PMSWINC_EL0, Performance Monitors Software Increment Register

The PMSWINC_EL0 characteristics are:

Purpose

Increments a counter that is configured to count the Software increment event, event `0x00`. For more information, see 'SW_INCR'.

Configuration

AArch64 System register PMSWINC_EL0 bits [31:0] are architecturally mapped to AArch32 System register PMSWINC[31:0].

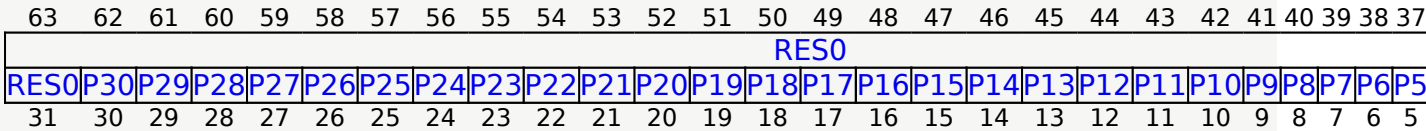
AArch64 System register PMSWINC_EL0 bits [31:0] are architecturally mapped to External register PMU.PMSWINC_EL0[31:0] when FEAT_PMUv3_EXT32 is implemented and FEAT_PMUv3p9 is not implemented.

This register is present only when FEAT_PMUv3 is implemented. Otherwise, direct accesses to PMSWINC_EL0 are undefined.

Attributes

PMSWINC_EL0 is a 64-bit register.

Field descriptions



Bits [63:31]

Reserved, res0.

P<m>, bit [m], for m = 30 to 0

Software increment.

P<m>	Meaning
0b0	Write is ignored.

0b1	Increment PMEVCNTR<m>_EL0, if PMEVCNTR<m>_EL0 is configured to count software increment events.

Accessing this field has the following behavior:

- This field ignores writes if any of the following are true:
  - All of the following are true:
    - FEAT_PMUv3p9 is implemented.
    - Accessed at EL0.
    - PMUSERENR_EL0.{UEN,SW} == {1,0}.
    - PMUACR_EL1.P<m> == 0.
  - All of the following are true:
    - EL2 is implemented and enabled in the current Security state.
    - m >= UInt(MDCR_EL2.HPMN).
    - Accessed at EL0 or EL1.
  - m >= UInt(PMCR_EL0.N).
- Otherwise access to this field is write-only.

Accessing PMSWINC_EL0

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

MSR PMSWINC_EL0, <Xt>

op0	op1	CRn	CRm	op2
0b11	0b011	0b1001	0b1100	0b100

```
if PSTATE.EL == EL0 then
    if Halted() && HaveEL(EL3) && EDSCR.SDD == '1'
&& boolean IMPLEMENTATION_DEFINED "EL3 trap priority
when SDD == '1'" && MDCR_EL3.TPM == '1' then
        UNDEFINED;
    elsif PMUSERENR_EL0.<SW,EN> == '00' then
        if EL2Enabled() && HCR_EL2.TGE == '1' then
            AArch64.SystemAccessTrap(EL2, 0x18);
        else
            AArch64.SystemAccessTrap(EL1, 0x18);
    elsif EL2Enabled() && HCR_EL2.<E2H,TGE> != '11'
&& IsFeatureImplemented(FEAT_FGT) && (!HaveEL(EL3)
|| SCR_EL3.FGTEn == '1') && HDFGWTR_EL2.PMSWINC_EL0
== '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    elsif EL2Enabled() && MDCR_EL2.TPM == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    elsif HaveEL(EL3) && MDCR_EL3.TPM == '1' then
```

```
            if Halted() && EDSCR.SDD == '1' then
                UNDEFINED;
            else
                AArch64.SystemAccessTrap(EL3, 0x18);
        else
            PMSWINC_EL0 = X[t, 64];
    elsif PSTATE.EL == EL1 then
        if Halted() && HaveEL(EL3) && EDSCR.SDD == '1'
    && boolean IMPLEMENTATION_DEFINED "EL3 trap priority
    when SDD == '1'" && MDCR_EL3.TPM == '1' then
            UNDEFINED;
        elsif EL2Enabled() &&
    IsFeatureImplemented(FEAT_FGT) && (!HaveEL(EL3) ||
    SCR_EL3.FGTEn == '1') && HDFGWTR_EL2.PMSWINC_EL0 ==
    '1' then
            AArch64.SystemAccessTrap(EL2, 0x18);
        elsif EL2Enabled() && MDCR_EL2.TPM == '1' then
            AArch64.SystemAccessTrap(EL2, 0x18);
        elsif HaveEL(EL3) && MDCR_EL3.TPM == '1' then
            if Halted() && EDSCR.SDD == '1' then
                UNDEFINED;
            else
                AArch64.SystemAccessTrap(EL3, 0x18);
        else
            PMSWINC_EL0 = X[t, 64];
    elsif PSTATE.EL == EL2 then
        if Halted() && HaveEL(EL3) && EDSCR.SDD == '1'
    && boolean IMPLEMENTATION_DEFINED "EL3 trap priority
    when SDD == '1'" && MDCR_EL3.TPM == '1' then
            UNDEFINED;
        elsif HaveEL(EL3) && MDCR_EL3.TPM == '1' then
            if Halted() && EDSCR.SDD == '1' then
                UNDEFINED;
            else
                AArch64.SystemAccessTrap(EL3, 0x18);
        else
            PMSWINC_EL0 = X[t, 64];
    elsif PSTATE.EL == EL3 then
        PMSWINC_EL0 = X[t, 64];
```


28/03/2023 16:02; 72747e43966d6b97dcbd230a1b3f0421d1ea3d94