

# TRCVICTLR, ViewInst Main Control Register

The TRCVICTLR characteristics are:

## Purpose

Controls instruction trace filtering.

## Configuration

AArch64 System register TRCVICTLR bits [31:0] are architecturally mapped to External register [TRCVICTLR\[31:0\]](#).

This register is present only when FEAT\_ETE is implemented and FEAT\_TRC\_SR is implemented. Otherwise, direct accesses to TRCVICTLR are undefined.

## Attributes

TRCVICTLR is a 64-bit register.

## Field descriptions

|   |  |    |  |    |  |    |  |    |  |    |  |    |  |
|---|--|----|--|----|--|----|--|----|--|----|--|----|--|
| 6362616059  |  | 58 |  | 57 |  | 56 |  | 55 |  | 54 |  | 53 |  |
| <div>RES0EXLEVEL_RL_EL2EXLEVEL_RL_EL1EXLEVEL_RL_EL0RES0EXLEVEL_NS_EL2EXLEVEL_NS_EL1</div> |  |    |  |    |  |    |  |    |  |    |  |    |  |
| 3130292827  |  | 26 |  | 25 |  | 24 |  | 23 |  | 22 |  | 21 |  |

### Bits [63:27]

Reserved, res0.

### EXLEVEL\_RL\_EL2, bit [26]

When TRCIDR6.EXLEVEL\_RL\_EL2 == 1:

Filter instruction trace for EL2 in Realm state.

| EXLEVEL_RL_EL2 | Meaning |
|----------------|---------|
|----------------|---------|

|     |  |
|-----|--|
| 0b0 | When<br>TRCVICTLR.EXLEVEL_NS_EL2<br>is 0 the trace unit generates<br>instruction trace for EL2 in<br>Realm state.<br>When<br>TRCVICTLR.EXLEVEL_NS_EL2<br>is 1 the trace unit does not<br>generate instruction trace for<br>EL2 in Realm state. |
| 0b1 | When<br>TRCVICTLR.EXLEVEL_NS_EL2<br>is 0 the trace unit does not<br>generate instruction trace for<br>EL2 in Realm state.<br>When<br>TRCVICTLR.EXLEVEL_NS_EL2<br>is 1 the trace unit generates<br>instruction trace for EL2 in<br>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.

#### **EXLEVEL\_RL\_EL1, bit [25]**

**When TRCIDR6.EXLEVEL\_RL\_EL1 == 1:**

Filter instruction trace for EL1 in Realm state.

| <b>EXLEVEL_RL_EL1</b> | <b>Meaning</b>   |
|-----------------------|--|
| 0b0                   | When<br>TRCVICTLR.EXLEVEL_NS_EL1<br>is 0 the trace unit generates<br>instruction trace for EL1 in<br>Realm state.<br>When<br>TRCVICTLR.EXLEVEL_NS_EL1<br>is 1 the trace unit does not<br>generate instruction trace for<br>EL1 in Realm state. |

0b1

When  
TRCVICTLR.EXLEVEL\_NS\_EL1  
is 0 the trace unit does not  
generate instruction trace for  
EL1 in Realm state.

When  
TRCVICTLR.EXLEVEL\_NS\_EL1  
is 1 the trace unit generates  
instruction trace for EL1 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.

**EXLEVEL\_RL\_ELO, bit [24]**

**When TRCIDR6.EXLEVEL\_RL\_ELO == 1:**

Filter instruction trace for EL0 in Realm state.

| EXLEVEL_RL_ELO | Meaning  |
|----------------|--|
| 0b0            | When<br>TRCVICTLR.EXLEVEL_NS_ELO<br>is 0 the trace unit generates<br>instruction trace for EL0 in<br>Realm state.<br>When<br>TRCVICTLR.EXLEVEL_NS_ELO<br>is 1 the trace unit does not<br>generate instruction trace for<br>EL0 in Realm state. |
| 0b1            | When<br>TRCVICTLR.EXLEVEL_NS_ELO<br>is 0 the trace unit does not<br>generate instruction trace for<br>EL0 in Realm state.<br>When<br>TRCVICTLR.EXLEVEL_NS_ELO<br>is 1 the trace unit generates<br>instruction trace for EL0 in<br>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.

**Bit [23]**

Reserved, res0.

**EXLEVEL\_NS\_EL2, bit [22]**

**When Non-secure EL2 is implemented:**

Filter instruction trace for EL2 in Non-secure state.

| EXLEVEL_NS_EL2 | Meaning   |
|----------------|---|
| 0b0            | The trace unit generates instruction trace for EL2 in Non-secure state.         |
| 0b1            | The trace unit does not generate instruction trace for EL2 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.

**EXLEVEL\_NS\_EL1, bit [21]**

**When Non-secure EL1 is implemented:**

Filter instruction trace for EL1 in Non-secure state.

| EXLEVEL_NS_EL1 | Meaning   |
|----------------|---|
| 0b0            | The trace unit generates instruction trace for EL1 in Non-secure state. |

0b1

The trace unit does not generate instruction trace for EL1 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.

**EXLEVEL\_NS\_ELO, bit [20]**

**When Non-secure EL0 is implemented:**

Filter instruction trace for EL0 in Non-secure state.

| <b>EXLEVEL_NS_ELO</b> | <b>Meaning</b>  |
|-----------------------|---|
| 0b0                   | The trace unit generates instruction trace for EL0 in Non-secure state.         |
| 0b1                   | The trace unit does not generate instruction trace for EL0 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.

**EXLEVEL\_S\_EL3, bit [19]**

**When EL3 is implemented:**

Filter instruction trace for EL3.

| <b>EXLEVEL_S_EL3</b> | <b>Meaning</b>                                      |
|----------------------|---|
| 0b0                  | The trace unit generates instruction trace for EL3. |

|     |   |
|-----|---|
| 0b1 | The trace unit does not generate instruction trace for EL3. |
|-----|---|

The reset behavior of this field is:

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

**Otherwise:**

Reserved, res0.

**EXLEVEL\_S\_EL2, bit [18]**

**When EL2 is implemented and FEAT\_SEL2 is implemented:**

Filter instruction trace for EL2 in Secure state.

| EXLEVEL_S_EL2 | Meaning   |
|---------------|---|
| 0b0           | The trace unit generates instruction trace for EL2 in Secure state.         |
| 0b1           | The trace unit does not generate instruction trace for EL2 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.

**EXLEVEL\_S\_EL1, bit [17]**

**When Secure EL1 is implemented:**

Filter instruction trace for EL1 in Secure state.

| EXLEVEL_S_EL1 | Meaning   |
|---------------|---|
| 0b0           | The trace unit generates instruction trace for EL1 in Secure state. |

|     |   |
|-----|---|
| 0b1 | The trace unit does not generate instruction trace for EL1 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.

**EXLEVEL\_S\_EL0, bit [16]**

**When Secure EL0 is implemented:**

Filter instruction trace for EL0 in Secure state.

| EXLEVEL_S_EL0 | Meaning   |
|---------------|---|
| 0b0           | The trace unit generates instruction trace for EL0 in Secure state.         |
| 0b1           | The trace unit does not generate instruction trace for EL0 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.

**Bits [15:12]**

Reserved, res0.

**TRCERR, bit [11]**

**When TRCIDR3.TRCERR == 1:**

Controls the forced tracing of System Error exceptions.

| TRCERR | Meaning |
|--------|---------|
|--------|---------|

|     |  |
|-----|--|
| 0b0 | Forced tracing of System Error exceptions is disabled. |
| 0b1 | Forced tracing of System Error exceptions is enabled.  |

The reset behavior of this field is:

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

#### Otherwise:

Reserved, res0.

#### TRCRESET, bit [10]

Controls the forced tracing of PE Resets.

| TRCRESET | Meaning                                  |
|----------|--|
| 0b0      | Forced tracing of PE Resets is disabled. |
| 0b1      | Forced tracing of PE Resets is enabled.  |

The reset behavior of this field is:

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

#### SSSTATUS, bit [9]

ViewInst start/stop function status.

| SSSTATUS | Meaning   |
|----------|---|
| 0b0      | Stopped State.<br>The ViewInst start/stop function is in the stopped state. |
| 0b1      | Started State.<br>The ViewInst start/stop function is in the started state. |

Before software enables the trace unit, it must write to this field to set the initial state of the ViewInst start/stop function. If the ViewInst start/stop function is not used then set this field to 1. Arm recommends that the value of this field is set before each trace session begins.



If the trace unit becomes disabled while a start point or stop point is still speculative, then the value of TRCVICTLR.SSSTATUS is unknown and might represent the result of a speculative start point or stop point.

If software which is running on the PE being traced disables the trace unit, either by clearing [TRCPRGCTLR.EN](#) or locking the OS Lock, Arm recommends that a DSB and an ISB instruction are executed before disabling the trace unit to prevent any start points or stop points being speculative at the point of disabling the trace unit. This procedure assumes that all start points or stop points occur before the barrier instructions are executed. The procedure does not guarantee that there are no speculative start points or stop points when disabling, although it helps minimize the probability.

The reset behavior of this field is:

- On a Trace unit 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:
  - TRCIDR4.NUMACPAIRS == 0b0000
  - TRCIDR4.NUMPC == 0b0000
- Otherwise, access to this field is **RW**.

#### Bit [8]

Reserved, res0.

#### EVENT\_TYPE, bit [7]

When TRCIDR4.NUMRSPAIR != 0b0000:

Chooses the type of Resource Selector.

| EVENT_TYPE | Meaning  |
|------------|--|
| 0b0        | A single Resource Selector. TRCVICTLR.EVENT.SEL[4:0] selects the single Resource Selector, from 0-31, used to activate the resource event. |

0b1 A Boolean-combined pair of Resource Selectors. TRCVICTLR.EVENT.SEL[3:0] selects the Resource Selector pair, from 0-15, that has a Boolean function that is applied to it whose output is used to activate the resource event. TRCVICTLR.EVENT.SEL[4] is res0.

---

The reset behavior of this field is:

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

**Otherwise:**

Reserved, res0.

**Bits [6:5]**

Reserved, res0.

**Bits[4:0]**

**When TRCIDR4.NUMRSPAIR != 0b0000:**

**EVENT\_SEL, bits [4:0] of bits [4:0]**

Defines the selected Resource Selector or pair of Resource Selectors. TRCVICTLR.EVENT.TYPE controls whether TRCVICTLR.EVENT.SEL is the index of a single Resource Selector, or the index of a pair of Resource Selectors.

If an unimplemented Resource Selector is selected using this field, the behavior of the resource event is unpredictable, and the resource event might fire or might not fire when the resources are not in the Paused state.

Selecting Resource Selector pair 0 using this field is unpredictable, and the resource event might fire or might not fire when the resources are not in the Paused state.

The reset behavior of this field is:

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

**When TRCIDR4.NUMRSPAIR == 0b0000:**

**Reserved, bits [4:0] of bits [4:0]**

This field is reserved:

- Bits [4:1] are res0.
- Bit [0] is res1.

**Otherwise:**

Reserved, res0.

## Accessing TRCVICTLR

Must be programmed.

Reads from this register might return an unknown value if the trace unit is not in either of the Idle or Stable states.

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

**MRS <Xt>, TRCVICTLR**

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b10 | 0b001 | 0b0000 | 0b0000 | 0b010 |

```
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.TRCVICTLR ==
    '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] = TRCVICTLR;
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] = TRCVICTLR;
elsif PSTATE.EL == EL3 then
    if CPTR_EL3.TTA == '1' then
        AArch64.SystemAccessTrap(EL3, 0x18);
    else
        X[t, 64] = TRCVICTLR;

```

## MSR TRCVICTLR, <Xt>

| op0  | op1   | CRn    | CRm    | op2   |
|------|-------|--------|--------|-------|
| 0b10 | 0b001 | 0b0000 | 0b0000 | 0b010 |

```

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.TRCVICTLR ==
'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
            TRCVICTLR = 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
        TRCVICTLR = X[t, 64];
elsif PSTATE.EL == EL3 then
    if CPTR_EL3.TTA == '1' then
        AArch64.SystemAccessTrap(EL3, 0x18);
    else
        TRCVICTLR = X[t, 64];

```

---

[AArch32  
Registers](#)

[AArch64  
Registers](#)

[AArch32  
Instructions](#)

[AArch64  
Instructions](#)

[Index by  
Encoding](#)

[External  
Registers](#)

28/03/2023 16:02; 72747e43966d6b97dcbd230a1b3f0421d1ea3d94

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