

TRBIDR_EL1, Trace Buffer ID Register

The TRBIDR_EL1 characteristics are:

Purpose

Describes constraints on using the Trace Buffer Unit to software, including whether the Trace Buffer Unit can be programmed at the current Exception level.

Configuration

AArch64 System register TRBIDR_EL1 bits [63:0] are architecturally mapped to External register [TRBIDR_EL1\[63:0\]](#) when FEAT_TRBE_EXT is implemented.

This register is present only when FEAT_TRBE is implemented. Otherwise, direct accesses to TRBIDR_EL1 are undefined.

Attributes

TRBIDR_EL1 is a 64-bit register.

Field descriptions

63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
RES0																															
RES0												EA		RES0	F	P	Align														
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bits [63:12]

Reserved, res0.

EA, bits [11:8] From Armv9.3:

External Abort handling. Describes how the PE manages External aborts on writes made by the Trace Buffer Unit to the trace buffer.

EA	Meaning
0b0000	Not described.
0b0001	The PE ignores External aborts on writes made by the Trace Buffer Unit.

0b0010	The External abort generates an SError interrupt at the PE.
--------	---

All other values are reserved.

From Armv9.3, the value 0b0000 is not permitted.

[TRBIDR_EL1](#).EA describes only External aborts generated by the write to memory. External aborts on a translation table walk made by the Trace Buffer Unit generate trace buffer management events reported as MMU faults using [TRBSR_EL1](#).

This field has an implementation defined value.

Access to this field is **RO**.

Otherwise:

Reserved, res0.

Bits [7:6]

Reserved, res0.

F, bit [5]

Flag updates. Describes how address translations performed by the Trace Buffer Unit manage the Access flag and dirty state.

F	Meaning
0b0	Hardware management of the Access flag and dirty state for accesses made by the Trace Buffer Unit is always disabled for all translation stages.
0b1	Hardware management of the Access flag and dirty state for accesses made by the Trace Buffer Unit is controlled in the same way as explicit memory accesses in the trace buffer owning translation regime.

Note

If hardware management of the Access flag is disabled for a stage of translation, an access to a Page or Block with the Access flag bit not set in the descriptor will generate an Access Flag fault.

If hardware management of the dirty state is disabled for a stage of translation, an access to a Page or Block will ignore the Dirty Bit Modifier in the descriptor and might generate a Permission fault, depending on the values of the access permission bits in the descriptor.

From Armv9.3, the value 0 is not permitted.

This field has an implementation defined value.

Access to this field is **RO**.

P, bit [4]

Programming not allowed. When read at EL3, this field reads as zero. Otherwise, indicates that the trace buffer is owned by a higher Exception level or another Security state. Defined values are:

P	Meaning
0b0	Programming is allowed.
0b1	Programming not allowed.

The value read from this field depends on the current Exception level and the Effective values of [MDCR_EL3.NSTB](#), [MDCR_EL3.NSTBE](#), and [MDCR_EL2.E2TB](#):

- If EL3 is implemented, [MDCR_EL3.NSTB](#) is 0b0x, and either FEAT_RME is not implemented, or Secure state is implemented and [MDCR_EL3.NSTBE](#) is 0, then this field reads as one from:
 - Non-secure EL1 and Non-secure EL2.
 - If FEAT_RME is implemented, Realm EL1 and Realm EL2.
 - If Secure EL2 is implemented and enabled, and [MDCR_EL2.E2TB](#) is 0b00, Secure EL1.
- If EL3 is implemented, [MDCR_EL3.NSTB](#) is 0b1x and either FEAT_RME is not implemented or [MDCR_EL3.NSTBE](#) is 0, then this field reads as one from:
 - If Secure state is implemented, Secure EL1.
 - If Secure EL2 is implemented, Secure EL2.
 - If EL2 is implemented and [MDCR_EL2.E2TB](#) is 0b00, Non-secure EL1.
 - If FEAT_RME is implemented, Realm EL1 and Realm EL2.
- If FEAT_RME is implemented, and [MDCR_EL3.{NSTB, NSTBE}](#) is {0b1x, 1}, then this field reads as one from:
 - Non-secure EL1 and Non-secure EL2.
 - If Secure state is implemented, Secure EL1 and Secure EL2.
 - If [MDCR_EL2.E2TB](#) is 0b00, Realm EL1.

- If EL3 is not implemented, EL2 is implemented, and [MDCR_EL2](#).E2TB is 0b00, then this field reads as one from EL1.

Otherwise, this field reads as zero.

Align, bits [3:0]

Defines the minimum alignment constraint for writes to [TRBPTR_EL1](#) and [TRBTRG_EL1](#). Defined values are:

Align	Meaning
0b0000	Byte.
0b0001	Halfword.
0b0010	Word.
0b0011	Doubleword.
0b0100	16 bytes.
0b0101	32 bytes.
0b0110	64 bytes.
0b0111	128 bytes.
0b1000	256 bytes.
0b1001	512 bytes.
0b1010	1KB.
0b1011	2KB.

All other values are reserved.

This field has an implementation defined value.

Access to this field is **RO**.

Accessing TRBIDR_EL1

Accesses to this register use the following encodings in the System register encoding space:

MRS <Xt>, TRBIDR_EL1

op0	op1	CRn	CRm	op2
0b11	0b000	0b1001	0b1011	0b111

```

if PSTATE.EL == EL0 then
    UNDEFINED;
elsif PSTATE.EL == EL1 then
    if EL2Enabled() &&
        IsFeatureImplemented(FEAT_FGT) && (!HaveEL(EL3)) ||

```

```
SCR_EL3.FGTEn == '1') && HDFGRTR_EL2.TRBIDR_EL1 ==  
'1' then  
    AArch64.SystemAccessTrap(EL2, 0x18);  
else  
    X[t, 64] = TRBIDR_EL1;  
elsif PSTATE.EL == EL2 then  
    X[t, 64] = TRBIDR_EL1;  
elsif PSTATE.EL == EL3 then  
    X[t, 64] = TRBIDR_EL1;
```

[AArch32
Registers](#)

[AArch64
Registers](#)

[AArch32
Instructions](#)

[AArch64
Instructions](#)

[Index by
Encoding](#)

[External
Registers](#)

28/03/2023 16:02; 72747e43966d6b97dcbd230a1b3f0421d1ea3d94

Copyright Â© 2010-2023 Arm Limited or its affiliates. All rights reserved. This document is Non-Confidential.