TRCITEEDCR, Instrumentation Trace Extension External Debug Control Register

The TRCITEEDCR characteristics are:

Purpose

Controls instrumentation trace filtering.

Configuration

This register is present only when FEAT_ETE is implemented, FEAT_TRC_SR is implemented and TRCIDR0.ITE == 1. Otherwise, direct accesses to TRCITEEDCR are undefined.

Attributes

TRCITEEDCR 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																														RL	S	NS	E3	E2	E1	E0
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 [63:7]

Reserved, res0.

RL, bit [6]
When FEAT_RME is implemented:

Instrumentation Trace in Realm state.

RL	Meaning
0b0	Instrumentation trace prohibited in Realm state.
0b1	Instrumentation trace permitted in Realm state.

This field is ignored when `SelfHostedTraceEnabled`() returns TRUE.

This field is used in conjunction with TRCCONFIGR.ITO and TRCITEEDCR.E<m> to control whether Instrumentation trace is permitted or prohibited in Realm state.

The reset behavior of this field is:

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

Otherwise:

Reserved, res0.

S, bit [5]
When Secure state is implemented:

Instrumentation Trace in Secure state.

S	Meaning
0b0	Instrumentation trace prohibited in Secure state.
0b1	Instrumentation trace permitted in Secure state.

This field is ignored when SelfHostedTraceEnabled() returns TRUE.

When FEAT_RME is not implemented, this field is used in conjunction with TRCCONFIGR.ITO, TRCITEEDCR.E3, and TRCITEEDCR.E<m> to control whether Instrumentation trace is permitted or prohibited in Secure state.

When FEAT_RME is implemented, this field is used in conjunction with TRCCONFIGR.ITO and TRCITEEDCR.E<m> to control whether Instrumentation trace is permitted or prohibited in Secure state.

The reset behavior of this field is:

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

Otherwise:

Reserved, res0.

NS, bit [4]
When Any of Non-secure EL2, EL1, or EL0 are implemented:

Instrumentation Trace in Non-secure state.

NS	Meaning

0b0	Instrumentation trace prohibited in Non-secure state.
0b1	Instrumentation trace permitted in Non-secure state.

This field is ignored when SelfHostedTraceEnabled() returns TRUE.

This field is used in conjunction with TRCCONFIGR.ITO and TRCITEEDCR.E<m> to control whether Instrumentation trace is permitted or prohibited in Non-secure state.

The reset behavior of this field is:

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

Otherwise:

Reserved, res0.

E3, bit [3]
When EL3 is implemented:

Instrumentation Trace Enable at EL3.

E3	Meaning
0b0	Instrumentation trace prohibited at EL3.
0b1	Instrumentation trace permitted at EL3.

This field is ignored when SelfHostedTraceEnabled() returns TRUE.

When FEAT_RME is not implemented, TRCITEEDCR.E3 is used in conjunction with TRCCONFIGR.ITO and TRCITEEDCR.S to control whether Instrumentation trace is permitted or prohibited at EL3.

When FEAT_RME is implemented, TRCITEEDCR.E3 is used in conjunction with TRCCONFIGR.ITO to control whether Instrumentation trace is permitted or prohibited at EL3.

The reset behavior of this field is:

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

Otherwise:

Reserved, res0.

E<m>, bit [m], for m = 2 to 0

Instrumentation Trace Enable at EL<m>.

E<m>	Meaning
0b0	Instrumentation trace prohibited at EL<m>.
0b1	Instrumentation trace permitted at EL<m>.

This field is ignored when SelfHostedTraceEnabled() returns TRUE.

This bit is used in conjunction with TRCCONFIGR.ITO, TRCITEEDCR.NS, TRCITEEDCR.S, and TRCITEEDCR.RL to control whether Instrumentation trace is permitted or prohibited at EL<m> in the specified Security states.

TRCITEEDCR.E<2> is res0 if EL2 is not implemented in any Security states.

The reset behavior of this field is:

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

Accessing TRCITEEDCR

Writes are constrained unpredictable if the trace unit is not in the Idle state.

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

MRS <Xt>, TRCITEEDCR

op0	op1	CRn	CRm	op2
0b10	0b001	0b0000	0b0010	0b001

```
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'" && CPTR_EL3.TTA == '1' then
        UNDEFINED;
    elsif CPACR_EL1.TTA == '1' then
        AArch64.SystemAccessTrap(EL1, 0x18);
    elsif EL2Enabled() && CPTR_EL2.TTA == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
```

```
    elsif EL2Enabled() &&
IsFeatureImplemented(FEAT_FGT) && (!HaveEL(EL3) ||
SCR_EL3.FGTEn == '1') && HDFGRTR_EL2.TRC == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    elsif HaveEL(EL3) && CPTR_EL3.TTA == '1' then
        if Halted() && EDSCR.SDD == '1' then
            UNDEFINED;
        else
            AArch64.SystemAccessTrap(EL3, 0x18);
    else
        X[t, 64] = TRCITEEDCR;
elsif PSTATE.EL == EL2 then
    if Halted() && HaveEL(EL3) && EDSCR.SDD == '1'
&& boolean IMPLEMENTATION_DEFINED "EL3 trap priority
when SDD == '1'" && CPTR_EL3.TTA == '1' then
        UNDEFINED;
    elsif CPTR_EL2.TTA == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    elsif HaveEL(EL3) && CPTR_EL3.TTA == '1' then
        if Halted() && EDSCR.SDD == '1' then
            UNDEFINED;
        else
            AArch64.SystemAccessTrap(EL3, 0x18);
    else
        X[t, 64] = TRCITEEDCR;
elsif PSTATE.EL == EL3 then
    if CPTR_EL3.TTA == '1' then
        AArch64.SystemAccessTrap(EL3, 0x18);
    else
        X[t, 64] = TRCITEEDCR;
```

MSR TRCITEEDCR, <Xt>

op0	op1	CRn	CRm	op2
0b10	0b001	0b0000	0b0010	0b001

```
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'" && CPTR_EL3.TTA == '1' then
        UNDEFINED;
    elsif CPACR_EL1.TTA == '1' then
        AArch64.SystemAccessTrap(EL1, 0x18);
    elsif EL2Enabled() && CPTR_EL2.TTA == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    elsif EL2Enabled() &&
IsFeatureImplemented(FEAT_FGT) && (!HaveEL(EL3) ||
SCR_EL3.FGTEn == '1') && HDFGWTR_EL2.TRC == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    elsif HaveEL(EL3) && CPTR_EL3.TTA == '1' then
        if Halted() && EDSCR.SDD == '1' then
```

```
            UNDEFINED;
        else
            AArch64.SystemAccessTrap(EL3, 0x18);
    else
        TRCITEEDCR = X[t, 64];
elsif PSTATE.EL == EL2 then
    if Halted() && HaveEL(EL3) && EDSCR.SDD == '1'
&& boolean IMPLEMENTATION_DEFINED "EL3 trap priority
when SDD == '1'" && CPTR_EL3.TTA == '1' then
        UNDEFINED;
    elsif CPTR_EL2.TTA == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    elsif HaveEL(EL3) && CPTR_EL3.TTA == '1' then
        if Halted() && EDSCR.SDD == '1' then
            UNDEFINED;
        else
            AArch64.SystemAccessTrap(EL3, 0x18);
    else
        TRCITEEDCR = X[t, 64];
elsif PSTATE.EL == EL3 then
    if CPTR_EL3.TTA == '1' then
        AArch64.SystemAccessTrap(EL3, 0x18);
    else
        TRCITEEDCR = X[t, 64];
```

28/03/2023 16:02; 72747e43966d6b97dcbd230a1b3f0421d1ea3d94