# TRBSR_EL1, Trace Buffer Status/syndrome Register

The TRBSR_EL1 characteristics are:

## Purpose

Provides syndrome information to software for a trace buffer management event.

## Configuration

External register TRBSR_EL1 bits [63:0] are architecturally mapped to AArch64 System register TRBSR_EL1[63:0].

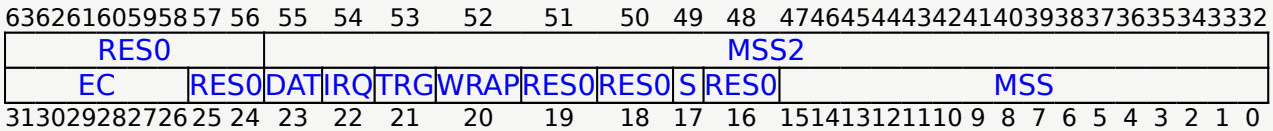
This register is present only when FEAT_TRBE_EXT is implemented. Otherwise, direct accesses to TRBSR_EL1 are res0.

TRBSR_EL1 is in the Core power domain.

## Attributes

TRBSR_EL1 is a 64-bit register.

## Field descriptions




### Bits [63:56]

Reserved, res0.

### MSS2, bits [55:32]

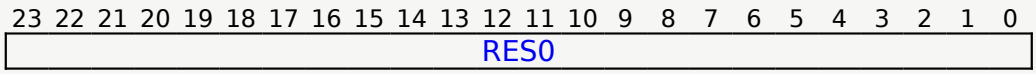
Management event Specific Syndrome 2. Contains syndrome specific to the management event.

The syndrome contents for each management event are described in the following sections.

The reset behavior of this field is:

- On a Cold reset, this field resets to an architecturally unknown value.

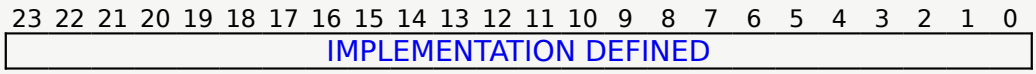
## MSS2 encoding for other trace buffer management events




**Bits [23:0]**

Reserved, res0.

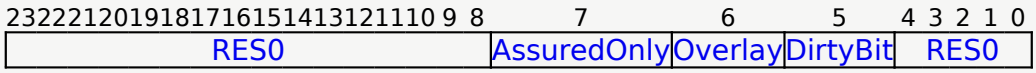
## MSS2 encoding for a buffer management event for an IMPLEMENTATION DEFINED reason




**IMPLEMENTATION DEFINED, bits [23:0]**

implementation defined.

## MSS2 encoding for stage 1 or stage 2 Data Aborts on write to trace buffer




**Bits [23:8]**

Reserved, res0.

**AssuredOnly, bit [7]**
**When FEAT_THE is implemented, TRBSR_EL1.EC == 0b100101 and GetTRBSR_EL1_FSC() == 0b0011xx:**

AssuredOnly flag. If a memory access generates a stage 2 Data Abort, then this field holds information about the fault.

| AssuredOnly | Meaning |
|---|---|
| 0b0 | Data Abort is not due to AssuredOnly. |
| 0b1 | Data Abort is due to AssuredOnly. |

**Otherwise:**

Reserved, res0.

**Overlay, bit [6]**
**When (FEAT_S1POE is implemented or FEAT_S2POE is implemented) and GetTRBSR_EL1_FSC() == 0b0011xx:**

Overlay flag. If a memory access generates a Data Abort for a Permission fault, then this field holds information about the fault.

| Overlay | Meaning |
|---|---|
| 0b0 | Data Abort due to Base Permissions. |
| 0b1 | Data Abort due to Overlay Permissions. |

**Otherwise:**

Reserved, res0.

**DirtyBit, bit [5]**
**When (FEAT_S1PIE is implemented or FEAT_S2PIE is implemented) and GetTRBSR_EL1_FSC() == 0b0011xx:**

DirtyBit flag. If a memory access generates a Data Abort (Write Access) for a Permission fault (When using Indirect Permission), then this field holds information about the fault.

| DirtyBit | Meaning |
|---|---|
| 0b0 | Permission Fault is not due to state of nDirty / Dirty bit. |
| 0b1 | Permission Fault is due to state of nDirty / Dirty bit. |

**Otherwise:**

Reserved, res0.

**Bits [4:0]**

Reserved, res0.

**EC, bits [31:26]**

Event Class. Top-level description of the cause of the trace buffer management event.

| EC | Meaning | MSS | Applies when |
|---|---|---|---|
| 0b000000 | Other trace buffer management event. All trace buffer management events other than those described by the other defined event class codes. | MSS encoding for other trace buffer management events | |

| | | | |
|---|---|---|---|
| 0b011110 | Granule Protection Check fault on write to trace buffer, other than Granule Protection Fault (GPF). That is, any of the following:<br>• Granule Protection Table (GPT) address size fault.<br>• GPT walk fault.<br>• Synchronous External abort on GPT fetch.<br>A GPF on translation table walk or update is reported as either a Stage 1 or Stage 2 Data Abort, as appropriate. Other GPFs are reported as a Stage 1 Data Abort. | MSS encoding for other trace buffer management events | When FEAT_RME is implemented |
| 0b011111 | Buffer management event for an implementation defined reason. | MSS encoding for a buffer management event for an IMPLEMENTATION DEFINED reason | |
| 0b100100 | Stage 1 Data Abort on write to trace buffer. | MSS encoding for stage 1 or stage 2 Data Aborts on write to trace buffer | |
| 0b100101 | Stage 2 Data Abort on write to trace buffer. | MSS encoding for stage 1 or stage 2 Data Aborts on write to trace buffer | |

All other values are reserved.

The reset behavior of this field is:

- On a Cold reset, this field resets to an architecturally unknown value.

**Bits [25:24]**

Reserved, res0.

**DAT, bit [23]**

Data. Indicates when the Trace Buffer Unit has trace data that has not yet been written to memory.

| DAT | Meaning |
|---|---|
| 0b0 | Internal buffers are empty. All Trace operations Accepted by the Trace Buffer Unit will Complete in finite time. |
| 0b1 | Internal buffers are not empty. |

When TRBSR_EL1.{DAT, S} is {0, 1}, meaning Collection is stopped and the Trace Buffer Unit internal buffers are empty, then all trace data has been written to memory. An additional Data Synchronization Barrier may be required to ensure that the writes are Complete. When TRBSR_EL1.DAT is 0 and Collection is not stopped, there may still be trace data held by the trace unit that the Trace Buffer Unit has not Accepted.

That is, TRBSR_EL1.DAT reads as 1 when the Trace Buffer Unit has Accepted trace data from the trace unit, but has not yet written it to memory.

The reset behavior of this field is:

- On a Cold reset, this field resets to an architecturally unknown value.

**IRQ, bit [22]**

Maintenance interrupt status.

| IRQ | Meaning |
|---|---|
| 0b0 | Maintenance interrupt is not asserted. |
| 0b1 | Maintenance interrupt is asserted. |

The reset behavior of this field is:

- On a Cold reset, this field resets to an architecturally unknown value.

**TRG, bit [21]**

Triggered.

| TRG | Meaning |
|---|---|
| 0b0 | No Detected Trigger has been observed since this field was last cleared to zero. |
| 0b1 | A Detected Trigger has been observed since this field was last cleared to zero. |

The reset behavior of this field is:

- On a Cold reset, this field resets to an architecturally unknown value.

**WRAP, bit [20]**

Wrapped.

| WRAP | Meaning |
|---|---|
| 0b0 | The current write pointer has not wrapped since this field was last cleared to zero. |
| 0b1 | The current write pointer has wrapped since this field was last cleared to zero. |

For each byte of trace the Trace Buffer Unit Accepts and writes to the trace buffer at the address in the current write pointer, if the current write pointer is equal to the Limit pointer minus one, the current write pointer is wrapped by setting it to the Base pointer, and this field is set to 1.

The reset behavior of this field is:

- On a Cold reset, this field resets to an architecturally unknown value.

**Bit [19]**

Reserved, res0.

**Bit [18]**
**From Armv9.3:**

Reserved, res0.

**When the PE sets this bit as the result of an External Abort:**

External Abort.

| EA | Meaning |
|---|---|
| 0b0 | An External Abort has not been asserted. |
| 0b1 | An External Abort has been asserted and detected by the Trace Buffer Unit. |

The reset behavior of this field is:

- On a Cold reset, this field resets to an architecturally unknown value.

**Otherwise:**

Reserved, res0.

**S, bit [17]**

Stopped.

| S | Meaning |
|---|---|
| 0b0 | Collection has not been stopped. |
| 0b1 | Collection is stopped. |

The reset behavior of this field is:

- On a Cold reset, this field resets to an architecturally unknown value.

**Bit [16]**

Reserved, res0.

**MSS, bits [15:0]**

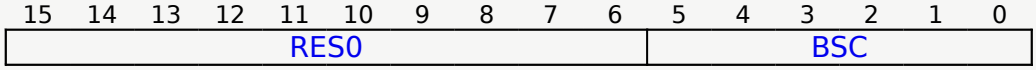
Management event Specific Syndrome. Contains syndrome specific to the management event.

The syndrome contents for each management event are described in the following sections.

The reset behavior of this field is:

- On a Cold reset, this field resets to an architecturally unknown value.

## MSS encoding for other trace buffer management events




**Bits [15:6]**

Reserved, res0.

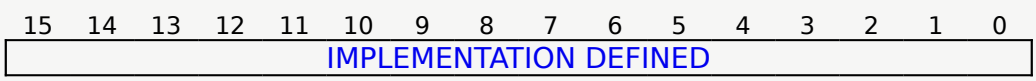
**BSC, bits [5:0]**

Trace buffer status code.

| BSC | Meaning |
|---|---|
| 0b000000 | Collection not stopped, or access not allowed. |
| 0b000001 | Trace buffer filled. Collection stopped because the current write pointer wrapped to the base pointer and the trace buffer mode is Fill mode. |
| 0b000010 | Trigger Event. Collection stopped because of a Trigger Event. See TRBTRG_EL1 for more information. |
| 0b000011 | Manual Stop. Collection stopped because of a Manual Stop event. See TRBCR.ManStop for more information. |

All other values are reserved.

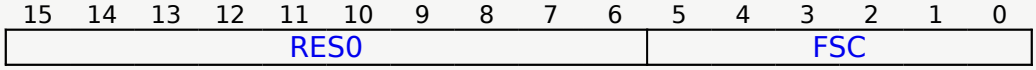
## MSS encoding for a buffer management event for an IMPLEMENTATION DEFINED reason




**IMPLEMENTATION DEFINED, bits [15:0]**

implementation defined.

## MSS encoding for stage 1 or stage 2 Data Aborts on write to trace buffer




### Bits [15:6]

Reserved, res0.

### FSC, bits [5:0]

Fault Status Code.

| FSC | Meaning | Applies when |
|---|---|---|
| 0b000000 | Address size fault, level 0 of translation or translation table base register. | |
| 0b000001 | Address size fault, level 1. | |
| 0b000010 | Address size fault, level 2. | |
| 0b000011 | Address size fault, level 3. | |
| 0b000100 | Translation fault, level 0. | |
| 0b000101 | Translation fault, level 1. | |
| 0b000110 | Translation fault, level 2. | |
| 0b000111 | Translation fault, level 3. | |
| 0b001001 | Access flag fault, level 1. | |
| 0b001010 | Access flag fault, level 2. | |
| 0b001011 | Access flag fault, level 3. | |
| 0b001000 | Access flag fault, level 0. | When FEAT_LPA2 is implemented |
| 0b001100 | Permission fault, level 0. | When FEAT_LPA2 is implemented |
| 0b001101 | Permission fault, level 1. | |

| | | |
|---|---|---|
| 0b001110 | Permission fault, level 2. | |
| 0b001111 | Permission fault, level 3. | |
| 0b010000 | Synchronous External abort, not on translation table walk or hardware update of translation table. | |
| 0b010001 | Asynchronous External abort. | |
| 0b010010 | Synchronous External abort on translation table walk or hardware update of translation table, level -2. | When FEAT_D128 is implemented |
| 0b010011 | Synchronous External abort on translation table walk or hardware update of translation table, level -1. | When FEAT_LPA2 is implemented |
| 0b010100 | Synchronous External abort on translation table walk or hardware update of translation table, level 0. | |

| | | |
|---|---|---|
| 0b010101 | Synchronous External abort on translation table walk or hardware update of translation table, level 1. | |
| 0b010110 | Synchronous External abort on translation table walk or hardware update of translation table, level 2. | |
| 0b010111 | Synchronous External abort on translation table walk or hardware update of translation table, level 3. | |
| 0b011011 | Synchronous parity or ECC error on memory access on translation table walk or hardware update of translation table, level -1. | When FEAT_LPA2 is implemented and FEAT_RAS is not implemented |
| 0b100001 | Alignment fault. | |
| 0b100010 | Granule Protection Fault on translation table walk or hardware update of translation table, level -2. | When FEAT_D128 is implemented and FEAT_RME is implemented |

| | | |
|---|---|---|
| 0b100011 | Granule Protection Fault on translation table walk or hardware update of translation table, level -1. | When FEAT_RME is implemented and FEAT_LPA2 is implemented |
| 0b100100 | Granule Protection Fault on translation table walk or hardware update of translation table, level 0. | When FEAT_RME is implemented |
| 0b100101 | Granule Protection Fault on translation table walk or hardware update of translation table, level 1. | When FEAT_RME is implemented |
| 0b100110 | Granule Protection Fault on translation table walk or hardware update of translation table, level 2. | When FEAT_RME is implemented |
| 0b100111 | Granule Protection Fault on translation table walk or hardware update of translation table, level 3. | When FEAT_RME is implemented |

| | | |
|---|---|---|
| 0b101000 | Granule Protection Fault, not on translation table walk or hardware update of translation table. | When FEAT_RME is implemented |
| 0b101001 | Address size fault, level -1. | When FEAT_LPA2 is implemented |
| 0b101010 | Translation fault, level -2. | When FEAT_D128 is implemented |
| 0b101011 | Translation fault, level -1. | When FEAT_LPA2 is implemented |
| 0b101100 | Address Size fault, level -2. | When FEAT_D128 is implemented |
| 0b110000 | TLB conflict abort. | |
| 0b110001 | Unsupported atomic hardware update fault. | When FEAT_HAFDBS is implemented |

All other values are reserved.

## Accessing TRBSR_EL1

The PE might ignore a write to TRBSR_EL1 if any of the following apply:

- TRBLIMITR_EL1.E == 1 and the Trace Buffer Unit is using Self-hosted mode.
- TRBLIMITR_EL1.XE == 1 and the Trace Buffer Unit is using External mode.

**TRBSR_EL1 can be accessed through the external debug interface:**

| Component | Offset | Instance |
|---|---|---|
| TRBE | 0x018 | TRBSR_EL1 |

This interface is accessible as follows:

- When DoubleLockStatus(), or !IsCorePowered(), or OSLockStatus() or !AllowExternalTraceBufferAccess(), accesses to this register generate an error response.
- Otherwise, accesses to this register are **RW**.

28/03/2023 16:02; 72747e43966d6b97dcbd230a1b3f0421d1ea3d94