# CNTPS_CTL_EL1, Counter-timer Physical Secure Timer Control Register

The CNTPS_CTL_EL1 characteristics are:

## Purpose

Control register for the secure physical timer, usually accessible at EL3 but configurably accessible at EL1 in Secure state.

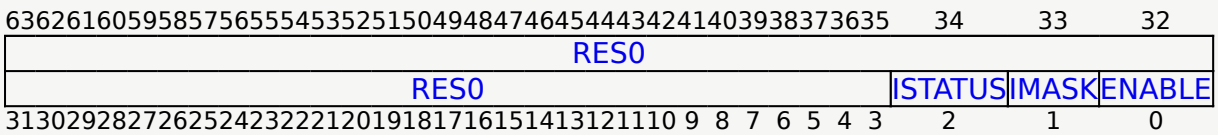
## Configuration

This register is present only when EL3 is implemented. Otherwise, direct accesses to CNTPS_CTL_EL1 are undefined.

## Attributes

CNTPS_CTL_EL1 is a 64-bit register.

## Field descriptions




### Bits [63:3]

Reserved, res0.

### ISTATUS, bit [2]

The status of the timer. This bit indicates whether the timer condition is met:

| ISTATUS | Meaning |
|---|---|
| 0b0 | Timer condition is not met. |
| 0b1 | Timer condition is met. |

When the value of the ENABLE bit is 1, ISTATUS indicates whether the timer condition is met. ISTATUS takes no account of the value of the IMASK bit. If the value of ISTATUS is 1 and the value of IMASK is 0 then the timer interrupt is asserted.

When the value of the ENABLE bit is 0, the ISTATUS field is unknown.

The reset behavior of this field is:

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

Access to this field is **RO**.

**IMASK, bit [1]**

Timer interrupt mask bit. Permitted values are:

| IMASK | Meaning |
|---|---|
| `0b0` | Timer interrupt is not masked by the IMASK bit. |
| `0b1` | Timer interrupt is masked by the IMASK bit. |

For more information, see the description of the ISTATUS bit.

The reset behavior of this field is:

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

**ENABLE, bit [0]**

Enables the timer. Permitted values are:

| ENABLE | Meaning |
|---|---|
| `0b0` | Timer disabled. |
| `0b1` | Timer enabled. |

Setting this bit to 0 disables the timer output signal, but the timer value accessible from CNTPS_TVAL_EL1 continues to count down.

---

**Note**

Disabling the output signal might be a power-saving option.

---

The reset behavior of this field is:

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

## Accessing CNTPS_CTL_EL1

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

### MRS <Xt>, CNTPS_CTL_EL1

| op0 | op1 | CRn | CRm | op2 |
|---|---|---|---|---|
| 0b11 | 0b111 | 0b1110 | 0b0010 | 0b001 |

```
if PSTATE.EL == EL0 then
    UNDEFINED;
elsif PSTATE.EL == EL1 then
    if HaveEL(EL3) && SCR_EL3.NS == '0' then
        if SCR_EL3.EEL2 == '1' then
            UNDEFINED;
        elsif SCR_EL3.ST == '0' then
            AArch64.SystemAccessTrap(EL3, 0x18);
        else
            X[t, 64] = CNTPS_CTL_EL1;
    else
        UNDEFINED;
elsif PSTATE.EL == EL2 then
    UNDEFINED;
elsif PSTATE.EL == EL3 then
    X[t, 64] = CNTPS_CTL_EL1;
```

### MSR CNTPS_CTL_EL1, <Xt>

| op0 | op1 | CRn | CRm | op2 |
|---|---|---|---|---|
| 0b11 | 0b111 | 0b1110 | 0b0010 | 0b001 |

```
if PSTATE.EL == EL0 then
    UNDEFINED;
elsif PSTATE.EL == EL1 then
    if HaveEL(EL3) && SCR_EL3.NS == '0' then
        if SCR_EL3.EEL2 == '1' then
            UNDEFINED;
        elsif SCR_EL3.ST == '0' then
            AArch64.SystemAccessTrap(EL3, 0x18);
        else
            CNTPS_CTL_EL1 = X[t, 64];
    else
        UNDEFINED;
elsif PSTATE.EL == EL2 then
    UNDEFINED;
```

```
elsif PSTATE.EL == EL3 then
    CNTPS_CTL_EL1 = X[t, 64];
```

---



28/03/2023 16:02; 72747e43966d6b97dcbd230a1b3f0421d1ea3d94