# TPIDRRO_EL0, EL0 Read-Only Software Thread ID Register

The TPIDRRO_EL0 characteristics are:

## Purpose

Provides a location where software executing at EL1 or higher can store thread identifying information that is visible to software executing at EL0, for OS management purposes.

The PE makes no use of this register.

## Configuration

AArch64 System register TPIDRRO_EL0 bits [31:0] are architecturally mapped to AArch32 System register TPIDRURO[31:0].

## Attributes

TPIDRRO_EL0 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Thread ID | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Thread ID | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 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 [63:0]**

Thread ID. Thread identifying information stored by software running at this Exception level.

## Accessing TPIDRRO_EL0

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

```
MRS <Xt>, TPIDRRO_EL0
```

| op0 | op1 | CRn | CRm | op2 |
|---|---|---|---|---|

| 0b11 | 0b011 | 0b1101 | 0b0000 | 0b011 |
|---|---|---|---|---|

```
if PSTATE.EL == EL0 then
    if EL2Enabled() && HCR_EL2.<E2H,TGE> != '11' &&
IsFeatureImplemented(FEAT_FGT) && (!HaveEL(EL3) ||
SCR_EL3.FGTEn == '1') && HFGRTR_EL2.TPIDRRO_EL0 ==
'1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    else
        X[t, 64] = TPIDRRO_EL0;
elsif PSTATE.EL == EL1 then
    if EL2Enabled() &&
IsFeatureImplemented(FEAT_FGT) && (!HaveEL(EL3) ||
SCR_EL3.FGTEn == '1') && HFGRTR_EL2.TPIDRRO_EL0 ==
'1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    else
        X[t, 64] = TPIDRRO_EL0;
elsif PSTATE.EL == EL2 then
    X[t, 64] = TPIDRRO_EL0;
elsif PSTATE.EL == EL3 then
    X[t, 64] = TPIDRRO_EL0;
```

## MSR TPIDRRO_EL0, <Xt>

| op0 | op1 | CRn | CRm | op2 |
|---|---|---|---|---|
| 0b11 | 0b011 | 0b1101 | 0b0000 | 0b011 |

```
if PSTATE.EL == EL0 then
    UNDEFINED;
elsif PSTATE.EL == EL1 then
    if EL2Enabled() &&
IsFeatureImplemented(FEAT_FGT) && (!HaveEL(EL3) ||
SCR_EL3.FGTEn == '1') && HFGWTR_EL2.TPIDRRO_EL0 ==
'1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    else
        TPIDRRO_EL0 = X[t, 64];
elsif PSTATE.EL == EL2 then
    TPIDRRO_EL0 = X[t, 64];
elsif PSTATE.EL == EL3 then
    TPIDRRO_EL0 = X[t, 64];
```