TRBLIMITR_EL1, Trace Buffer Limit Address Register

The TRBLIMITR_EL1 characteristics are:

Purpose

Defines the top address for the trace buffer, and controls the trace buffer modes and enable.

Configuration

AArch64 System register TRBLIMITR_EL1 bits [63:0] are architecturally mapped to External register TRBLIMITR_EL1[63:0] when FEAT_TRBE_EXT is implemented.

This register is present only when FEAT_TRBE is implemented. Otherwise, direct accesses to TRBLIMITR_EL1 are undefined.

Attributes

TRBLIMITR_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
LIMIT																															
LIMIT																				RES0						XE	nVM	TM		FM		E
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

LIMIT, bits [63:12]

Trace Buffer Limit pointer address. (TRBLIMITR_EL1.LIMIT << 12) is the address of the last byte in the trace buffer plus one. Bits [11:0] of the Limit pointer address are always zero. If the smallest implemented translation granule is not 4KB, then TRBLIMITR_EL1[N-1:12] are res0, where N is the implementation defined value Log_2(smallest implemented translation granule).

The reset behavior of this field is:

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

Bits [11:7]

Reserved, res0.

XE, bit [6]
When FEAT_TRBE_EXT is implemented:

Trace Buffer Unit External mode enable. Used for save/restore of TRBLIMITR_EL1.XE.

XE	Meaning
0b0	Trace Buffer Unit is not enabled by this control.
0b1	If SelfHostedTraceEnabled() is FALSE, the Trace Buffer Unit is enabled.

Software must treat this field as UNK/SBZP when the OS Lock is unlocked.

The reset behavior of this field is:

- On a Cold reset, this field resets to 0.

Accessing this field has the following behavior:

- When !OSLockStatus(), access to this field is **RO**.
- Otherwise, access to this field is **RW**.

Otherwise:

Reserved, res0.

nVM, bit [5]

Address mode.

nVM	Meaning
0b0	The trace buffer pointers are virtual addresses.

0b1	The trace buffer pointers are: • Physical address in the owning security state if the owning translation regime has no stage 2 translation. • Intermediate physical addresses in the owning security state if the owning translation regime has stage 2 translations.

When FEAT_TRBE_EXT is implemented and SelfHostedTraceEnabled() == FALSE, the trace buffer pointers are always physical addresses.

The reset behavior of this field is:

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

Accessing this field has the following behavior:

- Access is **RES1** if all of the following are true:
  - FEAT_TRBE_EXT is implemented
  - !SelfHostedTraceEnabled()
- Otherwise, access to this field is **RW**.

TM, bits [4:3]

Trigger mode.

TM	Meaning
0b00	Stop on trigger. Flush then stop collection and raise maintenance interrupt on Trigger Event.
0b01	IRQ on trigger. Continue collection and raise maintenance interrupt on Trigger Event.
0b11	Ignore trigger. Continue collection and do not raise maintenance interrupt on Trigger Event.

All other values are reserved.

The reset behavior of this field is:

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

FM, bits [2:1]

Trace buffer mode.

FM	Meaning
0b00	Fill mode. Stop collection and raise maintenance interrupt on current write pointer wrap.
0b01	Wrap mode. Continue collection and raise maintenance interrupt on current write pointer wrap.
0b11	Circular Buffer mode. Continue collection and do not raise maintenance interrupt on current write pointer wrap.

All other values are reserved.

The reset behavior of this field is:

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

E, bit [0]

Trace Buffer Unit enable. Controls whether the Trace Buffer Unit is enabled when SelfHostedTraceEnabled() == TRUE.

E	Meaning
0b0	Trace Buffer Unit is not enabled by this control.
0b1	If SelfHostedTraceEnabled() is TRUE, the Trace Buffer Unit is enabled.

If FEAT_TRBE_EXT is implemented and SelfHostedTraceEnabled() == FALSE, then TRBLIMITR_EL1.XE controls whether the Trace Buffer Unit is enabled.

If FEAT_TRBE_EXT is not implemented, then the Trace Buffer Unit is disabled when SelfHostedTraceEnabled() == FALSE.

All output is discarded by the Trace Buffer Unit when the Trace Buffer Unit is disabled.

The reset behavior of this field is:

- On a Warm reset, this field resets to 0.

Accessing TRBLIMITR_EL1

The PE might ignore a write to TRBLIMITR_EL1 if all the following are true:

- TRBLIMITR_EL1.E == 1.
- Either FEAT_TRBE_EXT is not implemented or the Trace Buffer Unit is using Self-hosted mode.
- The write does not set TRBLIMITR_EL1.E to 0.

If FEAT_TRBE_EXT is implemented, the PE might ignore a write to TRBLIMITR_EL1 if all the following are true:

- TRBLIMITR_EL1.XE == 1.
- The Trace Buffer Unit is using External mode.
- The write does not set TRBLIMITR_EL1.XE to 0.

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

MRS <Xt>, TRBLIMITR_EL1

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

```
if PSTATE.EL == EL0 then
    UNDEFINED;
elsif PSTATE.EL == EL1 then
    if Halted() && HaveEL(EL3) && EDSCR.SDD == '1'
&& boolean IMPLEMENTATION_DEFINED "EL3 trap priority
when SDD == '1'" && (MDCR_EL3.NSTB[0] == '0' ||
MDCR_EL3.NSTB[1] != SCR_EL3.NS ||
(IsFeatureImplemented(FEAT_RME) && MDCR_EL3.NSTBE !=
SCR_EL3.NSE)) then
        UNDEFINED;
    elsif EL2Enabled() &&
IsFeatureImplemented(FEAT_FGT) && (!HaveEL(EL3) ||
SCR_EL3.FGTEn == '1') && HDFGRTR_EL2.TRBLIMITR_EL1
== '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    elsif EL2Enabled() && MDCR_EL2.E2TB == 'x0' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    elsif HaveEL(EL3) && (MDCR_EL3.NSTB[0] == '0' ||
MDCR_EL3.NSTB[1] != SCR_EL3.NS ||
(IsFeatureImplemented(FEAT_RME) && MDCR_EL3.NSTBE !=
SCR_EL3.NSE)) then
        if Halted() && EDSCR.SDD == '1' then
            UNDEFINED;
        else
            AArch64.SystemAccessTrap(EL3, 0x18);
    else
        X[t, 64] = TRBLIMITR_EL1;
```

```
elsif PSTATE.EL == EL2 then
    if Halted() && HaveEL(EL3) && EDSCR.SDD == '1'
&& boolean IMPLEMENTATION_DEFINED "EL3 trap priority
when SDD == '1'" && (MDCR_EL3.NSTB[0] == '0' ||
MDCR_EL3.NSTB[1] != SCR_EL3.NS ||
(IsFeatureImplemented(FEAT_RME) && MDCR_EL3.NSTBE !=
SCR_EL3.NSE)) then
        UNDEFINED;
    elsif HaveEL(EL3) && (MDCR_EL3.NSTB[0] == '0' ||
MDCR_EL3.NSTB[1] != SCR_EL3.NS ||
(IsFeatureImplemented(FEAT_RME) && MDCR_EL3.NSTBE !=
SCR_EL3.NSE)) then
        if Halted() && EDSCR.SDD == '1' then
            UNDEFINED;
        else
            AArch64.SystemAccessTrap(EL3, 0x18);
    else
        X[t, 64] = TRBLIMITR_EL1;
elsif PSTATE.EL == EL3 then
    X[t, 64] = TRBLIMITR_EL1;
```

MSR TRBLIMITR_EL1, <Xt>

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

```
if PSTATE.EL == EL0 then
    UNDEFINED;
elsif PSTATE.EL == EL1 then
    if Halted() && HaveEL(EL3) && EDSCR.SDD == '1'
&& boolean IMPLEMENTATION_DEFINED "EL3 trap priority
when SDD == '1'" && (MDCR_EL3.NSTB[0] == '0' ||
MDCR_EL3.NSTB[1] != SCR_EL3.NS ||
(IsFeatureImplemented(FEAT_RME) && MDCR_EL3.NSTBE !=
SCR_EL3.NSE)) then
        UNDEFINED;
    elsif EL2Enabled() &&
IsFeatureImplemented(FEAT_FGT) && (!HaveEL(EL3) ||
SCR_EL3.FGTEn == '1') && HDFGWTR_EL2.TRBLIMITR_EL1
== '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    elsif EL2Enabled() && MDCR_EL2.E2TB == 'x0' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    elsif HaveEL(EL3) && (MDCR_EL3.NSTB[0] == '0' ||
MDCR_EL3.NSTB[1] != SCR_EL3.NS ||
(IsFeatureImplemented(FEAT_RME) && MDCR_EL3.NSTBE !=
SCR_EL3.NSE)) then
        if Halted() && EDSCR.SDD == '1' then
            UNDEFINED;
        else
            AArch64.SystemAccessTrap(EL3, 0x18);
    else
        TRBLIMITR_EL1 = X[t, 64];
```

```
elsif PSTATE.EL == EL2 then
    if Halted() && HaveEL(EL3) && EDSCR.SDD == '1'
&& boolean IMPLEMENTATION_DEFINED "EL3 trap priority
when SDD == '1'" && (MDCR_EL3.NSTB[0] == '0' ||
MDCR_EL3.NSTB[1] != SCR_EL3.NS ||
(IsFeatureImplemented(FEAT_RME) && MDCR_EL3.NSTBE !=
SCR_EL3.NSE)) then
        UNDEFINED;
    elsif HaveEL(EL3) && (MDCR_EL3.NSTB[0] == '0' ||
MDCR_EL3.NSTB[1] != SCR_EL3.NS ||
(IsFeatureImplemented(FEAT_RME) && MDCR_EL3.NSTBE !=
SCR_EL3.NSE)) then
        if Halted() && EDSCR.SDD == '1' then
            UNDEFINED;
        else
            AArch64.SystemAccessTrap(EL3, 0x18);
    else
        TRBLIMITR_EL1 = X[t, 64];
elsif PSTATE.EL == EL3 then
    TRBLIMITR_EL1 = X[t, 64];
```

28/03/2023 16:02; 72747e43966d6b97dcbd230a1b3f0421d1ea3d94