK.IRQEN. Fields:

new_value bool

The new value of IRQEN.

pmcg_index unsigned int

Index of the PMCG.

ssd_ns bool

True if this PMCG controlled by the non-secure world.

pmcg_merging_interrupts

An interrupt was wanted to be generated, but one was already pending so the two were merged together. Fields:

pmcg_index unsigned int

Index of the PMCG.

ssd_ns bool

True if this PMCG controlled by the non-secure world.

pmcg_trigger

A PMCG counter has been triggered. Fields:

NoStreamID bool

True if the transaction is a NoStreamID transaction.

counter_index unsigned int

The index of the counter within the PMCG.

event_id unsigned int

The event id that has been triggered.

ns_event bool

Is the event associated with non-secure state.

pmcg_index unsigned int

Index of the PMCG.

prior_counter_value unsigned int

The Counter value *before* the event has incremented it.

ssd enum

The security state associated with the event.

streamid unsigned int

The StreamID associated with the event, if there is one.

tbu_index unsigned int

The TBU index of the transaction, or ~0u if not applicable.

pmu_active_counter

Traces what active counters are in a PMCG and what StreamIDs it might filter on. Those counters that trace StreamIDs for multiple security states, or those that are not filtered by StreamID, will appear multiple times, once for each security state. All active counters for a PMCG are traced one after another. Fields:

NoStreamID bool

True if NoStreamID transactions will be traced.

begin_streamid unsigned int

The start StreamID to filter on.

counter_index unsigned int

The counter index within the PMCG.

end_incl_streamid unsigned int

The end inclusive StreamID to filter on.

evcnt unsigned int

The current count.

event_id unsigned int

The event ID to filter.

ns bool

Are the StreamIDs non-secure?.

pmcg_index unsigned int

The index of the PMCG that this counter belongs to.

ssd enum

SSD of the StreamID.

tbu_index_to_match unsigned int

The TBU index that must match, or ~0u if no matching applicable.

pmu_all_counters_in_pmcg_became_inactive

The PMCG was tracing some events and now is not tracing any. Fields:

pmcg_index unsigned int

The index of the PMCG.

pmu_capture

For some reason, a capture event occurred. Fields:

pmcg_index unsigned int

The index of the PMCG that the capture occurred on.

why enum

Why did the capture occur?.

pmu_counter_configured_to_use_unsupported_event

An enabled counter was configured to use a unsupported event. Fields:

counter_index unsigned int

The counter index within the PMCG.

event_id unsigned int

The unsupported event id.

pmcg_index unsigned int

The index of the PMCG that this counter belongs to.

pmu_counter_overflowed

A counter in a particular PMCG overflowed. Fields:

already_overflowed bool

True if the overflow flag was already set.

capture bool

True if it captured the other counter values.

counter_index unsigned int

The counter index within the PMCG.

interrupt bool

True if going to attempt to generate an interrupt.

interrupt_action enum

The interrupt action that is going to occur.

pmcg_index unsigned int

The index of the PMCG that this counter belongs to.

ssd_ns bool

The PMCG is controlled by the Non-secure security state.

priq_auto_response

A PRIQ auto response is generated. Fields:

prgindex unsigned int

The PRG Index of the response.

resp enum

The response to send.

streamid unsigned int

The StreamID of the response.

substreamid unsigned int

The PASID/SubstreamID attached to the response, or ~0u if there is none.

trans_id unsigned int

The ID of the PPR that is being lost.

why enum

Why the PPR is being lost.

priq_lost_ppr

The PRIQ is not reporting a PPR into the queue. It may or may not generate an auto-response. Fields:

is_stop_marker bool

The PPR being lost is a Stop Marker.

resp enum

What, if any, auto-response will be.

trans_id unsigned int

The ID of the PPR that is being lost.

why enum

Why the PPR is being lost.

priq_overflow_acking

Indicates that an overflow condition was acknowledged by writing to:-
SMMU_PRIQ_CONS.OVACKFLG. Fields:

new_ovackflg bool

The new value of the SMMU_PRIQ_CONS.OVACKFLG.

priq_overflow_asserting

Indicates that we are toggling the SMMU_PRIQ_PROD.OVFLG because we lost a PRI request due to the PRIQ being full and an existing overflow condition does not already exist. Fields:

new_ovflg bool

The new value of the SMMU_PRIQ_PROD.OVFLG.

trans_id unsigned int

The transaction ID of the PPR that caused the overflow.

priq_received

A PRIQ Request has been received and is queued waiting processing. This does not mean that it has been written to the PRIQ but has been placed in a pending queue awaiting a decision about what to do about it. Fields:

L bool

The Last in the Page Request Group (PRG).

R bool

Read permission is requested.

W bool

Read permission is requested.

prgindex unsigned int

The Page Request Group (PRG) index that is being used to identify this request to the EndPoint.

stop_marker bool

If LRW == 0b100 then this indicates this is a stop marker.

streamid unsigned int

The StreamID this request is associated with.

substreamid unsigned int

The SubstreamID (PASID) this request is associated with, of ~0u if no SubstreamID.

substreamid_P bool

If has a SubstreamID (!= ~0u) then Privileged Mode Requested.

substreamid_X bool

If has a SubstreamID (!= ~0u) then Execution Requested (R must also be 1).

trans_id unsigned int

The transaction ID that identifies this request.

untranslated_address unsigned int

The Untranslated Address that this is requesting.

priq_state

The trace of various fields that indicate the state of the PRIQ. Fields:

cons_incl_wrap unsigned int

The value of SMMU_PRIQ.CONS.RD_and_RD_wrap.

number_of_pprs unsigned int

The number of PPRs as indicated by the CONS/PROD.

number_of_pprs_still_to_deal_with unsigned int

This is the number of PPRs that are currently waiting to either be written to the PRIQ, or auto-responded to.

number_of_priq_writes_in_flight unsigned int

The number of writes to the PRIQ that are currently in flight.

ovackflg bool

The OVACKFLG which if different to OVFLG is used to indicate that the PRIQ overflowed.

ovflg bool

The OVFLG which if different to OVACKFLG is used to indicate that the PRIQ overflowed.

priq_abt_err bool

There is an active SMMU_GERROR{N}.PRIQ_ABT_ERR.

priqen bool

The value of SMMU_CRO.PRIQEN. The *effective* value is 0 if SMMUEN == 0.

priqenack bool

The value of SMMU_CROACK.PRIQEN.

prod_incl_wrap unsigned int

The value of SMMU_PRIQ.CONS.WR_and_WR_wrap.

queue_disabled_due_to_prior_programming_error bool

The queue was disabled as the programmer got CONS/PROD into an inconsistent state. The model will disable the PRIQ until SW disables and re-enables the queue via SMMU_CRO.PRIQEN.

smmuen bool

The value of SMMU_CRO.SMMUEN. If this is 0 then the effective PRIQEN is 0.

smmuenack bool

The value of SMMU_CRO.SMMUEN. If this is 0 then the effective PRIQEN is 0.

table_size_in_elements unsigned int

The size of the table in the number of items it can hold.

priq_write_aborted

A PRIQ write aborted. The PRIQ now goes into an error state and will start auto-responding to PRI requests. Fields:

prod_incl_wrap unsigned int

The PROD including the Wrap bit where we were trying to write.

trans_id unsigned int

The transaction ID of the PPR that aborted.

priq_write_ok

A PRIQ write completed OK. Fields:

prod_incl_wrap unsigned int

The PROD including the Wrap bit where we wrote..

trans_id unsigned int

The transaction ID of the PPR.

priq_write_start

A PRIQ request has been received and is going to be attempt to be written to the queue. Fields:

L bool

The Last in the Page Request Group (PRG).

R bool

Read permission is requested.

W bool

Read permission is requested.

prgindex unsigned int

The Page Request Group (PRG) index that is being used to identify this request to the EndPoint.

prod_incl_wrap unsigned int

PROD position including the wrap bit.

stop_marker bool

If LRW == 0b100 then this indicates this is a stop marker.

streamid unsigned int

The StreamID this request is associated with.

substreamid unsigned int

The SubstreamID (PASID) this request is associated with, of ~0u if no SubstreamID.

substreamid_P bool

If has a SubstreamID (!= ~0u) then Privileged Mode Requested.

substreamid_X bool

If has a SubstreamID (!= ~0u) then Execution Requested (R must also be 1).

trans_id unsigned int

The transaction ID that identifies this request.

untranslated_address unsigned int

The Untranslated Address that this is requesting.

ptw_read

Page Table Walk (read). This is the result of the physical access that the SMMU is making.
Fields:

abort enum

Non-zero if the access aborted/failed.

adomain enum

The actual attributes of the access that was used, after IMP DEF mangling.

aprot enum

The actual attributes of the access that was used, after IMP DEF mangling. PAS[1] (bit[3]), Instruction/Data (bit[2]), PAS[0] (bit[1]), Privileged/User (bit[0]).

data unsigned int

The data fetch if it didn't abort.

desckind enum

The kind of descriptor.

inner_cache enum

The actual attributes of the access that was used, after IMP DEF mangling.

input_address unsigned int

Input address this lookup is for (after TBI normalisation).

mecid unsigned int

The MECID of the TTD fetch.

ns enum

Non-Secure on the bus.

outer_cache enum

The actual attributes of the access that was used, after IMP DEF mangling.

pa_address unsigned int

Physical address.

pas enum

The PAS of the TTD fetch.

ssd enum

SSD of the transaction.

ssd_ns enum

Security state of stream.

streamid unsigned int

The StreamID this is for.

substreamid unsigned int

The SubstreamID this is for.

trans_id unsigned int

Transaction id.

ttb_grain_stage_and_level unsigned int

Which TTB used in bits[31:24]. Which Grain size used in bit[23:16]:- 0 – 4 KiB 1 – 16 KiB 2 – 64 KiB 3 – 64 KiB ARMv8.2 where descriptor bits [15:12] for bits [51:48] of output address 4 – 4 KiB FEAT_LPA2 where address is {bits[9:8], bits[49:12]} 5 – 16 KiB FEAT_LPA2 where address is {bits[9:8], bits[49:14], 2'b00} 6 – 64 KiB ARMv8.2 where descriptor bits [15:12] are ignored Stage in bits[15:8]. Level in bits[7:0], level -1 is 0xF.

ptw_read_stl_invalid_long_descriptor

Page Table Walk (read). The descriptor that the PTW fetched is invalid. Fields:

desckind enum

The kind of descriptor.

input_address unsigned int

Input address this lookup is for (after TBI normalisation).

mecid unsigned int

The MECID of the TTD fetch.

ns enum

Non-Secure on the bus.

pa_address unsigned int

Physical address.

pas enum

The PAS of the TTD fetch.

ssd enum

SSD of the transaction.

ssd_ns enum

Security state of stream.

streamid unsigned int

The StreamID this is for.

substreamid unsigned int

The SubstreamID this is for.

trans_id unsigned int

Transaction id.

ttb_grain_stage_and_level unsigned int

Which TTB used in bits[31:24]. Which Grain size used in bit[23:16]:- 0 – 4 KiB 1 – 16 KiB 2 – 64 KiB 3 – 64 KiB ARMv8.2 where descriptor bits [15:12] for bits [51:48] of output address 4 – 4 KiB FEAT_LPA2 where address is {bits[9:8], bits[49:12]} 5 – 16

KiB FEAT_LPA2 where address is {bits[9:8], bits[49:14], 2'b00} 6 – 64 KiB ARMv8.2 where descriptor bits [15:12] are ignored Stage in bits[15:8]. Level in bits[7:0], level -1 is 0xF.

ptw_read_st1_leaf_long_descriptor

Page Table Walk (read). The descriptor that the PTW fetched is a block or page and this is the decode. Fields:

AF bool

Access Flag.

AMEC enum

Use Primary or Alternative MECID.

AP21 enum

The access permissions.

AttrIndx210 unsigned int

The attribute index into the MAIR0/1. If AIE is implemented then this is the full index AttrIndx[3:0].

DBM bool

Dirty Bit Modifier. This only has meaning if running in AArch64 mode.

NS bool

The encoding is for non-secure if this is a secure fetch.

PIE_Dirty enum

The Dirty bit if S2PIE is in use.

PIE_PIIIndex unsigned int

The S2PIIndex if S2PIE is in use, or 0xFFFF if not.

PXN bool

Privileged eXecute Never.

Protected enum

Is the descriptor producing an AssuredTranslation.

SH10 enum

The shareability.

XN bool

eXecute Never.

contiguous bool

Contiguous hint.

desckind enum

The kind of descriptor.

hwu_pbha unsigned int

Top four bits are appropriate CD.HWU, *bottom bits[62:59] of descriptor. Page Based Hardware Attributes: only valid on a bit-per-bit basis enabled by CD.HWU0/CD.HWU1**.

input_address unsigned int

Input address this lookup is for (after TBI normalisation).

mecid unsigned int

The MECID of the TTD fetch.

nG bool

not Global.

nT bool

SMMUv3.2: If true then do not cache this entry in such a way that it can cause a TLB conflict abort (F_TLB_CONFLICT). The entry must produce a consistent result.

ns enum

Non-Secure on the bus.

output_address unsigned int

Output address.

pa_address unsigned int

Physical address.

pas enum

The PAS of the TTD fetch.

ssd enum

SSD of the transaction.

ssd_ns enum

Security state of stream.

streamid unsigned int

The StreamID this is for.

substreamid unsigned int

The SubstreamID this is for.

trans_id unsigned int

Transaction id.

ttb_grain_stage_and_level unsigned int

Which TTB used in bits[31:24]. Which Grain size used in bit[23:16]:- 0 – 4 KiB 1 – 16 KiB 2 – 64 KiB 3 – 64 KiB ARMv8.2 where descriptor bits [15:12] for bits [51:48] of output address 4 – 4 KiB FEAT_LPA2 where address is {bits[9:8], bits[49:12]} 5 – 16 KiB FEAT_LPA2 where address is {bits[9:8], bits[49:14], 2'b00} 6 – 64 KiB ARMv8.2 where descriptor bits [15:12] are ignored Stage in bits[15:8]. Level in bits[7:0], level -1 is 0xF.

ptw_read_st1_table_long_descriptor

Page Table Walk (read). The descriptor that the PTW fetched is a Table, this decodes the fields. Fields:

AF enum

.

APTable enum

Remove permissions independently of subsequent descriptors.

NSTable bool

The next level table descriptor is forced to non-secure.

PXNTable bool

Force PXN independently of subsequent descriptors.

Protected enum

Is the descriptor capable of producing an AssuredTranslation.

TableAddress unsigned int

Address of the next table.

XNTable bool

Force XN independently of subsequent descriptors.

desckind enum

The kind of descriptor.

input_address unsigned int

Input address this lookup is for (after TBI normalisation).

mecid unsigned int

The MECID of the TTD fetch.

ns enum

Non-Secure on the bus.

pa_address unsigned int

Physical address.

pas enum

The PAS of the TTD fetch.

ssd enum

SSD of the transaction.

ssd_ns enum

Security state of stream.

streamid unsigned int

The StreamID this is for.

substreamid unsigned int

The SubstreamID this is for.

trans_id unsigned int

Transaction id.

ttb_grain_stage_and_level unsigned int

Which TTB used in bits[31:24]. Which Grain size used in bit[23:16]:- 0 – 4 KiB 1 – 16 KiB 2 – 64 KiB 3 – 64 KiB ARMv8.2 where descriptor bits [15:12] for bits [51:48] of output address 4 – 4 KiB FEAT_LPA2 where address is {bits[9:8], bits[49:12]} 5 – 16

KiB FEAT_LPA2 where address is {bits[9:8], bits[49:14], 2'b00} 6 – 64 KiB ARMv8.2 where descriptor bits [15:12] are ignored Stage in bits[15:8]. Level in bits[7:0], level -1 is 0xF.

ptw_read_st2_invalid_descriptor

Page Table Walk (read). The descriptor that the PTW fetched is invalid. Fields:

desckind enum

The kind of descriptor.

input_address unsigned int

Input address this lookup is for (after TBI normalisation).

mecid unsigned int

The MECID of the TTD fetch.

ns enum

Non-Secure on the bus.

pa_address unsigned int

Physical address.

pas enum

The PAS of the TTD fetch.

ssd enum

SSD of the transaction.

ssd_ns enum

Security state of stream.

streamid unsigned int

The StreamID this is for.

substreamid unsigned int

The SubstreamID this is for.

trans_id unsigned int

Transaction id.

ttb_grain_stage_and_level unsigned int

Which TTB used in bits[31:24]. Which Grain size used in bit[23:16]:- 0 – 4 KiB 1 – 16 KiB 2 – 64 KiB 3 – 64 KiB ARMv8.2 where descriptor bits [15:12] for bits [51:48] of output address 4 – 4 KiB FEAT_LPA2 where address is {bits[9:8], bits[49:12]} 5 – 16 KiB FEAT_LPA2 where address is {bits[9:8], bits[49:14], 2'b00} 6 – 64 KiB ARMv8.2 where descriptor bits [15:12] are ignored Stage in bits[15:8]. Level in bits[7:0], level -1 is 0xF.

ptw_read_st2_leaf_descriptor

Page Table Walk (read). The descriptor that the PTW fetched is a block or page and this is the decode. Fields:

AF bool

Access Flag.

AMEC enum

Use Primary or Alternative MECID.

AssuredOnly enum

The descriptor is marked as AssuredOnly.

DBM bool

Dirty Bit Modifier. This only has meaning if running in AArch64 mode.

HAP21 enum

The access permissions.

MemAttr3_0 enum

The memory attributes.

NS enum

Whether this descriptor forces NS.

PIE_Dirty enum

The Dirty bit if S2PIE is in use.

PIE_PIIIndex unsigned int

The S2PIIndex if S2PIE is in use, or 0xFFFF if not.

POE_POIndex unsigned int

The S2POIndex if S2POE is in use, or 0xFFFF if not.

SH10 enum

The shareability.

XN enum

eXecute Never.

contiguous bool

Contiguous hint.

desckind enum

The kind of descriptor.

input_address unsigned int

Input address this lookup is for (after TBI normalisation).

mecid unsigned int

The MECID of the TTD fetch.

nT bool

SMMUv3.2: If true then do not cache this entry in such a way that it can cause a TLB conflict abort (F_TLB_CONFLICT). The entry must produce a consistent result.

ns enum

Non-Secure on the bus.

output_address unsigned int

Output address.

pa_address unsigned int

Physical address.

pas enum

The PAS of the TTD fetch.

s2hwu_pbha unsigned int

Top four bits are STE.S2HWU, bottom bits[62:59] of descriptor. *Page Based Hardware Attributes: only valid on a bit-per-bit basis enabled by STE.S2HWU.*

ssd enum

SSD of the transaction.

ssd_ns enum

Security state of stream.

streamid unsigned int

The StreamID this is for.

substreamid unsigned int

The SubstreamID this is for.

trans_id unsigned int

Transaction id.

ttb_grain_stage_and_level unsigned int

Which TTB used in bits[31:24]. Which Grain size used in bit[23:16]:- 0 – 4 KiB 1 – 16 KiB 2 – 64 KiB 3 – 64 KiB ARMv8.2 where descriptor bits [15:12] for bits [51:48] of output address 4 – 4 KiB FEAT_LPA2 where address is {bits[9:8], bits[49:12]} 5 – 16 KiB FEAT_LPA2 where address is {bits[9:8], bits[49:14], 2'b00} 6 – 64 KiB ARMv8.2 where descriptor bits [15:12] are ignored Stage in bits[15:8]. Level in bits[7:0], level -1 is 0xF.

ptw_read_st2_table_descriptor

Page Table Walk (read). The descriptor that the PTW fetched is a Table, this decodes the fields. Fields:

AF enum

.

APTable enum

Remove permissions independently of subsequent descriptors.

PXNTable bool

Force PXN independently of subsequent descriptors.

TableAddress unsigned int

Address of the next table.

XNTable bool

Force XN independently of subsequent descriptors.

desckind enum

The kind of descriptor.

input_address unsigned int

Input address this lookup is for (after TBI normalisation).

mecid unsigned int

The MECID of the TTD fetch.

ns enum

Non-Secure on the bus.

pa_address unsigned int

Physical address.

pas enum

The PAS of the TTD fetch.

ssd enum

SSD of the transaction.

ssd_ns enum

Security state of stream.

streamid unsigned int

The StreamID this is for.

substreamid unsigned int

The SubstreamID this is for.

trans_id unsigned int

Transaction id.

ttb_grain_stage_and_level unsigned int

Which TTB used in bits[31:24]. Which Grain size used in bit[23:16]:- 0 – 4 KiB 1 – 16 KiB 2 – 64 KiB 3 – 64 KiB ARMv8.2 where descriptor bits [15:12] for bits [51:48] of output address 4 – 4 KiB FEAT_LPA2 where address is {bits[9:8], bits[49:12]} 5 – 16 KiB FEAT_LPA2 where address is {bits[9:8], bits[49:14], 2'b00} 6 – 64 KiB ARMv8.2 where descriptor bits [15:12] are ignored Stage in bits[15:8]. Level in bits[7:0], level -1 is 0xF.

raw_register_end_read

The raw register read transaction. This is the transaction as directed to the register port.
Fields:

address unsigned int

The address of the transaction.

id unsigned int

Id of this transaction.

ok bool

Was the access OK? The bus response will always be OK, but is the register access reasonable.

pas enum

The PAS of the transaction.

read_data unsigned int

The data read.

raw_register_end_write

The raw register write transaction. This is the transaction as directed to the register port.
Fields:

address unsigned int

The address of the transaction.

id unsigned int

Id of this transaction.

ok bool

Was the access OK? The bus response will always be OK, but is the register access reasonable.

pas enum

The PAS of the transaction.

raw_register_start_read

The raw register read transaction. This is the transaction as directed to the register port.
Fields:

address unsigned int

Address of transaction.

id unsigned int

Id of this transaction. Top bit is set if is a debug transaction.

ns enum

Is the transaction non-secure?.

number_of_beats unsigned int

Number of beats.

pas enum

The PAS of the transaction.

size_of_beat_in_bytes unsigned int

Size of each beat in bytes.

raw_register_start_write

The raw register write transaction. This is the transaction as directed to the register port.
Fields:

address unsigned int

Address of transaction.

id unsigned int

Id of this transaction. Top bit is set if is a debug transaction.

ns enum

Is the transaction non-secure?.

number_of_beats unsigned int

Number of beats.

pas enum

The PAS of the transaction.

size_of_beat_in_bytes unsigned int

Size of each beat in bytes.

write_data unsigned int

The data to write.

register_disallowed_read_string

A text representation of the read of a register that was disallowed. Fields:

out string

The text description of the register value read.

register_disallowed_write_string

A text representation of the write of a register write that was disallowed. Fields:

in string

The text description of the register value written.

register_read_reserved

A text representation of an access to a register address that is reserved. Fields:

in string

The text description of the register value.

register_read_string

A text representation of the read of a register. Fields:

out string

The text description of the register value read.

register_write_reserved

A text representation of an access to a register address that is reserved or a write.to a **RES0** field in a register. Fields:

in string

The text description of the register value.

register_write_string

A text representation of the write of a register. Fields:

in string

The text description of the register value written.

reset_in

The reset signal. Fields:

value bool

The value of the signal.

rl_priq_auto_response

A PRIQ auto response is generated. Fields:

prgindex unsigned int

The PRG Index of the response.

resp enum

The response to send.

streamid unsigned int

The StreamID of the response.

substreamid unsigned int

The PASID/SubstreamID attached to the response, or ~0u if there is none.

trans_id unsigned int

The ID of the PPR that is being lost.

why enum

Why the PPR is being lost.

rl_priq_lost_ppr

The PRIQ is not reporting a PPR into the queue. It may or may not generate an auto-response. Fields:

is_stop_marker bool

The PPR being lost is a Stop Marker.

resp enum

What, if any, auto-response will be.

trans_id unsigned int

The ID of the PPR that is being lost.

why enum

Why the PPR is being lost.

rl_priq_received

A PRIQ Request has been received and is queued waiting processing. This does not mean that it has been written to the PRIQ but has been placed in a pending queue awaiting a decision about what to do about it. Fields:

L bool

The Last in the Page Request Group (PRG).

R bool

Read permission is requested.

W bool

Read permission is requested.

prgindex unsigned int

The Page Request Group (PRG) index that is being used to identify this request to the EndPoint.

stop_marker bool

If LRW == 0b100 then this indicates this is a stop marker.

streamid unsigned int

The StreamID this request is associated with.

substreamid unsigned int

The SubstreamID (PASID) this request is associated with, of ~0u if no SubstreamID.

substreamid_P bool

If has a SubstreamID (!= ~0u) then Privileged Mode Requested.

substreamid_X bool

If has a SubstreamID (!= ~0u) then Execution Requested (R must also be 1).

trans_id unsigned int

The transaction ID that identifies this request.

untranslated_address unsigned int

The Untranslated Address that this is requesting.

s_cmd_sync_completed_irq

“Secure” CMD_SYNC completed its IRQ action. Fields:

cmd_id unsigned int

The ID of the command being issued.

cons_incl_wrap unsigned int

CONS pointer of CMD_SYNC.

is_synthetic bool

This is a synthetic sync. The cons_incl_wrap is 0xFFFF'ffff.

s_cmd_sync_completed_no_action

Secure CMD_SYNC completed, there was no CMD_SYNC action (SIG_NONE). Fields:

cmd_id unsigned int

The ID of the command being issued.

cons_incl_wrap unsigned int

CONS pointer of CMD_SYNC.

is_synthetic bool

This is a synthetic sync. The cons_incl_wrap is 0xFFFF'ffff.

s_cmd_sync_completed_sev

Secure CMD_SYNC completed its action of SIG_SEV. Fields:

cmd_id unsigned int

The ID of the command being issued.

cons_incl_wrap unsigned int

CONS pointer of CMD_SYNC.

is_synthetic bool

This is a synthetic sync. The cons_incl_wrap is 0xFFFF'ffff.

s_cmd_sync_completed_with_error

Secure CMD_SYNC completed, however, there was an error associated with the completion of the CMD_SYNC. Fields:

error enum

The error associated with the CMD_SYNC.

cmd_id unsigned int

The ID of the command being issued.

cons_incl_wrap unsigned int

CONS pointer of CMD_SYNC.

is_synthetic bool

This is a synthetic sync. The cons_incl_wrap is 0xFFFF'ffff.

s_cmd_sync_issued

Secure CMD_SYNC has now completed issuing and we now know if something is hazarding the CMD_SYNC from completing. Fields:

cmd_id unsigned int

The ID of the command being issued.

cons_incl_wrap unsigned int

CONS pointer of CMD_SYNC.

hazarded bool

The CMD_SYNC is currently hazarded from completing.

is_synthetic bool

This is a synthetic sync. The cons_incl_wrap is 0xFFFF'ffff.

s_cmd_sync_issuing

Secure CMD_SYNC is starting to issue, we are going to start accounting for which things have to be part of the prior set of things that have to complete before the CMD_SYNC can complete. Fields:

cmd_id unsigned int

The ID of the command being issued.

cons_incl_wrap unsigned int

CONS pointer of CMD_SYNC.

is_synthetic bool

This is a synthetic sync. The cons_incl_wrap is 0xFFFF'ffff.

s_cmd_sync_starting_completion_action_irq

Secure CMD_SYNC starting IRQ completion action. Fields:

cmd_id unsigned int

The ID of the command being issued.

cons_incl_wrap unsigned int

CONS pointer of CMD_SYNC.

is_synthetic bool

This is a synthetic sync. The cons_incl_wrap is 0xFFFF'ffff.

msiaddress unsigned int

The MSI address to use (or 0 if not sending an MSI).

msidata unsigned int

The MSI data to send.

s_cmd_sync_starting_completion_action_sev

Secure CMD_SYNC starting its completion action of SIG_SEV. Fields:

cmd_id unsigned int

The ID of the command being issued.

cons_incl_wrap unsigned int

CONS pointer of CMD_SYNC.

is_synthetic bool

This is a synthetic sync. The cons_incl_wrap is 0xFFFF'ffff.

s_cmdq_cmd_sync_error

The CMD_SYNC can fault if an ATC Invalidate timeout occurs. The timeout will be attached to the next CMD_SYNC on the queue on which the invalidate was launched. Fields:

cerror enum

The CERROR to attach.

cmd_queue unsigned int

The top bits indicate the CMDQ id. These bits are attached to all cmd_ids generated for this queue. The bits [63:44] contain the queue ID. [63:62] 0: s-ssd, 1: reserved, 2: ns-ssd. 3: rl-ssd [61] is ECMDQ [59:52] QCP index [51:44] ECMDQ index in QCP.

herror enum

The HERROR to attach, or HERROR_NONE if we are attaching a CERROR.

s_cmdq_cmd_sync_error_delivered

The CMD_SYNC can fault if an ATC Invalidate timeout occurs. This is the CMD_SYNC delivering that fault. Fields:

cerror enum

The CERROR to attach.

cmd_id unsigned int

Command ID of the CMD_SYNC.

s_cmdq_ecmdq_enack

Trace the changes to ENACK. Fields:

cerror enum

The error being exposed, if any.

cmd_queue unsigned int

The top bits indicate the CMDQ id. These bits are attached to all cmd_ids generated for this queue. The bits [63:44] contain the queue ID. [63:62] 0: s-ssd, 1: reserved, 2: ns-ssd. 3: rl-ssd [61] is ECMDQ [59:52] QCP index [51:44] ECMDQ index in QCP.

enack bool

The value of ENACK.

forced_error bool

The queue was enabled with ERR != ERRACK and so the implementation chooses to trigger the error reporting mechanism.

s_cmdq_issue

A command is actually being executed. Fields:

cmd_id unsigned int

Command ID of the command being executed.

cons unsigned int

CONS of the command.

what string

What is being executed.

s_cmdq_pointers_state

The command queue pointers. Fields:

cmd_queue unsigned int

The top bits indicate the CMDQ id. These bits are attached to all cmd_ids generated for this queue. The bits [63:44] contain the queue ID. [63:62] 0: s-ssd, 1: reserved, 2: ns-ssd. 3: rl-ssd [61] is ECMDQ [59:52] QCP index [51:44] ECMDQ index in QCP.

disable_fetch bool

The model has decided that as you programmed PROD/CONS inconsistently that it would not fetch as it has no confidence as to which are commands and which are not.

q_issuable_and_issued_cons unsigned int

The CONS pointer of the region that is issuable and/or issued (including wrap).

q_issuable_and_issued_prod unsigned int

The PROD pointer of the region that is issuable and/or issued (including wrap).

q_programmer_cons unsigned int

The CONS pointer as seen by the programmer (including wrap).

q_programmer_prod unsigned int

The PROD pointer as seen by the programmer (including wrap).

q_unfetched_cons unsigned int

The CONS pointer of the region unfetched (including wrap).

q_unfetched_prod unsigned int

The PROD pointer of the region unfetched (including wrap).

s_cmdq_state

A description of what the CMDQ can do now. Fields:

can_cerror bool

There is a CERROR that can be recognised.

can_cmd_sync bool

There is a CMD_SYNC that can be recognised.

can_fetch bool

The SMMU can fetch.

can_herror bool

There is an HERROR that can be recognised.

can_issue bool

There are fetched commands that are eligible to be issued.

can_update_cons bool

The programmer view of CMDQ_CONS can be moved to say some have been consumed.

cmd_queue unsigned int

The top bits indicate the CMDQ id. These bits are attached to all cmd_ids generated for this queue. The bits [63:44] contain the queue ID. [63:62] 0: s-ssd, 1: reserved, 2: ns-ssd, 3: rl-ssd [61] is ECMDQ [59:52] QCP index [51:44] ECMDQ index in QCP.

cmd_sync enum

Why a CMD_SYNC cannot be issued.

cmdqen bool

CMDQEN, when enabled then the CMDQ can fetch commands.

cmdqen_1_to_0_unacknowledged bool

CMDQEN has been set to 0, but we have yet to acknowledge it.

current-real-cerror enum

The current CERROR that would be visible to the programmer if there was an error.

fetch-disabled_due_to_programmer_error bool

The PROD/CONS pointer was inconsistent and the model decided to stop processing commands.

fetch-number_of_unfetched_commands unsigned int

The number of commands that have yet to be fetched from memory, or ~0u if CMDQEN == 0.

fetch-state enum

The state of the fetch state machine.

fetch-too_many_commands_outstanding_so_not_fetching_any_more bool

If there are too many issued commands, or pending to be issued then we won't fetch any more.

issue-number_of_issuable_commands unsigned int

The number of commands that have been fetched but not yet issued.

issue-number_of_unissuable_commands unsigned int

This might include commands behind an illegal command, or behind a CMD_SYNC. Under some circumstances this can include a command representing an external abort.

issue-state enum

The internal issued state machine state.

outstanding_fetches_since_reset unsigned int

The number of outstanding fetches.

synthetic_sync_can_be_completed bool

The synthetic sync that is outstanding can be completed.

waiting_for_synthetic_sync bool

There is a synthetic sync that is outstanding.

s_eventq_adding_to_cmd_sync_prior_set

The event created by the specified transaction id must become visible to the programmer before the CMD_SYNC can complete. Fields:

trans_id unsigned int

The transaction id that caused the event.

s_eventq_auto_retry_stalled_transaction

A transaction that was stalled but not yet reported on the Event Queue was retried. The retry can happen if the transaction didn't write because it was de-duplicated behind a reported stalling transaction and that stalling transaction was CMD_RESUMEd, or if the queue was unwritable and the queue is now writable. Fields:

trans_id unsigned int

Transaction id of the auto-retried transaction.

why enum

Why the transaction retried.

s_eventq_cmd_sync_unhazarded

The CMD_SYNC has been unhazarded as the prior set is empty and as much of it as required is visible in the programmers' view. Fields:

prod_incl_wrap unsigned int

The prod index including wrap bit that was required to become visible.

prog_prod_incl_wrap unsigned int

The current programmer-visible prod including wrap bit.

s_eventq_eventqen_and_eventqenack

Trace the values of EVENTQEN and EVENTQENACK as they change. Fields:

about enum

Some more information about the queue state.

cons_incl_wrap unsigned int

The CONS pointer (including wrap bits).

eventqen bool

EVENTQEN.

eventqenack bool

EVENTQENACK.

prod_incl_wrap unsigned int

The PROD pointer (including wrap bits).

s_eventq_external_abort

A particular eventq record got an external abort. The event will subsequently appear in the ns/s_eventq_lost_event_records trace. Fields:

prod unsigned int

The prod pointer (excluding wrap bit) of the event record that aborted.

trans_id unsigned int

The transaction id that produced the event record that aborted.

s_eventq_lost_event_records

Event records that are lost and never record appear on this trace. Fields:

dword0 unsigned int

Double word 0 of event.

dword1 unsigned int

Double word 1 of event.

dword2 unsigned int

Double word 2 of event.

dword3 unsigned int

Double word 3 of event.

event enum

Event code that was thrown away.

prod unsigned int

The PROD index of the event if appropriate or ~0u otherwise.

reason enum

The reason the record was lost.

trans_id unsigned int

The transaction id that this corresponds to.

s_eventq_overflow_acknowledged

When events are lost then they appear on the eventq_lost_event_records_* trace. This will activate the overflow flag in the corresponding SMMU_s_EVENTQ_PROD register. Software can then acknowledge that flag using the overflow acknowledge flag in SMMU_s_EVENTQ_CONS. Fields:

bad bool

If true then the programmer has attempted to acknowledge an overflow condition that didn't exist.

count_of_events_lost_due_to_overflow unsigned int

This is the number of events that were lost because the event queue was full. This does not include any lost because of an EVENTQ_ABT_ERR.

s_eventq_pending_event_records

Event records that are pending to be dealt with by the eventq state machine. A specific event can be traced multiple times if it becomes part of the prior set of transactions of a CMD_SYNC that has to complete before the CMD_SYNC is allowed to be completed. Fields:

dword0 unsigned int

Double word 0 of event.

dword1 unsigned int

Double word 1 of event.

dword2 unsigned int

Double word 2 of event.

dword3 unsigned int

Double word 3 of event.

event enum

Event code.

is_prior_set bool

The transaction forms part of the prior set of a CMD_SYNC that have to become visible (or thrown away if necessary) before the CMD_SYNC can complete.

trans_id unsigned int

The transaction id that this corresponds to.

s_eventq_prod_cons_updated

This records when the programmer visible PROD/CONS pointer is updated. More can be visible in memory than are advertised in the programmer-visible PROD pointer. This also traces the latest PROD that could have been advertised. Fields:

mem_prod unsigned int

The PROD pointer value for all records that are visible in the memory.

num_events_available unsigned int

The number of events available to the programmer.

prog_cons unsigned int

The programmer-visible CONS pointer.

prog_prod unsigned int

The programmer-visible PROD pointer.

what enum

What changed?.

s_eventq_raising_eventq_abt_err

EVENTQ_ABT_ERR can now be raised.

s_eventq_removing_from_cmd_sync_prior_set

The event captured by a CMD_SYNC and created by the specified transaction id is now in memory. Fields:

prod unsigned int

The prod index (no wrap bit) that must become visible to the programmer to see this record. If not head_of_line then the record will not become visible and this field is meaningless.

reason enum

The reason the event was removed.

trans_id unsigned int

The transaction id that caused the event.

s_eventq_stashing_unreported_stalled_transaction_for_auto_retry

A stall event record would have been lost, instead it is stashed away to auto-retry when it can. Fields:

dword0 unsigned int

Double word 0 of event.

dword1 unsigned int

Double word 1 of event.

dword2 unsigned int

Double word 2 of event.

dword3 unsigned int

Double word 3 of event.

event enum

Event code that was thrown away.

reason enum

The reason the record was lost.

trans_id unsigned int

The transaction id that this corresponds to.

s_eventq_write_event

An event is committed to being written. Fields:

dword0 unsigned int

Double word 0 of event.

dword1 unsigned int

Double word 1 of event.

dword2 unsigned int

Double word 2 of event.

dword3 unsigned int

Double word 3 of event.

event enum

Event code that was thrown away.

prod unsigned int

The PROD pointer (without wrap) that this event will be written to.

stall_event bool

Describes an event for a stalled transaction.

trans_id unsigned int

Transaction ID that generated this event.

s_eventq_write_event_visible_in_memory

An event is now visible in memory. Fields:

abort bool

The write aborted.

dword0 unsigned int

Double word 0 of event.

dword1 unsigned int

Double word 1 of event.

dword2 unsigned int

Double word 2 of event.

dword3 unsigned int

Double word 3 of event.

event enum

Event code that was thrown away.

prod unsigned int

The PROD pointer (without wrap) that this event will be written to.

stall_event bool

Describes an event for a stalled transaction.

trans_id unsigned int

Transaction ID that generated this event.

s_gerror_inverted

A GERROR bit was attempted to be inverted to record a fault. Fields:

already_different_to_gerrorn bool

True if the bit was already different to the corresponding bit in GERRORN and so it was not inverted.

gerrorn unsigned int

GERRORN register value.

interrupt_potentially_sent bool

An interrupt is potentially sent, depending on IRQEN.
GERROR(N).MSI_GERROR_ABT_ERR does not send an interrupt.

new_gerror unsigned int

GERROR register value after any inversion.

which enum

Which bit was attempted to be inverted.

s_gerrorn_acknowledge

A GERROR was acknowledged by SW. Fields:

which enum

Which GERROR was acknowledged.

s_priq_auto_response

A PRIQ auto response is generated. Fields:

prgindex unsigned int

The PRG Index of the response.

resp enum

The response to send.

streamid unsigned int

The StreamID of the response.

substreamid unsigned int

The PASID/SubstreamID attached to the response, or ~0u if there is none.

trans_id unsigned int

The ID of the PPR that is being lost.

why enum

Why the PPR is being lost.

s_priq_lost_ppr

The PRIQ is not reporting a PPR into the queue. It may or may not generate an auto-response. Fields:

is_stop_marker bool

The PPR being lost is a Stop Marker.

resp enum

What, if any, auto-response will be.

trans_id unsigned int

The ID of the PPR that is being lost.

why enum

Why the PPR is being lost.

s_priq_received

A PRIQ Request has been received and is queued waiting processing. This does not mean that it has been written to the PRIQ but has been placed in a pending queue awaiting a decision about what to do about it. Fields:

L bool

The Last in the Page Request Group (PRG).

R bool

Read permission is requested.

W bool

Read permission is requested.

prgindex unsigned int

The Page Request Group (PRG) index that is being used to identify this request to the EndPoint.

stop_marker bool

If LRW == 0b100 then this indicates this is a stop marker.

streamid unsigned int

The StreamID this request is associated with.

substreamid unsigned int

The SubstreamID (PASID) this request is associated with, of ~0u if no SubstreamID.

substreamid_P bool

If has a SubstreamID (!= ~0u) then Privileged Mode Requested.

substreamid_X bool

If has a SubstreamID (!= ~0u) then Execution Requested (R must also be 1).

trans_id unsigned int

The transaction ID that identifies this request.

untranslated_address unsigned int

The Untranslated Address that this is requesting.

sev

Send a SEV. Fields:

why enum

Why the SEV was generated.

smmu_atc_inv

The CMD_ATC_INV command is sent. Fields:

address unsigned int

Untranslated Address to invalidate.

cmd_id unsigned int

Command id.

cons unsigned int

CONS of the command.

global bool

Global flag.

log2_size_in_bytes unsigned int

Log 2 of the size in bytes of the region to invalidate.

ns_number_in_flight unsigned int

Number of ATC invalidates in flight issued from the Non-secure CMDQ.

ns_number_waiting_to_go_out unsigned int

The total number of ATC invalidates waiting to be pushed out from the Non-secure CMDQ.

rl_number_in_flight unsigned int

Number of ATC invalidates in flight issued from the Realm CMDQ.

rl_number_waiting_to_go_out unsigned int

The total number of ATC invalidates waiting to be pushed out from the Realm CMDQ.

s_number_in_flight unsigned int

Number of ATC invalidates in flight issued from the Secure CMDQ.

s_number_waiting_to_go_out unsigned int

The total number of ATC invalidates waiting to be pushed out from the Secure CMDQ.

ssd_of_queue enum

The SSD of the CMDQ.

ssd_of_streamid enum

The SSD of the StreamID.

streamid unsigned int

StreamID in the invalidate command.

substreamid unsigned int

SubstreamID or ~0u if SSValid == false.

smmu_atc_inv_completed

The CMD_ATC_INV command completed. Fields:

cmd_id unsigned int

Command id, if top-bit is set then was issued from the Non-secure CMDQ.

ns_number_in_flight unsigned int

Number of ATC invalidates in flight now this one has completed, that were issued by the Non-secure CMDQ.

ns_number_waiting_to_go_out unsigned int

The total number of ATC invalidates waiting to be pushed out from the Non-secure CMDQ.

response enum

The response.

rl_number_in_flight unsigned int

Number of ATC invalidates in flight now this one has completed, that were issued by the Realm CMDQ.

rl_number_waiting_to_go_out unsigned int

The total number of ATC invalidates waiting to be pushed out from the Realm CMDQ.

s_number_in_flight unsigned int

Number of ATC invalidates in flight now this one has completed, that were issued by the Secure CMDQ.

s_number_waiting_to_go_out unsigned int

The total number of ATC invalidates waiting to be pushed out from the Secure CMDQ.

ssd_of_cmdq enum

The SSD of the CMDQ.

ssd_of_streamid enum

The SSD of the StreamID.

smmu_atc_inv_end

An ATC invalidate request has been started. Fields:

PASID_global bool

If has a SubstreamID (PASID) then if it is 'global' or not.

address unsigned int

The Untranslated Address to use.

log2_size_in_bytes unsigned int

Log2 of the size of the region in bytes to invalidate.

response enum

The response to the ATC invalidate.

ssd enum

The SSD of the stream.

streamid unsigned int

The StreamID to use.

substreamid unsigned int

The SubstreamID to use (or ~0u if not used). This corresponds to the PASID.

smmu_atc_inv_start

An ATC invalidate request has been started. Fields:

PASID_global bool

If has a SubstreamID (PASID) then if it is 'global' or not.

address unsigned int

The Untranslated Address to use.

log2_size_in_bytes unsigned int

Log2 of the size of the region in bytes to invalidate.

ssd enum

The SSD of the stream.

streamid unsigned int

The StreamID to use.

substreamid unsigned int

The SubstreamID to use (or ~0u if not used). This corresponds to the PASID.

smmu_ats_initial

This is the initial ATS request. Fields:

XT bool

The XT bit.

ia unsigned int

Input address.

max_number_of_replies unsigned int

The maximum number of replies allowed to return.

no_write bool

The NW (no write flag) of the ATS request. If clear then the requester is going to do a write.

pasid_execute_requested bool

If there is a PASID (substreamid != ~0u) then this represents the requester asking for execution permissions.

pasid_privileged_mode_requested bool

If there is a PASID (substreamid != ~0u) then this represents the requester asking for the privileged mode's permissions.

protected_mode bool

Is the ATS Request protected-mode?.

source_cxl bool

Does the ATS request have Source.CXL set?.

ssd enum

The SSD of the ATS request.

ssd_ns bool

Incoming SSD is non-secure.

streamid unsigned int

StreamID of the ATS request.

substreamid unsigned int

SubstreamID (which is identical to the PASID) of the ATS request. If no PASID-prefix is sent then this is ~0u.

tbu unsigned int

Translation Buffer Unit number.

smmu_ats_reply_failure

This is an ATS reply indicating failure. Fields:

event enum

Equivalent event number that would have been generated for an equivalent ordinary transaction.

failure enum

What is the failure response code?.

state enum

The transaction state of the successfully ATS request.

smmu_ats_reply_success

This is an ATS reply, typically the SMMU will only return a single response, even if the requester indicated it could accept more replies. NOTE that the SMMU responds with 'success' in some cases when a fault is encountered and RW==0. Fields:

N bool

Non-snooped access. If one then the requester must clear the NoSnoop bit on transactions, unless otherwise enabled in a Function-specific manner.

P bool

Privileged mode. These permissions related to privileged mode.

RWX enum

Read/Write/Execute.

U bool

Untranslated access. If one, and RW !=0 then use UntranslatedAccesses for the allowed accesses by RW(X).

cxl_io bool

The CXL.io response.

inner enum

The inner cacheability attributes to use for TranslatedAccesses.

input_address unsigned int

Input address of the ATS request.

instcfg enum

The STE.INSTCFG field.

outer enum

The outer cacheability attributes to use for TranslatedAccesses.

pas enum

The PAS this mapping corresponds to. This holds the same information as the TE bit for realm streams.

privcfg enum

The STE.PRIVCFG field.

shareability unsigned int

The shareability to use for TranslatedAccesses.

size unsigned int

The size of the region covered by this translation.

state enum

The transaction state of the successfully ATS request.

translated_address unsigned int

If RW!=0 && U != 0, then the Translated Address that a TranslatedAccess can be made with.

smmu_axi_stream_msi

An SMMU generated MSI is directly sent through the axi_stream_msi_m port, typically connected to the GIC port axi_stream_msi_s. Fields:

TDEST unsigned int

Routing information for the data stream, typically identifying the GIC.

TID unsigned int

Data stream identifier for the SMMU.

axi_stream_msi_addr_to_match unsigned int

Current address to match for SMMU-originated MSIs to send out of the axi_stream_msi_m port.

data unsigned int

The MSI sent.

smmu_final_transaction

This is the transaction group request to remap has completed one set of remapping. For ordinary transactions, this represents a bundle of transactions with the same attributes but different addresses within a certain range around the address. If it stalls then it will report through this trace source, stall (stag_if_stalling != ~Ou) and when resume will issue another smmu_initial_transaction as it undergoes remapping again. This trace source can also represent part of the process for ATOS/ATS or finding STE.PPAR for PRI requests that need to be auto-responded to. Fields:

axmmuflow enum

The AxMMUFLOW for this transaction group. stallable – allow the transaction to be stalled if configured. translated-access – is a PCIe Translated Access non-stallable – do not stall transaction even if configured. transfault-flow – do not report for Translation Related Faults, but tell client device to use a page request mechanism instead (for example PRI).

begin_input_address_range unsigned int

The start of the input address range that is size_of_region_in_bytes.

begin_ipa_range unsigned int

The start of the IPA range that is of size_of_region_in_bytes.

begin_output_address_range unsigned int

The start of the output address range that is of size_of_region_in_bytes.

cmo_point enum

The point associated with the CMO, if applicable.

input_address unsigned int

The input address of the transaction group.

input_inner enum

Inner cacheability for the input attributes.

input_outer enum

Outer cacheability for the input attributes.

input_prot enum

The instruction/data, bus-ns/s and privileged/user marking.

input_sh enum

Shareability for the input attributes.

ipa_address unsigned int

The IPA of the transaction.

kind enum

The kind of access this transaction group that this represents.

mecid unsigned int

The MECID of the transaction.

mpam_pmg_and_partid unsigned int

Bits[23:16] are the MPAM PMG. Bits[15:0] are the physical PARTID.

mpam_sp enum

The MPAM_SP of the transaction.

operation enum

The kind of operation that this represents.

output_address unsigned int

The input address of the transaction group.

output_inner enum

Inner cacheability for the output attributes.

output_outer enum

Outer cacheability for the output attributes.

output_prot enum

The instruction/data, bus-ns/s and privileged/user marking.

output_sh enum

Shareability for the output attributes.

output_vmid unsigned int

The output VMID/GBPA.IMPDEF or ~0u if not valid.

size_of_region_in_bytes unsigned int

An imp def size of region for which this translation is valid for.

ssd enum

The SSD of the transaction.

ssd_ns enum

The security state of the transaction.

stag_if_stalling unsigned int

This is the STAG used by the transaction if it is going to stall. It is ~0u if it is not going to stall.

state enum

The final transaction state.

streamid unsigned int

The StreamID of the transaction. ~0ull if NoStreamID.

substreamid unsigned int

The SubstreamID of the transaction or 0xFFFFffff if no SubstreamID.

tbu unsigned int

Translation Buffer Unit number.

trans_id unsigned int

Transaction id. Top bit set if debug transaction.

smmu_gpt_tlb_invalidate

A GPT TLB invalidate has been initiated. Fields:

address unsigned int

The address as it appears in the operation.

encoded_size unsigned int

The size as it is encoded in the operation.

kind enum

The kind of operation this is.

pgs_in_bytes unsigned int

The PGS size in bytes.

size_in_bytes unsigned int

For range operations, the size as it appears in the operation.

source enum

Where the TLBI came from.

state enum

Is the operation well formed.

trans_id unsigned int

The transaction id of this invalidate.

smmu_gpt_tlb_invalidate_complete

The GPT TLB invalidate completed. Fields:

source enum

Where the TLBI came from.

trans_id unsigned int

The transaction id of this invalidate.

smmu_initial_transaction

This is the transaction group request to remap is going to start one set of remapping. For ordinary transactions, this represents a bundle of transactions with the same attributes but different addresses within a certain range around the address. A stalling transaction will report through this trace source when it unstalls. This trace source can also represent part of the process for ATOS/ATS or finding STE.PPAR for PRI requests that need to be auto-responded to. Fields:

XT_and_output_pas_checking enum

The XT bit for PCIe Transactions. This specifies the requested check on the output PAS that the device asked for.

axmmuflow enum

The AxMMUFLOW for this transaction group. storable – allow the transaction to be stalled if configured. translated-access – is a PCIe Translated Access non-storable – do not stall transaction even if configured. transfault-flow – do not report for Translation Related Faults, but tell client device to use a page request mechanism instead (for example PRI).

cmo_point enum

The point associated with the CMO, if applicable.

dcmdq_qcp_and_index unsigned int

If this is a DCMDQ fetch being translated then this field indicates the DCMDQ QCP index in bits [23:8] and the index in the page in bits [7:0]. Otherwise this is 0xFFFFffff.

input_address unsigned int

The input address of the transaction group.

input_inner enum

Inner cacheability for the input attributes.

input_outer enum

Outer cacheability for the input attributes.

input_prot enum

The instruction/data, bus-ns/s and privileged/user marking.

input_sh enum

Shareability for the input attributes.

kind enum

The kind of access this transaction group that this represents.

mecid unsigned int

The incoming MECID for NoStreamID transactions. ~0u for all other transactions.

mpam_partid unsigned int

The MPAM_PARTID for NoStreamID transactions. ~0u for all other transactions.

mpam_pmg unsigned int

The MPAM_PMG for NoStreamID transactions. ~0u for all other transactions.

mpam_sp enum

The MPAM_SP for NoStreamID transactions. ~0u for all other transactions.

operation enum

The kind of operation that this represents.

protected_mode enum

The PM bit.

ssd enum

The SSD of the transaction.

ssd_ns enum

The security state of the transaction.

streamid unsigned int

The StreamID of the transaction. ~0ull means NoStreamID.

substreamid unsigned int

The SubstreamID of the transaction or 0xFFFFffff if no SubstreamID.

tbu unsigned int

Translation Buffer Unit number.

trans_id unsigned int

Transaction id. Top bit set if debug transaction.

smmu_normalize_input_transaction

If the input transaction is normalized before being processed then this traceevent will fire. This is usually dependent on parameters of the implementation. Fields:

incoming_inner enum

The inner cacheability attributes.

incoming_is_instruction bool

The incoming transaction is marked as 'instruction'.

incoming_is_privileged bool

The incoming transaction is marked as 'privileged'.

incoming_outer enum

The outer cacheability attributes.

incoming_pas enum

The PAS of the incoming transaction.

incoming_shareability unsigned int

The incoming shareability.

normalized_inner enum

The normalized inner cacheability attributes.

normalized_is_instruction bool

The incoming transaction is marked as 'instruction'.

normalized_is_privileged bool

The incoming transaction is marked as 'privileged'.

normalized_outer enum

The normalized outer cacheability attributes.

normalized_pas enum

The PAS this mapping corresponds to.

normalized_shareability unsigned int

The normalized shareability.

ssd enum

The SSD of the transaction.

streamid unsigned int

The StreamID of the transaction. ~0ull means NoStreamID.

substreamid unsigned int

The SubstreamID of the transaction or 0xFFFFFFFF if no SubstreamID.

trans_id unsigned int

ID of the original transaction.

smmu_poison_tw_data

Poison data has been returned to a table walk transaction. Fields:

bitmap_of_poison unsigned int

The bitmap of which beats of the transaction where poisoned.

is_cas bool

True if this is a compare-and-swap operation.

number_of_64bit_beats unsigned int

Number of beats this transaction fetched.

paddress unsigned int

Physical address of the table walk transaction.

pas enum

The PAS of the bus transaction.

ras_group_id unsigned int

If non-~0u then is the RAS group that we wish to attach the error to. Otherwise will use the IMP DEF policy.

ras_record_index_in_group unsigned int

If non-~0u then is the RAS record index in the group that we wish to attach the error to. Otherwise will use the IMP DEF policy.

what enum

What table walk was being performed.

smmu_pri_resp

The CMD_PRI_RESP command is queued to be sent to the PCIe system. Fields:

auto_response_trans_id unsigned int

trans_id of PRI request we are auto-responding to, or ~0ull if not valid.

cmd_id unsigned int

Command id, or ~0ull if not valid.

cons unsigned int

CONS of the command. ~0u if an auto-response.

prgindex unsigned int

PRG Index.

response enum

If has a SubstreamID (PASID) then if it is 'global' or not.

ssd enum

The SSD of the StreamID this corresponds to.

streamid unsigned int

StreamID in the invalidate command.

substreamid unsigned int

SubstreamID or ~0u if SSValid == false.

smmu_pri_resp_nop

The CMD_PRI_RESP command was NOPped. Fields:

cmd_id unsigned int

Command id.

cons unsigned int

CONS of the command.

prgindex unsigned int

PRG Index.

response enum

If has a SubstreamID (PASID) then if it is 'global' or not.

ssd enum

The SSD of the StreamID this corresponds to.

streamid unsigned int

StreamID in the invalidate command.

substreamid unsigned int

SubstreamID or ~0u if SSValid == false.

why enum

Why the CMD_PRI_RESP was NOPped.

smmu_priq_resp_fake_return

A PRIQ Response is posted to the PCIe subsystem and so has no acknowledgement that it is received. However, in the model then we artificially know when the the PRIQ Response has been delivered to the PCIe subsystem, even if the ATC has not yet acted on it. Fields:

prgindex unsigned int

The PRG Index this request corresponds to.

ssd enum

SSD of response.

streamid unsigned int

The StreamID to use.

substreamid unsigned int

The SubstreamID to use (or ~0u if not used). This corresponds to the PASID.

what enum

What happened.

smmu_priq_resp_start

A PRIQ Response has been posted to the PCIe subsystem. As the response is posted then there is no way of knowing when it is received by the EndPoint. Fields:

prgindex unsigned int

The PRG Index this request corresponds to.

response enum

If has a SubstreamID (PASID) then if it is 'global' or not.

ssd enum

SSD of response.

streamid unsigned int

The StreamID to use.

substreamid unsigned int

The SubstreamID to use (or ~0u if not used). This corresponds to the PASID.

smmu_thread_wait_wake

Traces a thread's wait/wake status. Fields:

current_ticks unsigned int

The current tick count of simulated time.

event enum

What is happening to this thread.

thread_index unsigned int

The ID of this thread.

ticks unsigned int

If the event relates to a time then this is held in this field. Otherwise, 0.

stall_transaction

A transaction is about to stall. Fields:

stag unsigned int

STAG.

streamid unsigned int

StreamID.

streamid_ns bool

The StreamID was for the non-secure world.

trans_id unsigned int

Transaction id.

stall_transaction_inhibited_by_STALL_MAX

A transaction is about to stall but the maximum number of transactions have stalled and we can't report this one to the event queue (even if non-full). Fields:

streamid unsigned int

StreamID.

streamid_ns bool

The StreamID was for the non-secure world.

trans_id unsigned int

Transaction id.

stall_transaction_resuming

A stalled transaction is resuming. Fields:

stag unsigned int

STAG if appropriate, or if was inhibited by STALL_MAX then 0xFAFA.

stallresult enum

What the transaction resumed to do.

streamid unsigned int

StreamID.

streamid_ns bool

The StreamID was for the non-secure world.

trans_id unsigned int

Transaction id.

start_ptw_read

Page Table Walk (read). This is the start of the physical access that the SMMU is making. Fields:

adomain enum

The actual attributes of the access that was used, after IMP DEF mangling.

aprot enum

The actual attributes of the access that was used, after IMP DEF mangling. PAS[1] (bit[3]), Instruction/Data (bit[2]), PAS[0] (bit[1]), Privileged/User (bit[0]).

desckind enum

The kind of descriptor.

inner_cache enum

The actual attributes of the access that was used, after IMP DEF mangling.

input_address unsigned int

Input address this lookup is for (after TBI normalisation).

mecid unsigned int

The MECID of the TTD fetch.

mpam_pmg_and_partid unsigned int

Bits[23:16] are the MPAM PMG. Bits[15:0] are the physical PARTID.

ns enum

Non-Secure on the bus.

outer_cache enum

The actual attributes of the access that was used, after IMP DEF mangling.

pa_address unsigned int

Physical address.

pas enum

The PAS of the TTD fetch.

ssd enum

SSD of the transaction.

ssd_ns enum

Security state of stream.

streamid unsigned int

The StreamID this is for.

substreamid unsigned int

The SubstreamID this is for.

trans_id unsigned int

Transaction id.

ttb_grain_stage_and_level unsigned int

Which TTB used in bits[31:24]. Which Grain size used in bit[23:16]:- 0 – 4 KiB 1 – 16 KiB 2 – 64 KiB 3 – 64 KiB ARMv8.2 where descriptor bits [15:12] for bits [51:48] of output address 4 – 4 KiB FEAT_LPA2 where address is {bits[9:8], bits[49:12]} 5 – 16 KiB FEAT_LPA2 where address is {bits[9:8], bits[49:14], 2'b00} 6 – 64 KiB ARMv8.2 where descriptor bits [15:12] are ignored Stage in bits[15:8]. Level in bits[7:0], level -1 is 0xF.

ste_cc.STE_commentary

A commentary from the cache about what is going on. Fields:

output string

The stream output.

ste_cc.STE_entry_invalidated

An entry in the cache has been invalidated. Fields:

entry_id unsigned int

The entry id that is being invalidated.

reason enum

Why the entry is being invalidated.

ste_entry_allocated

An STE entry has been allocated. Fields:

entry_id unsigned int

The id of the cache entry. Top bit is set if is a debug entry.

ns enum

For the non-secure world.

ssd enum

The SSD of the entry.

ste string

A textual description of the STE.

streamid unsigned int

The StreamID of the entry this will match.

tlb_entry_allocated

A TLB entry has been allocated. Fields:

aset unsigned int

The ASET of this entry.

asid unsigned int

ASID if appropriate.

index unsigned int

Index of TLB entry.

input_end_incl_address unsigned int

The end inclusive address of the input range that this matches.

input_start_address unsigned int

The start address of the input range that this matches.

output_end_incl_address unsigned int

The end inclusive address of the output range.

output_start_address unsigned int

The start address of the output range.

scheme enum

The tagging scheme used.

ssd enum

The SSD of the tagging scheme.

ssd_ns enum

Security State.

tbi bool

Was the entry formed using Top Byte Ignore (TBI).

vmid unsigned int

VMID if appropriate.

tlb_info_tlb_entries_overlap

A TLB entry was inserted into the TLB and it overlaps an existing entry. This isn't a problem as it was inserted in such a way that it architecturally works. Fields:

end_address_of_new_entry unsigned int

End address of new entry.

end_address_of_old_entry unsigned int

End address of old entry.

how_inserted enum

How the entry was inserted.

index_of_new_entry unsigned int

Index of new entry.

index_of_old_entry unsigned int

Index of old entry.

start_address_of_new_entry unsigned int

Start address of new entry.

start_address_of_old_entry unsigned int

Start address of old entry.

verbose_commentary

This is a verbose commentary on the translation process the SMMU is performing. Fields:

output string

The stream output.

warning_MSI_EVENTQ_address_out_of_range_of_oas

The MSI address is configured outside of the range of the downstream address bus size. Fields:

address unsigned int

The untruncated address of the MSI.

warning_MSI_GERROR_address_out_of_range_of_oas

The MSI address is configured outside of the range of the downstream address bus size. Fields:

address unsigned int

The untruncated address of the MSI.

warning_MSI_HACDBS_PROCESSING_COMPLETE_address_out_of_range_of_oas

The MSI address is configured outside of the range of the downstream address bus size.

Fields:

address unsigned int

The untruncated address of the MSI.

warning_MSI_HDBSS_TABLE_FULL_address_out_of_range_of_oas

The MSI address is configured outside of the range of the downstream address bus size.

Fields:

address unsigned int

The untruncated address of the MSI.

warning_MSI_PRIQ_address_out_of_range_of_oas

The MSI address is configured outside of the range of the downstream address bus size.

Fields:

address unsigned int

The untruncated address of the MSI.

warning_MSI_S_EVENTQ_address_out_of_range_of_oas

The MSI address is configured outside of the range of the downstream address bus size.

Fields:

address unsigned int

The untruncated address of the MSI.

warning_MSI_S_GERROR_address_out_of_range_of_oas

The MSI address is configured outside of the range of the downstream address bus size.

Fields:

address unsigned int

The untruncated address of the MSI.

warning_MSI_S_HACDBS_PROCESSING_COMPLETE_address_out_of_range_of_oas

The MSI address is configured outside of the range of the downstream address bus size.

Fields:

address unsigned int

The untruncated address of the MSI.

warning_MSI_S_HDBSS_TABLE_FULL_address_out_of_range_of_oas

The MSI address is configured outside of the range of the downstream address bus size.

Fields:

address unsigned int

The untruncated address of the MSI.

warning_MSI_pmcg_address_out_of_range_of_oas

The MSI Address of the Performance Monitor Counter Group (PMCG) is out of range of the OAS and so will be silently truncated. Fields:

address unsigned int

The untruncated address of the MSI.

pmcg_index unsigned int

Index of the PMCG.

ssd_ns bool

True if this PMCG controlled by the non-secure world.

warning_discarding_interrupt_EVENTQ_as_irqen_low

Interrupt generation is turned off by SMMU_IRQ_CTRL.EVENTQ_IRQEN.

warning_discarding_interrupt_GERROR_as_irqen_low

Interrupt generation is turned off by SMMU_IRQ_CTRL.GERROR_IRQEN.

warning_discarding_interrupt_HACDBS_PROCESSING_COMPLETE_as_irqen_low

Interrupt generation is turned off by SMMU_IRQ_CTRL.HACDBS_IRQEN.

warning_discarding_interrupt_HDBSS_TABLE_FULL_as_irqen_low

Interrupt generation is turned off by SMMU_IRQ_CTRL.HDBSS_IRQEN.

warning_discarding_interrupt_PRIQ_as_irqen_low

Interrupt generation is turned off by SMMU_IRQ_CTRL.PRIQ_IRQEN.

warning_discarding_interrupt_S_EVENTQ_as_irqen_low

Interrupt generation is turned off by SMMU_S_IRQ_CTRL.EVENTQ_IRQEN.

warning_discarding_interrupt_S_GERROR_as_irqen_low

Interrupt generation is turned off by SMMU_S_IRQ_CTRL.GERROR_IRQEN.

warning_discarding_interrupt_S_HACDBS_PROCESSING_COMPLETE_as_irqen_low

Interrupt generation is turned off by SMMU_IRQ_S_CTRL.HACDBS_IRQEN.

warning_discarding_interrupt_S_HDBSS_TABLE_FULL_as_irqen_low

Interrupt generation is turned off by SMMU_IRQ_S_CTRL.HDBSS_IRQEN.

warning_ns_gerrorn_bad_acknowledge

A GERROR was acknowledged by SW in the GERRORN register that did not have an active error. The result is **UNPREDICTABLE** if this will generate an interrupt or not. Fields:

which enum

Which GERROR was acknowledge when there was no active error.

warning_reg_after_doesnt_match_written_value

A write occurred that tried to set bits in a register, that for one reason or another, failed to get written. Fields:

desc string

The textual description of what happened.

warning_s_gerrorn_bad_acknowledge

A GERROR was acknowledged by SW in the GERRORN register that did not have an active error. The result is **UNPREDICTABLE** if this will generate an interrupt or not. Fields:

which enum

Which GERROR was acknowledge when there was no active error.

what_going_to_do_with_terminated_event

A terminating transaction has produced an event, this tells you what the model is going to do with the event. Fields:

CD.S bool

The CD.S field if available.

S2 bool

The event is related to Stage 2.

STE.S1STALLD bool

The STE.S1STALLD field if available.

STE.S2S bool

The STE.S2S field if available.

aborts bool

The transaction will abort.

axmmuflow enum

The AxMMUFLOW for this transaction group. storable – allow the transaction to be stalled if configured. translated-access – is a PCIe Translated Access non-storable – do not stall transaction even if configured. transfault-flow – do not report for Translation Related Faults, but tell client device to use a page request mechanism instead (for example PRI).

is_tr_fault bool

Is a Translation Related fault.

protected_mode bool

The transaction is protected-mode. As such, it cannot stall and will obey the report configuration bits.

reports bool

The transaction will attempt to report.

ssd enum

The SSD of the transaction.

ssd_ns bool

The transaction is classified as SSD non-secure.

supports_stall_model bool

The implementation supports the stall model.

trans_id unsigned int

The transaction id.

why_abort_decision enum

The reason why the transaction aborted/did not abort.

why_report_decision enum

The reason why the transaction reported/did not report.

2.167 SMMUv3TestEngine

This section describes the trace sources.

access_denied_due_to_security

The access was denied as the frame and the security PCTRL.SSD_NS did not allow access.

Fields:

address unsigned int

Address.

frame_in_set unsigned int

Which frame in the 64 KiB set it belongs to.

is_read bool

Is the transaction a read.

ns bool

Is the PAS of the transaction non-secure.

pas enum

The pas of the transaction.

priv bool

Is the set the privileged set of frames?.

set_pair unsigned int

Which 128 KiB set-pair it belongs to.

ssd enum

The SSD of the frame.

ssd_ns bool

The SSD NS of the frame.

frame_misconfigured_trace

The frame was written but is misconfigured. Fields:

cmd unsigned int

The command that is running.

frame_in_set unsigned int

Which frame in the 64 KiB set it belongs to.

priv bool

Is the set the privileged set of frames?.

set_pair unsigned int

Which 128 KiB set-pair it belongs to.

why enum

Why it is misconfigured?.

why_data unsigned int

Data related to why it is misconfigured:- src-attributes-illegal : the halfword attributes bits[15-0], NoStreamID bit[16] ,SSD bits[20-17] and PAS bits[24-21] dest-attributes-illegal : the halfword attributes bits[15-0], NoStreamID bit[16] ,SSD bits[20-17] and PAS bits[24-21] msi-attributes-illegal : the halfword attributes bits[15-0], NoStreamID bit[16] ,SSD bits[20-17] and PAS bits[24-21] downstream-port-index-out-of-range : port index substreamid-out-of-range : SubstreamID SUM64, begin is not aligned to uint64_t : begin SUM64, end_incl is not aligned to uint64_t : end_incl PCIe MSIAddress field should be 0/1 : msiaddress PCIe MSIData field should be < 2048 : msidata PCIe SSD not either Realm or Non-Secure : ssd range-* : the end_incl.

msi_aborted

An MSI aborted. Fields:

frame unsigned int

Frame generating MSI.

msiaddress_or_pcie_msi_enable unsigned int

MSI address, or in PCIe enable MSI.

msiattr unsigned int

MSI attributes. Ignored for PCIe.

msidata_or_pcie_msi_index unsigned int

MSI data, or in PCIe the MSI index to use.

pcie_result enum

If this MSI was generated on PCIe then the result.

streamid unsigned int

StreamID.

substreamid unsigned int

SubstreamID.

msi_generated

An MSI is generated for the specified work load. Fields:

frame unsigned int

Frame generating MSI.

msiaddress unsigned int

MSI address.

msiattr unsigned int

MSI attributes.

msidata unsigned int

MSI data.

streamid unsigned int

StreamID.

substreamid unsigned int

SubstreamID.

read_access

A successful read access occurred. Fields:

address unsigned int

Address.

data unsigned int

The data that was read.

frame_in_set unsigned int

Which frame in the 64 KiB set it belongs to.

ns bool

Is the PAS of the transaction non-secure.

offset unsigned int

Offset of the first byte read.

pas enum

The pas of the transaction.

priv bool

Is the set the privileged set of frames?.

set_pair unsigned int

Which 128 KiB set-pair it belongs to.

ssd enum

The SSD of the frame.

work_completed

The specified work was completed. Fields:

frame unsigned int

Frame that completed.

work_memcpy

A frame has been setup to do memcpy. Fields:

attributes unsigned int

Attributes.

begin unsigned int

begin address.

dest unsigned int

Destination address.

end_incl unsigned int

End inclusive address.

frame unsigned int

Frame setup.

msiaddress unsigned int

The msiaddress field.

msiattr unsigned int

The msiattr field.

msidata unsigned int

The msidata field.

seed unsigned int

Seed.

streamid unsigned int

The streamid to use if controllable.

stride unsigned int

Stride.

substreamid unsigned int

The substreamid to use if controllable.

work_rand48

A frame has been setup to do rand48. Fields:

attributes unsigned int

Attributes.

begin unsigned int

begin address.

end_incl unsigned int

End inclusive address.

frame unsigned int

Frame setup.

msiaddress unsigned int

The msiaddress field.

msiattr unsigned int

The msiattr field.

msidata unsigned int

The msidata field.

seed unsigned int

Seed.

streamid unsigned int

The streamid to use if controllable.

stride unsigned int

Stride.

substreamid unsigned int

The substreamid to use if controllable.

work_sum64

A frame has been setup to do sum64. Fields:

attributes unsigned int

Attributes.

begin unsigned int

begin address.

end_incl unsigned int

End inclusive address.

frame unsigned int

Frame setup.

msiaddress unsigned int

The msiaddress field.

msiattr unsigned int

The msiattr field.

msidata unsigned int

The msidata field.

seed unsigned int

Seed.

start_of_sum unsigned int

The value of udata[1] that is the initial value that we start adding to.

streamid unsigned int

The streamid to use if controllable.

stride unsigned int

Stride.

substreamid unsigned int

The substreamid to use if controllable.

work_unit_memcpy_finished

Part (or all) of the memcpy engine has finished a work unit. Fields:

attr unsigned int

Source/Destination attributes from the frame.

dest_addr unsigned int

Destination address.

frame unsigned int

Frame doing work.

number_of_bytes unsigned int

Number of bytes for this work unit.

ok bool

Transfer was OK.

src_addr unsigned int

Source address.

work_unit_memcpy_started

Part (or all) of the memcpy engine has started a work unit. Fields:

attr unsigned int

Source/Destination attributes from the frame.

dest_addr unsigned int

Destination address.

frame unsigned int

Frame doing work.

number_of_bytes unsigned int

Number of bytes for this work unit.

src_addr unsigned int

Source address.

work_unit_rand48_finished

Part (or all) of the rand48 engine has finished a work unit. Fields:

attr unsigned int

Source/Destination attributes from the frame.

dest_addr unsigned int

Source address.

frame unsigned int

Frame doing work.

number_of_bytes unsigned int

Number of bytes for this work unit.

ok bool

Transfer was OK.

seed_for_work_unit unsigned int

The seed used for the work unit.

work_unit_rand48_started

Part (or all) of the rand48 engine has started a work unit. Fields:

attr unsigned int

Source/Destination attributes from the frame.

dest_addr unsigned int

Destination address.

frame unsigned int

Frame doing work.

number_of_bytes unsigned int

Number of bytes for this work unit.

seed_for_work_unit unsigned int

The seed used for the work unit.

work_unit_sum64_finished

Part (or all) of the sum64 engine has finished a work unit. Fields:

attr unsigned int

Source/Destination attributes from the frame.

frame unsigned int

Frame doing work.

number_of_bytes unsigned int

Number of bytes for this work unit.

ok bool

Transfer was OK.

running_sum unsigned int

The current running sum.

src_addr unsigned int

Source address.

work_unit_sum64_started

Part (or all) of the sum64 engine has started a work unit. Fields:

attr unsigned int

Source/Destination attributes from the frame.

frame unsigned int

Frame doing work.

number_of_bytes unsigned int

Number of bytes for this work unit.

src_addr unsigned int

Source address.

write_access

A successful write access occurred. Fields:

address unsigned int

Address.

data unsigned int

The data that was read.

frame_in_set unsigned int

Which frame in the 64 KiB set it belongs to.

ns bool

Is the PAS of the transaction non-secure.

offset unsigned int

Offset of the first byte read.

pas enum

The pas of the transaction.

priv bool

Is the set the privileged set of frames?.

set_pair unsigned int

Which 128 KiB set-pair it belongs to.

ssd enum

The SSD of the frame.

write_denied_as_running

The write was denied as the frame is running. Fields:

address unsigned int

Address.

cmd unsigned int

The command that is running.

frame_in_set unsigned int

Which frame in the 64 KiB set it belongs to.

priv bool

Is the set the privileged set of frames?.

set_pair unsigned int

Which 128 KiB set-pair it belongs to.

write_to_PCTRL_SSD_NS_ignored_trace

The write to PCTRL.SSD was ignored because the transaction is trying to change the frame to security state. The rest of the write to PCTRL has taken effect. Fields:

address unsigned int

Address.

frame_in_set unsigned int

Which frame in the 64 KiB set it belongs to.

pas enum

The pas of the transaction.

set_pair unsigned int

Which 128 KiB set-pair it belongs to.

ssd enum

The SSD of the frame.

2.168 SSU

This section describes the trace sources.

Safety Mechanisms registers

Safety Mechanisms registers Trace: . Fields:

ERR_CTRL unsigned int

.

ERR_FR unsigned int

.

ERR_STATUS unsigned int

.

SMEN unsigned int

.

SMERR unsigned int

.

Source string

.

Status Control Register

Status Control Register Update: . Fields:

CNTRL_REG_VAL unsigned int

.

register_value_change

Register %{REG_NAME}: Type %{REG_TYPE}: Previous value %{PREV_VAL}: Current value %{CURR_VAL}. Fields:

CURR_VAL unsigned int

Current register value.

OFFSET unsigned int

Register offset.

PREV_VAL unsigned int

Previous register value.

REG_NAME string

Register name.

REG_TYPE string

Register type.

WRITE_VAL unsigned int

Value to be written into the register.

ssu_state_change

SSU STATE TRANSITION: %{PREV_STATE} -> %{CURR_STATE}. Fields:

CURR_STATE string

CURRENT SSU STATE.

PREV_STATE string

PREVIOUS SSU STATE.

2.169 SYSTEM_FMU

This section describes the trace sources.

interrupt_status

The status of %{ERR_TYPE} interrupt is %{STATUS}. Fields:

INTERRUPT_TYPE string

Type of interrupt whose status is shown.

STATUS string

Set/Clear status of the interrupt.

key_registers_values

Trace the key FMU register's values. Fields:

ERRGSR unsigned int

.

ERRGSR2 unsigned int

.

SMEN unsigned int

.

SMERR unsigned int

.

Source string

.

lock_status

The FMU Access KEY register changed from %{PREV_ACCESS_KEY_STATUS} to %{CURR_ACCESS_KEY_STATUS}. Fields:

CURR_ACCESS_KEY_STATUS string

Current lock status of the FMU KEY register.

PREV_ACCESS_KEY_STATUS string

Previous lock status of the FMU KEY register.

register_value_change

Register %{REG_NAME}: Type %{REG_TYPE}: Previous value %{PREV_VAL}: Current value %{CURR_VAL}. Fields:

CURR_VAL unsigned int

Current register value.

OFFSET unsigned int

Register offset.

PREV_VAL unsigned int

Previous register value.

REG_NAME string

Register name.

REG_TYPE string

Register type.

WRITE_VAL unsigned int

Value to be written into the register.

upstream_fmu_signals

Received %{ERROR_TYPE} interrupt from Upstream FMU index %{UPSTREAM_FMU_INDEX} with status %{STATUS}. Fields:

ERROR_TYPE string

The type of error received by SystemFMU.

STATUS string

Set/Clear status of the incoming interrupt.

UPSTREAM_FMU_INDEX unsigned int

The index of the upstream FMU sending the signal.

2.170 SharedR

This section describes the trace sources.

dsu.register_read

DISPLAY Core %{core} makes %{is_debug:normal|debug} read access to DSU register %{reg_name}, and value %{value} is returned. Fields:

core unsigned int

core from which this read access comes from.

is_debug bool

true if this transaction is debug one, false otherwise.

reg_name string

Name of the register to which this read access is made.

value unsigned int

register value which is returned.

dsu.register_write

DISPLAY Core %{core} makes %{is_debug:normal|debug} write access to DSU register %{reg_name}, with value %{value} and write mask %{mask}. Fields:

core unsigned int

core from which this write access comes from.

is_debug bool

true if this transaction is debug one, false otherwise.

mask unsigned int

mask applied to the value.

reg_name string

Name of the register to which this write access is made.

value unsigned int

register value to be written.

dsu.register_write_ignored

DISPLAY Write access to %{reg_name} is made, which has been ignored. Fields:

reg_name string

Name of the register to which this write access is made.

dsu.utility_bus.access

DISPLAY %{is_debug:Normal|Debug} %{is_read:write|read} access has been made to UtilityBus to access %{component_accessed} %{access_is_to_cluster_reg:to %{target_core} core}. Fields:

access_is_to_cluster_reg bool

true if this access is against cluster registers, false otherwise.

component_accessed string

component to which this transaction is made.

is_debug bool

true if the access is a debug one, false otherwise.

is_read bool

true if the access is a read one, false otherwise.

offset unsigned int

Offset to the utility bus base address used to identify which register is being accessed.

target_core unsigned int

is the target core if the transaction is to core register.

dsu.utility_bus.access_is_razwi

DISPLAY %{is_read:Write|Read} access has been made to UtilityBus with offset %{offset}, but this access results in **RAZWI** because %{msg}. Fields:

is_read bool

true if the access is a read one, false otherwise.

msg string

the reason why this access is **RAZWI**.

offset unsigned int

Offset to the base address of utility bus used to identify which register is being accessed.

2.171 SignalDriver

This section describes the trace sources.

SIGNAL

Input signal to IRQgen driven. Fields:

Source enum

What triggered the signal change.

Value enum

Value of signal being driven.

2.172 SignalLogger

This section describes the trace sources.

SIGNAL

Input signal to SignalLogger driven. Fields:

Value enum

Value of signal being driven.

2.173 TLB

This section describes the trace sources.

ArchMsg.Error.normal_nsh_wb_cacheable_access

TLB Fill: Normal memory with non-shareable and inner/outer-WriteBack attributes. Fields:

EL unsigned int

Exception level of translation regime.

NON_SECURE unsigned int

Security state of translation regime.

PBase unsigned int

Physical address (stage1+2) of the start of the region covered by the TLB entry.

VBase unsigned int

Virtual page number of the start of the region covered by the TLB entry.

MMU_TLB_CONFLICT

TLB conflict. Fields:

AMEC bool

Alternative MECID.

ASID unsigned int

The Address Space Identifier of this TLB entry.

CORE_NUM unsigned int

Core number in a multi processor.

Hyp bool

Entry matches in Hyp state only.

MANAGER_ID_64 unsigned int

ManagerID64 number in a multi processor. Lowest 8 bits matches CORE_NUM field.

NS enum

Secure state in which TLB Entry will match.

PAGESIZE unsigned int

Size of the region (log2).

PBASE unsigned int

Physical base address of the region.

REGIME_EL enum

Entry matches in this translation regime.

VBASE unsigned int

Virtual base address of the region.

VMID unsigned int

Virtual Machine Identifier.

nG enum

Flag indicating whether ASID will be matched.

MMU_TLB_EVICT

TLB evict. Fields:

AMEC bool

Alternative MECID.

ASID unsigned int

The Address Space Identifier of this TLB entry.

CORE_NUM unsigned int

Core number in a multi processor.

Hyp bool

Entry matches in Hyp state only.

IS_CLEAN_STAGE2_DIRTY bool

Is entry being evicted or just its S2 Dirty state being cleaned.

MANAGER_ID_64 unsigned int

ManagerID64 number in a multi processor. Lowest 8 bits matches CORE_NUM field.

NS enum

Secure state in which TLB Entry will match.

PAGESIZE unsigned int

Size of the region (log2).

PAS enum

The Physical Address Space of the page.

PBASE unsigned int

Physical base address of the region.

REGIME_EL enum

Entry matches in this translation regime.

SECURITY_STATE enum

The Current Security State.

VBASE unsigned int

Virtual base address of the region.

VMID unsigned int

Virtual Machine Identifier.

nG enum

Flag indicating whether ASID will be matched.

MMU_TLB_FILL

TLB fill. Fields:

AMEC bool

Alternative MECID.

ASID unsigned int

The Address Space Identifier of this TLB entry.

ASSURED bool

Whether translation is assured.

CORE_NUM unsigned int

Core number in a multi processor.

EPAN bool

Enhanced PAN cached value for this page.

Hyp bool

Entry matches in Hyp state only.

INNERCACHE_RA bool

Is the inner cache allocate on read.

INNERCACHE_TYPE enum

Inner Caching scheme (NC/MB/WA).

INNERCACHE_WA bool

Is the inner cache allocate on write.

MANAGER_ID_64 unsigned int

ManagerID64 number in a multi processor. Lowest 8 bits matches CORE_NUM field.

MEMTYPE enum

Memory type.

NS enum

Secure state in which TLB Entry will match.

NSDesc enum

Secure state of transactions made using the TLB Entry.

OUTERCACHE_RA bool

Is the outer cache allocate on read.

OUTERCACHE_TYPE enum

Outer Caching scheme (NC/MB/WA).

OUTERCACHE_WA bool

Is the outer cache allocate on write.

PAGESIZE unsigned int

Size of the region (log2).

PAS enum

The Physical Address Space of the page.

PBASE unsigned int

Physical base address of the region.

PXN bool

Privileged Execute Never.

REGIME_EL enum

Entry matches in this translation regime.

SECURITY_STATE enum

The Current Security State.

SH enum

Shareability.

SIDE enum

Inst / Data.

TCMA bool

Is the relevant TCMA/TCMA0/TCMA1 bit enabled.

VBASE unsigned int

Virtual base address of the region.

VMID unsigned int

Virtual Machine Identifier.

X16HINT bool

16 Entry Contiguous Hint.

XN bool

Execute Never.

XS bool

Limited TLBI XS value for this page.

nG enum

Flag indicating whether ASID will be matched.

MMU_TLB_FLUSH

TLB flush.

MMU_TLB_FLUSH_ADDR

TLB flush, match address. Fields:

ADDR unsigned int

Address.

NS bool

Is Non-Secure.

NSHYP bool

Is Non-Secure HYP.

MMU_TLB_FLUSH_ADDR_RANGE

TLB flush, match address range. Fields:

ADDR_FIRST unsigned int

First address in range.

ADDR_LAST unsigned int

Last address in range.

NS bool

Is Non-Secure.

NSHYP bool

Is Non-Secure HYP.

MMU_TLB_FLUSH_ALL

TLB flush all entries.

MMU_TLB_FLUSH_ASID

TLB flush, match ASID. Fields:

ASID unsigned int

ASID of TLB flush.

MMU_TLB_FLUSH_VMID

TLB flush, match VMID. Fields:

VMID unsigned int

VMID of TLB flush.

MMU_TLB_HIT

TLB accesses hit. Fields:

ASID unsigned int

The Address Space Identifier of this TLB entry.

CORE_NUM unsigned int

Core number in a multi processor.

Hyp bool

Entry matches in Hyp state only.

MANAGER_ID_64 unsigned int

ManagerID64 number in a multi processor. Lowest 8 bits matches CORE_NUM field.

NS enum

Secure state in which TLB Entry will match.

REGIME_EL enum

Entry matches in this translation regime.

SIDE enum

Inst / Data.

VADDR unsigned int

Virtual address of the access.

VMID unsigned int

Virtual Machine Identifier.

MMU_TLB_MISS

TLB access miss. Fields:

ASID unsigned int

The Address Space Identifier of this TLB entry.

CORE_NUM unsigned int

Core number in a multi processor.

Hyp bool

Entry matches in Hyp state only.

MANAGER_ID_64 unsigned int

ManagerID64 number in a multi processor. Lowest 8 bits matches CORE_NUM field.

NS enum

Secure state in which TLB Entry will match.

REGIME_EL enum

Entry matches in this translation regime.

SIDE enum

Inst / Data.

VADDR unsigned int

Virtual address of the access.

VMID unsigned int

Virtual Machine Identifier.

MMU_TLB_SPILL

TLB removals caused by fill to a occupied slot.

2.174 Value64Logger

This section describes the trace sources.

VALUE

Input signal to ValueLogger driven. Fields:

Value unsigned int

Value of signal being driven.

2.175 ValueLogger

This section describes the trace sources.

VALUE

Input signal to ValueLogger driven. Fields:

Value unsigned int

Value of signal being driven.

2.176 VirtioBlockDevice

This section describes the trace sources.

data_read

The data read by the device. Fields:

addr unsigned int

Address.

data unsigned int

The data read.

ns bool

Non-secure access.

data_read_abort

The data read by the device aborted. Fields:

addr unsigned int

Address.

ns bool

Non-secure access.

size_in_bytes unsigned int

The data size in bytes.

data_read_start

The data read by the device. Fields:

addr unsigned int

Address.

ns bool

Non-secure access.

size_in_bytes unsigned int

The size to read.

data_write

The data write by the device completed successfully. Fields:

addr unsigned int

Address.

ns bool

Non-secure access.

size_in_bytes unsigned int

The size of the write that completed successfully.

data_write_abort

The data write by the device aborted. Fields:

addr unsigned int

Address.

ns bool

Non-secure access.

size_in_bytes unsigned int

The size of the write that aborted.

data_write_start

The data attempted to be written by the device. Fields:

addr unsigned int

Address.

data unsigned int

The data attempted to be written.

ns bool

Non-secure access.

register_read

A read of the register file. Fields:

addr unsigned int

Address.

data unsigned int

Data read.

what enum

What region of MMIO space it accessed.

register_read_abort

A read of the register file caused an abort. Fields:

addr unsigned int

Address.

what enum

What region of MMIO space it accessed.

width_in_bytes unsigned int

Width of data read.

register_write

A write of the register file. Fields:

addr unsigned int

Address.

data unsigned int

Data written.

what enum

What region of MMIO space it accessed.

register_write_abort

A write of the register file caused an abort. Fields:

addr unsigned int

Address.

data unsigned int

Data attempted to be written.

what enum

What region of MMIO space it accessed.

2.177 VirtioPCIBlockDevice

This section describes the trace sources.

data_read

The data read by the device. Fields:

addr unsigned int

Address.

data unsigned int

The data read.

ns bool

Non-secure access.

data_read_abort

The data read by the device aborted. Fields:

addr unsigned int

Address.

ns bool

Non-secure access.

size_in_bytes unsigned int

The data size in bytes.

data_read_start

The data read by the device. Fields:

addr unsigned int

Address.

ns bool

Non-secure access.

size_in_bytes unsigned int

The size to read.

data_write

The data write by the device completed successfully. Fields:

addr unsigned int

Address.

ns bool

Non-secure access.

size_in_bytes unsigned int

The size of the write that completed successfully.

data_write_abort

The data write by the device aborted. Fields:

addr unsigned int

Address.

ns bool

Non-secure access.

size_in_bytes unsigned int

The size of the write that aborted.

data_write_start

The data attempted to be written by the device. Fields:

addr unsigned int

Address.

data unsigned int

The data attempted to be written.

ns bool

Non-secure access.

register_read

A read of the register file. Fields:

addr unsigned int

Address.

data unsigned int

Data read.

what enum

What region of MMIO space it accessed.

register_read_abort

A read of the register file caused an abort. Fields:

addr unsigned int

Address.

what enum

What region of MMIO space it accessed.

width_in_bytes unsigned int

Width of data read.

register_write

A write of the register file. Fields:

addr unsigned int

Address.

data unsigned int

Data written.

what enum

What region of MMIO space it accessed.

register_write_abort

A write of the register file caused an abort. Fields:

addr unsigned int

Address.

data unsigned int

Data attempted to be written.

what enum

What region of MMIO space it accessed.

2.178 VirtualEthernetCrossover

This section describes the trace sources.

FRAME

Trace Ethernet frames as they go through. Fields:

data string

ASCII encoded packet data. Each line starts with a hex packet offset (in bytes) followed by up to 16 space separated bytes encoded as 2 hex digits. This data can be read by the tool text2pcap to generate a file which can be read by wireshark.

source enum

Identifies the source port from which the frame originated.

2.179 atc

This section describes the trace sources.

ArchMsg.Error.error

These messages are about activity occurring on the ATC that is considered an error. DISPLAY %{output}. Fields:

output string

The stream output.

ArchMsg.Info.info

These are information messages about what is happening in the ATC. DISPLAY %{output}. Fields:

output string

The stream output.

ArchMsg.Warning.atc_pri_allocation_changed_whilest_not_stopped

The PRI has PRGs in flight that were using credit_used credits, however, the PRI interface was disabled and the allocation of credits changed. Fields:

credits_used unsigned int

The number of credits currently in use by in-flight PRGs.

new_allocation unsigned int

The new allocation of credits set in the PRI header.

ArchMsg.Warning.atc_pri_allocation_reduced_below_current_credits_used

The PRI has PRGs in flight that were using credit_used credits, however, the PRI interface was disabled and the allocation of credits changed to a value less than that of those that are currently in use. The SW should probably of first have serviced the outstanding PRGs and

then waited for the interface to report Stopped before reducing the allocation. The PCIe spec is unclear as to the required behaviour. The model will continue to count credits. It will not allow the PRI interface to be re-enabled until Stopped (this is an implementation defined choice) and so when successfully re-enabled will then obey the new allocation. Fields:

credits_used unsigned int

The number of credits currently in use by in-flight PRGs.

new_allocation unsigned int

The new allocation of credits set in the PRI header.

ArchMsg.Warning.atc_pri_members_of_prg_inconsistent

A client has formed a PRG and all the members of the PRG should have the same execute mode and privilege request in. However, this PRG member has different ones to the first. PCIe says that the effect is undefined. The trace source pri_members_of_prg traces all members of the group. Fields:

PRGIndex unsigned int

The PRGIndex of this request.

expected_execute bool

The expected execute permission requested.

expected_priv bool

The expected privilege permission requested.

member_index unsigned int

The index of this member within the PRG.

number_of_members unsigned int

The number of members in this PRG.

page_address unsigned int

Page address of the request.

priv bool

Privileged permissions requested. Valid only for those requests with a PASID (SubstreamID).

rwX enum

Requested Read/Write/Execute permissions. Execute permission are only available for those requests with a PASID (SubstreamID), and read permission will automatically be asked for as well.

ssd enum

The SSD of the PRG.

streamid unsigned int

The StreamID of the PRI request.

substreamid unsigned int

The SubstreamID sent with the PRG request.

trans_id unsigned int

The transaction id of the failing ATS transaction.

ArchMsg.Warning.atc_pri_reset_receivedWhilstOutputtingPRG

The ATC was in the process of sending the members of a PRG to the SMMU but before it had finished sending them, then a PRI Reset or signal reset occurred and the rest of the PRG will not be sent. The may leave a partial PRG in the PRI queue of the SMMU. Note the difference to having sent the entire PRG (including Last) and then receiving some sort of reset. Fields:

id unsigned int

The ID of this request.

number_of_PRG_members unsigned int

The number of members of the PRG, or 0 if run out of credits or PRGIndexes.

prgindex unsigned int

The PRGIndex expected, or ~0u if run out of credits or PRGIndexes.

signal_reset bool

True if it was a signal reset, false if it was a PRI Reset via the PRI Capability Header.

ssd enum

The SSD of the stream.

substreamid unsigned int

The expected PASID on the *response*.

which_member_was_being_sent unsigned int

The index of which member of the PRG was being sent whilst we received reset. All subsequent members of the PRG will not be sent.

ArchMsg.Warning.atc_pri_response_Response_Failure

A PRI Response has been received with a Response Failure. The PRI interface is required to disable itself, set RF in the header and will wake all waiting PRGs, forcing them to fault. Fields:

pasid unsigned int

The PASID prefix on the response, or ~0u if none.

prgindex unsigned int

The PRGIndex in the response.

response enum

The response code.

ssd enum

The SSD of the stream.

ArchMsg.Warning.atc_pri_response_ignored_as_RF_set

Any PRI Responses received whilst the RF (Response Failure) bit of the PRI header is required to be ignored by the PCIe specification. This is a response being ignored due to that reason. Fields:

pasid unsigned int

The PASID prefix on the response, or ~0u if none.

prgindex unsigned int

The PRGIndex in the response.

response enum

The response code.

ssd enum

The SSD of the stream.

ArchMsg.Warning.pri_enabled_with_zero_allocation

The PRI interface was enabled but the assigned credit allocation is zero. There is no useful way that the PRI interface can operate in this state and the model will fault any transaction that needs to use PRI. Fields:

capacity unsigned int

The PRI Credit Capacity for this device.

ArchMsg.Warning.warning

These messages are about unusual (but not necessarily incorrect) activity occurring on the ATC. DISPLAY %{output}. Fields:

output string

The stream output.

atc_commentary

A commentary from the cache about what is going on. Fields:

output string

The stream output.

atc_entry

This monitors the formation or change of an atc entry. Fields:

CXL_io bool

The response's value for CXL.io.

XT_and_checking enum

The input XT / PAS checking intent.

atc_entry_id unsigned int

An ID identifying this entry.

change_inconsistent_nug bool

The change was inconsistent in NUG, and was ignored. This is almost certainly an error.

change_inconsistent_permissions bool

The change was inconsistent in the permissions granted that were known before. The change was rejected for known changes. This may represent an error.

change_inconsistent_remap bool

The input and/or output address ranges were changed. This may represent an error.

id unsigned int

The ID of the transaction that caused this fill.

input_address_begin unsigned int

The input address range covered by this entry.

input_address_end_incl unsigned int

The input address range covered by this entry.

is_protected_mode bool

Is a protected mode transaction.

nugrwxrwx_change string

The N, U, G (or TE bit) and rwxrwx flags before and after any change.

output_XT_and_checking enum

The output XT / PAS checking intent on ATS Translated Transactions.

output_address_begin unsigned int

The output address range covered by this entry.

output_address_end_incl unsigned int

The output address range covered by this entry.

reply_is_cacheable bool

The reply was cacheable and so changed the known permissions.

ssd enum

The SSD of the stream.

substreamid unsigned int

The SubstreamID of the entry.

atc_entry_allocated

An ATC entry has been allocated. Fields:

XT_and_checking enum

The input XT / PAS checking intent.

atc_entry_id unsigned int

Index of ATC entry.

input_address_begin unsigned int

The input address range covered by this entry.

input_address_end_incl unsigned int

The input address range covered by this entry.

is_CXL_io bool

The entry is CXL.io.

is_protected_mode bool

Is a protected mode transaction.

nugrwxrwx string

N, U, G bits and which permissions are unknown ('?'), granted (r/w/x) and denied ('-').
For rl-Streams, the G bit is the TE bit and traced as 'T' or '-!.

output_XT_and_checking enum

The output XT / PAS checking intent on ATS Translated Transactions.

output_address_begin unsigned int

The input address range covered by this entry.

output_address_end_incl unsigned int

The input address range covered by this entry.

pas enum

The PAS that this entry obtained from the SMMU.

ssd enum

The security state (SSD) of the entry.

substreamid unsigned int

The SubstreamID of the entry.

atc_entry_invalidated

An entry in the cache has been invalidated. Fields:

entry_id unsigned int

The entry id that is being invalidated.

reason enum

Why the entry is being invalidated.

atc_entry_none_formed

An ATC entry was attempted to be created but the ATS request forbade it being created.
Fields:

XT_and_checking enum

The input XT / PAS checking intent.

fault enum

Fault reason.

id unsigned int

The ID of the transaction that caused this fill.

input_address unsigned int

The input address range covered by this entry.

is_protected_mode bool

Is a protected mode transaction.

ssd enum

The SSD of the stream.

substreamid unsigned int

The SubstreamID that would have been used.

atc_inv_wait

An ATC Invalidate has to wait before completing because of outstanding transactions that are using the entry. Fields:

atc_inv_id unsigned int

The ID of the ATC Invalidate transaction.

atc_inv_woke

An ATC Invalidate had to wait before completing because of outstanding transactions that are using the entry. This is it waking up. Fields:

atc_inv_id unsigned int

The ID of the ATC Invalidate transaction.

because_of_reset bool

The ATC Invalidate was killed because of reset.

atc_invalidate

An ATC Invalidate message was received. Fields:

Global_if_substreamid bool

If has a SubstreamID then the Global flag.

aligned_address unsigned int

The aligned address to invalidate.

atc_inv_id unsigned int

The ID of this ATC Invalidate message.

atc_inv_ssd enum

The SSD of the ATC Invalidate message.

size_mask unsigned int

The mask implied by the Size field.

substreamid unsigned int

The SubstreamID (or ~0u if no SubstreamID).

atc_invalidate_kills_in_flight_ats

If an ATC Invalidate comes in whilst an ATS request is in flight that is affected by that ATC invalidate then it is killed and will be remade. Fields:

Global_if_substreamid bool

If has a SubstreamID then the Global flag.

aligned_address unsigned int

The aligned address to invalidate.

atc_id unsigned int

The ID of the ATC that caused the ATS request.

atc_inv_id unsigned int

The ID of this ATC Invalidate message.

atc_inv_ssd enum

The SSD of the ATC Invalidate message.

size_mask unsigned int

The mask implied by the Size field.

substreamid unsigned int

The SubstreamID (or ~0u if no SubstreamID).

atc_pri_allocation_reduced_whilest_client_forming_prg

The client device was creating a PRG whilst the allocation for the PRI was changed. This left the PRI unable to satisfy the client device's request and we have to ask the client again what to do. Fields:

credits_used unsigned int

The number of credits currently in use by in-flight PRGs.

new_allocation unsigned int

The new allocation of credits set in the PRI header.

atc_pri_ats_transaction_failed

The ATS transaction failed to get the required permissions from the SMMU. PRI has been enabled so it now tries to form a PRI request. Fields:

allocation unsigned int

The number of credits allocated to this device.

capacity unsigned int

The advertised capacity of the device.

credits_used unsigned int

The number of credits that are in use by in-flight PRG requests.

pri_enabled bool

The PRI interface is enabled.

priv bool

Privileged mode requested.

rx enum

Requested Read/Write/Execute permissions. Execute permission are only available for those requests with a PASID (SubstreamID), and read permission will automatically be asked for as well.

ssd enum

The SSD of the stream.

streamid unsigned int

The StreamID of the request.

substreamid unsigned int

The SubstreamID (PASID) of the request or ~0u if none.

trans_id unsigned int

The transaction id of the ATS request.

untranslated_address unsigned int

The untranslated address of the ATS request.

atc_pri_client_device_chose_not_to_form_a_prg

The client device was given an opportunity to form a PRG and it chose not to. Instead the transaction will be faulted. Fields:

trans_id unsigned int

The id of the failing ATS request.

atc_pri_could_not_form_prg_due_to_lack_of_prgindexes_or_credits

The client device was told that it couldn't form a PRG because of a lack of free PRGIndexes or because there were no free credits. The client device has the opportunity to either fault the transaction or to wait for some to become free. Fields:

trans_id unsigned int

The id of the failing ATS request.

wait bool

The client device chose to wait.

atc_pri_is_stopped

Traces why the PRI interface reports itself as stopped or not. Fields:

Stopped bool

Is the PRI Stopped?.

client_device_forming_requests bool

The number of requests to the client device that are forming a PRG.

credits_outstanding bool

The number of outstanding credits in outstanding PRGs.

atc_pri_members_of_prg

A client has formed a PRG and this describes each member of that PRG. Fields:

PRGIndex unsigned int

The PRGIndex of this request.

member_index unsigned int

The index of this member within the PRG.

number_of_members unsigned int

The number of members in this PRG.

page_address unsigned int

Page address of the request.

priv bool

Privileged permissions requested. Valid only for those requests with a PASID (SubstreamID).

rwX enum

Requested Read/Write/Execute permissions. Execute permission are only available for those requests with a PASID (SubstreamID), and read permission will automatically be asked for as well.

ssd enum

The SSD of the stream.

streamid unsigned int

The StreamID of the PRI request.

substreamid unsigned int

The SubstreamID sent with the PRG request.

trans_id unsigned int

The transaction id of the failing ATS transaction.

atc_pri_outstanding_prg_request_killed_by_PRI_Reset_or_Response_Failure

This indicates that a PRI request (or one that was waiting for credits or a PRGIndexes to become free) was killed because either:- * software used PRI Reset in the PCIe PRI header, or * a PRI response was received with Response Failure, the effect of which is to kill all requests waiting for a PRG response. Fields:

trans_id unsigned int

The transaction id of the ATS request that formed the PRI request.

atc_pri_outstanding_prg_when_interface_disabled

The PRI interface was disabled but there are outstanding PRI requests that have to be completed before the interface is stopped. Fields:

credits unsigned int

The number of credits that this group uses.

expected_PASID_on_response unsigned int

The expected PASID that should be on the PRI response (or ~0u if none expected).

prgindex unsigned int

The PRGIndex of the outstanding request.

trans_id unsigned int

The transaction id of the ATS request that formed the PRI request.

atc_pri_prg_request

A PRI Page Request Group (PRG) has . Fields:

PASID_on_response unsigned int

The expected PASID on the response, or ~0u if none.

allocation unsigned int

The number of credits allowed to be used.

credits_used unsigned int

The number of credits in use now.

number_of_members unsigned int

The number of members in the group.

prgindex unsigned int

The PRGIndex used.

trans_id unsigned int

The transaction id of the failing ATS transaction.

atc_pri_response_unrecognised_causes_uprgi

A PRI Response has been received that doesn't match a waiting PRG. The UPRGI bit in the PRI header will be set and then this response will be ignored. Fields:

match_PRG_currently_committing bool

This is true if the PRI Response does match a PRG that is currently being sent to the SMMU, but because it is not yet complete then it should not have received a response yet and so the device treats this as UPRGI.

pasid unsigned int

The PASID prefix on the response, or ~0u if none.

prgindex unsigned int

The PRGIndex in the response.

response enum

The response code.

ssd enum

The SSD of the stream.

atc_pri_response_valid

A PRI Response has been received that is valid and we are going to wake up some threads of activity. Fields:

credits_to_return unsigned int

Credits that were used by the PRG and are now returned.

pasid unsigned int

The PASID prefix on the response, or ~0u if none.

prgindex unsigned int

The PRGIndex in the response.

response enum

The response code.

ssd enum

The SSD of the stream.

atc_pri_throwing_away_ppr

A client device asked for too many PRI Page Requests (PPR) in the PRG. The model is discarding PPRs to form the PRG. Fields:

page_address unsigned int

Page address of the request.

priv bool

Privileged permissions requested. Valid only for those requests with a PASID (SubstreamID).

rwX enum

Requested Read/Write/Execute permissions. Execute permission are only available for those requests with a PASID (SubstreamID), and read permission will automatically be asked for as well.

ssd enum

The SSD of the stream.

streamid unsigned int

The StreamID of the PRI request.

substreamid unsigned int

SubstreamID (PASID), or ~0u if none.

trans_id unsigned int

The transaction id of the failing ATS transaction.

atc_pri_waiting

Track a transaction that is waiting for a PRI request. Fields:

id unsigned int

The ID of this request.

number_of_PRG_members unsigned int

The number of members of the PRG, or 0 if run out of credits or PRGIndexes.

prgindex unsigned int

The PRGIndex expected, or ~0u if run out of credits or PRGIndexes.

sleeping bool

True if this is the trace of the transaction sleeping, false if it is waking.

ssd enum

The SSD of the stream.

substreamid unsigned int

The expected PASID on the *response*.

atc_pri_woken_response

A transaction that was waiting for a PRI Response has been woken. Fields:

action enum

The action now to take now woken.

id unsigned int

The ID of this request.

prgindex unsigned int

The PRGIndex expected, or ~0u if run out of credits or PRGIndexes.

ssd enum

The SSD of the stream.

substreamid unsigned int

The expected PASID on the *response*.

control_write

This indicates a write to one of the control field of the ATC. Fields:

comments enum

Comments on the change.

value_after_write unsigned int

The value that the field was after the write.

which enum

Which control field was written to.

written_value unsigned int

The value that was written.

error_atc_entries_overlap

An ATC entry was attempted to be inserted into the ATC and was determined that it overlaps an existing entry. This check is not perfect but will catch simple errors. Fields:

end_address_of_new_entry unsigned int

End address of new entry.

end_address_of_old_entry unsigned int

End address of old entry.

index_of_new_entry unsigned int

Index of new entry.

index_of_old_entry unsigned int

Index of old entry.

start_address_of_new_entry unsigned int

Start address of new entry.

start_address_of_old_entry unsigned int

Start address of old entry.

final_remap_request

The final ATS result of the remap request. If this fails then PRI might be applied. In addition, PRI might make this remap request retry multiple times and is tracked by the `retry_number`. Fields:

XT_and_checking enum

The input XT / PAS checking intent.

address unsigned int

Address of the request.

id unsigned int

ID of ATC transaction.

if_ATST_send_substreamid bool

If we make an ATS-TranslatedAccess then send any SubstreamID (PASID) with the transaction also.

inner enum

The inner memory attribute.

instruction bool

Is an instruction access.

is_debug bool

Is a debug.

is_protected_mode bool

Is a protected mode transaction.

is_read bool

Is a read request.

outer enum

The outer memory attribute.

output_XT_and_checking enum

The output XT / PAS checking intent on ATS Translated Transactions.

output_address unsigned int

Output address.

privileged bool

Is a privileged access.

produced_ATST_transaction bool

The ATC produced an ATS-TranslatedTransaction.

result enum

The result of the translation.

retry_number unsigned int

The number of times this transaction has retried.

ssd enum

The SSD of the transaction.

substreamid unsigned int

Substream ID of the request (or ~0u if no SubstreamID).

initial_remap_request

The remap request for a bunch of transactions round this address. PRI might make this remap request retry multiple times and is tracked by the `retry_number` field. Fields:

XT_and_checking enum

The input XT / PAS checking intent.

address unsigned int

Address of the request.

id unsigned int

Unique ID of this request.

if_ATST_send_substreamid bool

If we make an ATS-TranslatedAccess then send any SubstreamID (PASID) with the transaction also.

instruction bool

Is an instruction access.

is_debug bool

Is a debug.

is_protected_mode bool

Is a protected mode transaction.

is_read bool

Is a read request.

privileged bool

Is a privileged access.

retry_number unsigned int

The number of times this transaction has retried.

ssd enum

The SSD of the transaction.

substreamid unsigned int

Substream ID of the request (or ~0u if no SubstreamID).

same_as

This request is the same as the specified other ATC transaction. Fields:

atc_entry_id unsigned int

ID of the ATC Entry that it uses (or ~0ull if not relevant).

id unsigned int

ID of ATC transaction.

same_as_id unsigned int

ID of the ATC transaction it is the same as.

2.180 v7_VGIC

This section describes the trace sources.

debug_out

Debug trace from the VGIC model, this contains implementation detail that is unlikely to be generally useful. Fields:

message string

The message from the VGIC.

sequence_id unsigned int

Each successive message is labeled with an increasing number.

gic_log_errors_out

Errors from the VGIC model. Fields:

message string

The message from the VGIC.

sequence_id unsigned int

Each successive message is labeled with an increasing number.

gic_log_fatal_out

Fatal error from the VGIC model. Fields:

message string

The message from the VGIC.

gic_log_warnings_out

Warnings from the VGIC model. Fields:

message string

The message from the VGIC.

sequence_id unsigned int

Each successive message is labeled with an increasing number.

vgic_cfgsdisable

The VGIC CFGSDISABLE pin has changed its state, this (un)locks down some configuration registers. Fields:

state bool

The state of the signal.

vgic_distributor_register_access

Distributor interface accesses on the VGIC. Fields:

cpu_id unsigned int

The CPU that is making the request.

data unsigned int

The data in the access aligned to the LSB.

is_read enum

Is the access a read?.

offset enum

Register within interface, top bit indicates NS state.

offset_in_word unsigned int

The offset of the access in the word.

size_in_bytes unsigned int

The size of the access in bytes.

vgic_fiq_out

The FIQ output signal to the core. Fields:

cpu_id unsigned int

The CPU id.

state bool

The state of the signal.

vgic_hypervisor_register_access

Hypervisor interface register accesses. Fields:

cpu_id unsigned int

The CPU that is making the request.

data unsigned int

The data in the access aligned to the LSB.

is_read enum

Is the access a read?.

offset enum

Register within interface, top bit indicates NS state.

offset_in_word unsigned int

The offset of the access in the word.

size_in_bytes unsigned int

The size of the access in bytes.

vgic_irq_out

The IRQ output signal to the core. Fields:

cpu_id unsigned int

The CPU id.

state bool

The state of the signal.

vgic_legacy_fiq_in

The legacy FIQ input signal into the VGIC. Fields:

cpu_id unsigned int

The CPU id. There is one legacy IRQ/FIQ per core.

state bool

The state of the signal.

vgic_legacy_irq_in

The legacy IRQ input signal into the VGIC. Fields:

cpu_id unsigned int

The CPU id. There is one legacy IRQ/FIQ per core.

state bool

The state of the signal.

vgic_physical_register_access

Physical interface accesses on the VGIC. Fields:

cpu_id unsigned int

The CPU that is making the request.

data unsigned int

The data in the access aligned to the LSB.

is_read enum

Is the access a read?.

offset enum

Register within interface, top bit indicates NS state.

offset_in_word unsigned int

The offset of the access in the word.

size_in_bytes unsigned int

The size of the access in bytes.

vgic_reset

The reset pin on the VGIC has been changed. Fields:

state bool

The state of the SPI signal.

vgic_spi

Shared Peripheral Interrupt signal changed. Fields:

id unsigned int

The interrupt id this will generate. SPIs start at ID 32 and so you will only see ids of >= 32.

state bool

The state of the SPI signal.

vgic_vfiq_out

The VFIQ output signal to the core. Fields:

cpu_id unsigned int

The CPU id.

state bool

The state of the signal.

vgic_virq_out

The VIRQ output signal to the core. Fields:

cpu_id unsigned int

The CPU id.

state bool

The state of the signal.

vgic_virtual_machine_register_access

Virtual machine interface register accesses. Fields:

cpu_id unsigned int

The CPU that is making the request.

data unsigned int

The data in the access aligned to the LSB.

is_read enum

Is the access a read?.

offset enum

Register within interface, top bit indicates NS state.

offset_in_word unsigned int

The offset of the access in the word.

size_in_bytes unsigned int

The size of the access in bytes.

vgic_virtual_maintenance_interrupt

The VGIC is signaling that it needs software help from the hypervisor to maintain the VGIC interfaces. Fields:

cpu_id unsigned int

The CPU id.

state bool

The state of the signal.

vgic_wakeup_fiq

The wakeup FIQ signal. Fields:

cpu_id unsigned int

The CPU id.

state bool

The state of the signal.

vgic_wakeup_irq

The wakeup IRQ signal. Fields:

cpu_id unsigned int

The CPU id.

state bool

The state of the signal.

Proprietary Notice

This document is protected by copyright and other related rights and the use or implementation of the information contained in this document may be protected by one or more patents or pending patent applications. No part of this document may be reproduced in any form by any means without the express prior written permission of Arm Limited ("Arm"). No license, express or implied, by estoppel or otherwise to any intellectual property rights is granted by this document unless specifically stated.

Your access to the information in this document is conditional upon your acceptance that you will not use or permit others to use the information for the purposes of determining whether the subject matter of this document infringes any third party patents.

The content of this document is informational only. Any solutions presented herein are subject to changing conditions, information, scope, and data. This document was produced using reasonable efforts based on information available as of the date of issue of this document. The scope of information in this document may exceed that which Arm is required to provide, and such additional information is merely intended to further assist the recipient and does not represent Arm's view of the scope of its obligations. You acknowledge and agree that you possess the necessary expertise in system security and functional safety and that you shall be solely responsible for compliance with all legal, regulatory, safety and security related requirements concerning your products, notwithstanding any information or support that may be provided by Arm herein. In addition, you are responsible for any applications which are used in conjunction with any Arm technology described in this document, and to minimize risks, adequate design and operating safeguards should be provided for by you.

This document may include technical inaccuracies or typographical errors. THIS DOCUMENT IS PROVIDED "AS IS". ARM PROVIDES NO REPRESENTATIONS AND NO WARRANTIES, EXPRESS, IMPLIED OR STATUTORY, INCLUDING, WITHOUT LIMITATION, THE IMPLIED WARRANTIES OF MERCHANTABILITY, SATISFACTORY QUALITY, NON-INFRINGEMENT OR FITNESS FOR A PARTICULAR PURPOSE WITH RESPECT TO THE DOCUMENT. For the avoidance of doubt, Arm makes no representation with respect to, and has undertaken no analysis to identify or understand the scope and content of, any patents, copyrights, trade secrets, trademarks, or other rights.

TO THE EXTENT NOT PROHIBITED BY LAW, IN NO EVENT WILL ARM BE LIABLE FOR ANY DAMAGES, INCLUDING WITHOUT LIMITATION ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL, PUNITIVE, OR CONSEQUENTIAL DAMAGES, HOWEVER CAUSED AND REGARDLESS OF THE THEORY OF LIABILITY, ARISING OUT OF ANY USE OF THIS DOCUMENT, EVEN IF ARM HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Reference by Arm to any third party's products or services within this document is not an express or implied approval or endorsement of the use thereof.

This document consists solely of commercial items. You shall be responsible for ensuring that any permitted use, duplication, or disclosure of this document complies fully with any relevant

export laws and regulations to assure that this document or any portion thereof is not exported, directly or indirectly, in violation of such export laws. Use of the word “partner” in reference to Arm’s customers is not intended to create or refer to any partnership relationship with any other company. Arm may make changes to this document at any time and without notice.

This document may be translated into other languages for convenience, and you agree that if there is any conflict between the English version of this document and any translation, the terms of the English version of this document shall prevail.

The validity, construction and performance of this notice shall be governed by English Law.

The Arm corporate logo and words marked with ® or ™ are registered trademarks or trademarks of Arm Limited (or its affiliates) in the US and/or elsewhere. Please follow Arm’s trademark usage guidelines at <https://www.arm.com/company/policies/trademarks>. All rights reserved. Other brands and names mentioned in this document may be the trademarks of their respective owners.

Arm Limited. Company 02557590 registered in England.

110 Fulbourn Road, Cambridge, England CB1 9NJ.

PRE-1121-V1.0

Product and document information

Read the information in these sections to understand the release status of the product and documentation, and the conventions used in Arm documents.

Product status

All products and services provided by Arm require deliverables to be prepared and made available at different levels of completeness. The information in this document indicates the appropriate level of completeness for the associated deliverables.

Product completeness status

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

Revision history

These sections can help you understand how the document has changed over time.

Document release information

The Document history table gives the issue number and the released date for each released issue of this document.

Document history

Issue	Date	Confidentiality	Change
1130-00	19 November 2025	Non-Confidential	Update for v11.30
1129-00	16 May 2025	Non-Confidential	Update for v11.29
1128-00	19 February 2025	Non-Confidential	Update for v11.28
1127-00	16 September 2024	Non-Confidential	Update for v11.27
1126-00	19 June 2024	Non-Confidential	Update for v11.26
1125-00	13 March 2024	Non-Confidential	Update for v11.25.
1124-00	6 December 2023	Non-Confidential	Update for v11.24.
1123-00	13 September 2023	Non-Confidential	Update for v11.23.
1122-00	14 June 2023	Non-Confidential	Update for v11.22.
1121-00	22 March 2023	Non-Confidential	First release.

Change history

For information about the functional changes to Fast Models, see the [Fast Models Release Notes](#).

Conventions

The following subsections describe conventions used in Arm documents.

Glossary

The Arm Glossary is a list of terms used in Arm documentation, together with definitions for those terms. The Arm Glossary does not contain terms that are industry standard unless the Arm meaning differs from the generally accepted meaning.

See the Arm Glossary for more information: developer.arm.com/glossary.

Typographic conventions

Arm documentation uses typographical conventions to convey specific meaning.

Convention	Use
italic	Citations.
bold	Interface elements, such as menu names. Terms in descriptive lists, where appropriate.
monospace	Text that you can enter at the keyboard, such as commands, file and program names, and source code.
monospace <u>underline</u>	A permitted abbreviation for a command or option. You can enter the underlined text instead of the full command or option name.
<and>	Encloses replaceable terms for assembler syntax where they appear in code or code fragments. For example: <div>MRC p15, 0, <Rd>, <CRn>, <CRm>, <Opcode_2></div>
SMALL CAPITALS	Terms that have specific technical meanings as defined in the <i>Arm® Glossary</i> . For example, IMPLEMENTATION DEFINED , IMPLEMENTATION SPECIFIC , UNKNOWN , and UNPREDICTABLE .



We recommend the following. If you do not follow these recommendations your system might not work.



Your system requires the following. If you do not follow these requirements your system will not work.



You are at risk of causing permanent damage to your system or your equipment, or harming yourself.



This information is important and needs your attention.



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



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

Useful resources

This document contains information that is specific to this product. See the following resources for other useful information.

Arm documents are available on developer.arm.com/documentation.

Confidential documents are only available to licensees, when logged in. Each document link in the tables below provides direct access to the online version of the document.

Arm product resources	Document ID	Confidentiality
Fast Models Model Trace Interface Reference Manual	DUI 0819	Non-Confidential
Fast Models Reference Guide	100964	Non-Confidential