# TRCIDR3, ID Register 3

The TRCIDR3 characteristics are:

## Purpose

Returns the base architecture of the trace unit.

## Configuration

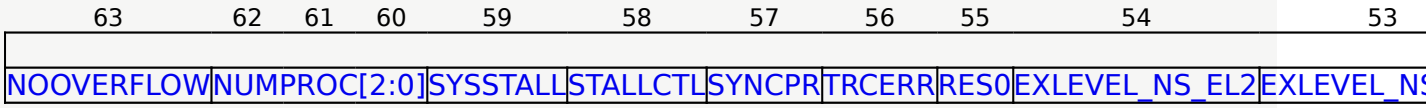
AArch64 System register TRCIDR3 bits [31:0] are architecturally mapped to External register TRCIDR3[31:0].

This register is present only when FEAT_ETE is implemented and FEAT_TRC_SR is implemented. Otherwise, direct accesses to TRCIDR3 are undefined.

## Attributes

TRCIDR3 is a 64-bit register.

## Field descriptions




### Bits [63:32]

Reserved, res0.

### NOOVERFLOW, bit [31]

Indicates if overflow prevention is implemented.

| NOOVERFLOW | Meaning |
|---|---|
| 0b0 | Overflow prevention is not implemented. |
| 0b1 | Overflow prevention is implemented. |

If TRCIDR3.STALLCTL == 0 then this field is 0.

**NUMPROC, bits [13:12, 30:28]**

Indicates the number of PEs available for tracing.

| NUMPROC | Meaning |
|---|---|
| 0b00000 | The trace unit can trace one PE. |

This field reads as 0b00000.

The NUMPROC field is split as follows:

- NUMPROC[2:0] is TRCIDR3[30:28].
- NUMPROC[4:3] is TRCIDR3[13:12].

**SYSSTALL, bit [27]**

Indicates if stalling of the PE is permitted.

| SYSSTALL | Meaning |
|---|---|
| 0b0 | Stalling of the PE is not permitted. |
| 0b1 | Stalling of the PE is permitted. |

The value of this field might be dynamic and change based on system conditions.

If TRCIDR3.STALLCTL == 0 then this field is 0.

**STALLCTL, bit [26]**

Indicates if trace unit implements stalling of the PE.

| STALLCTL | Meaning |
|---|---|
| 0b0 | Stalling of the PE is not implemented. |
| 0b1 | Stalling of the PE is implemented. |

**SYNCPR, bit [25]**

Indicates if an implementation has a fixed synchronization period.

| SYNCPR | Meaning |
|---|---|
| 0b0 | TRCSYNCPR is read/write so software can change the synchronization period. |
| 0b1 | TRCSYNCPR is read-only so the synchronization period is fixed. |

This field reads as 0.

**TRCERR, bit [24]**

Indicates forced tracing of System Error exceptions is implemented.

| TRCERR | Meaning |
|---|---|
| 0b0 | Forced tracing of System Error exceptions is not implemented. |
| 0b1 | Forced tracing of System Error exceptions is implemented. |

This field reads as 1.

**Bit [23]**

Reserved, res0.

**EXLEVEL_NS_EL2, bit [22]**

Indicates if Non-secure EL2 is implemented.

| EXLEVEL_NS_EL2 | Meaning |
|---|---|
| 0b0 | Non-secure EL2 is not implemented. |
| 0b1 | Non-secure EL2 is implemented. |

**EXLEVEL_NS_EL1, bit [21]**

Indicates if Non-secure EL1 is implemented.

| EXLEVEL_NS_EL1 | Meaning |
|---|---|
| 0b0 | Non-secure EL1 is not implemented. |
| 0b1 | Non-secure EL1 is implemented. |

**EXLEVEL_NS_EL0, bit [20]**

Indicates if Non-secure EL0 is implemented.

| EXLEVEL_NS_EL0 | Meaning |
|---|---|
| 0b0 | Non-secure EL0 is not implemented. |
| 0b1 | Non-secure EL0 is implemented. |

**EXLEVEL_S_EL3, bit [19]**

Indicates if EL3 is implemented.

| EXLEVEL_S_EL3 | Meaning |
|---|---|
| 0b0 | EL3 is not implemented. |
| 0b1 | EL3 is implemented. |

**EXLEVEL_S_EL2, bit [18]**

Indicates if Secure EL2 is implemented.

| EXLEVEL_S_EL2 | Meaning |
|---|---|
| 0b0 | Secure EL2 is not implemented. |
| 0b1 | Secure EL2 is implemented. |

**EXLEVEL_S_EL1, bit [17]**

Indicates if Secure EL1 is implemented.

| EXLEVEL_S_EL1 | Meaning |
|---|---|
| 0b0 | Secure EL1 is not implemented. |
| 0b1 | Secure EL1 is implemented. |

**EXLEVEL_S_EL0, bit [16]**

Indicates if Secure EL0 is implemented.

| EXLEVEL_S_EL0 | Meaning |
|---|---|
| 0b0 | Secure EL0 is not implemented. |
| 0b1 | Secure EL0 is implemented. |

**Bits [15:14]**

Reserved, res0.

**CCITMIN, bits [11:0]**

Indicates the minimum value that can be programmed in TRCCCCTLR.THRESHOLD.

If TRCIDR0.TRCCCI == 1 then the minimum value of this field is 0x001.

If TRCIDR0.TRCCCI == 0 then this field is zero.

## Accessing TRCIDR3

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

### MRS <Xt>, TRCIDR3

| op0 | op1 | CRn | CRm | op2 |
|---|---|---|---|---|
| 0b10 | 0b001 | 0b0000 | 0b1011 | 0b111 |

```
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.TRCID == '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] = TRCIDR3;
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] = TRCIDR3;
elsif PSTATE.EL == EL3 then
    if CPTR_EL3.TTA == '1' then
        AArch64.SystemAccessTrap(EL3, 0x18);
    else
```

```
            X[t, 64] = TRCIDR3;
```



28/03/2023 16:02; 72747e43966d6b97dcbd230a1b3f0421d1ea3d94