PMCNTENSET_EL0, Performance Monitors Count Enable Set Register

The PMCNTENSET_EL0 characteristics are:

Purpose

Enables the Cycle Count Register, PMCCNTR_EL0, and any implemented event counters PMEVCNTR<n>_EL0. Reading this register shows which counters are enabled.

Configuration

AArch64 System register PMCNTENSET_EL0 bits [31:0] are architecturally mapped to AArch32 System register PMCNTENSET[31:0].

AArch64 System register PMCNTENSET_EL0 bits [31:0] are architecturally mapped to External register PMU.PMCNTENSET_EL0[31:0].

AArch64 System register PMCNTENSET_EL0 bits [63:32] are architecturally mapped to External register PMU.PMCNTENSET_EL0[63:32] when FEAT_PMUv3_EXT64 is implemented.

This register is present only when FEAT_PMUv3 is implemented. Otherwise, direct accesses to PMCNTENSET_EL0 are undefined.

Attributes

PMCNTENSET_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
RES0																											

C	P30	P29	P28	P27	P26	P25	P24	P23	P22	P21	P20	P19	P18	P17	P16	P15	P14	P13	P12	P11	P10	P9	P8	P7	P6	P5	P4
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

Bits [63:33]

Reserved, res0.

F<m>, bit [m+32], for m = 0
When FEAT_PMUv3_ICNTR is implemented:

Fixed-function counter <m> enable. On writes, allows software to enable fixed-function counter <m>. On reads, returns the fixed-function counter <m> enable status.

F<m>	Meaning
0b0	Fixed-function counter <m> disabled.
0b1	Fixed-function counter <m> enabled.

PMCNTENSET_EL0.F0 holds the enable status for PMICNTR_EL0.

Accessing this field has the following behavior:

- This field reads-as-zero if all of the following are true:
  - Any of the following are true:
    - EL3 is implemented and SCR_EL3.FGTEn2 == 0.
    - HDFGRTR2_EL2.nPMICFILTR_EL0 == 0.
  - FEAT_FGT2 is implemented.
  - EL2 is implemented and enabled in the current Security state.
  - Accessed at EL1 or EL0.
  - HCR_EL2.{E2H,TGE} != {1,1}.
- This field reads-as-zero and ignores writes if any of the following are true:
  - All of the following are true:
    - EL3 is implemented.
    - MDCR_EL3.EnPM2 == 0.
    - Accessed at EL2, EL1, or EL0.
  - All of the following are true:
    - PMUSERENR_EL0.UEN == 0 or PMUACR_EL1.F<m> == 0.
    - Accessed at EL0.
- This field ignores writes if any of the following are true:
  - All of the following are true:
    - EL3 is implemented and SCR_EL3.FGTEn2 == 0, or HDFGWTR2_EL2.nPMICFILTR_EL0 == 0.
    - FEAT_FGT2 is implemented.
    - EL2 is implemented and enabled in the current Security state.
    - Accessed at EL1 or EL0.
    - HCR_EL2.{E2H,TGE} != {1,1}.
  - All of the following are true:
    - Accessed at EL0.
    - PMUSERENR_EL0.IR == 1.
- Otherwise access to this field is W1S.

The reset behavior of this field is:

- On a Warm reset, this field resets to `0`.

Otherwise:

Reserved, res0.

C, bit [31]

PMCCNTR_EL0 enable. On writes, allows software to enable PMCCNTR_EL0. On reads, returns the PMCCNTR_EL0 enable status.

C	Meaning
`0b0`	PMCCNTR_EL0 disabled.
`0b1`	PMCCNTR_EL0 enabled.

Accessing this field has the following behavior:

- This field reads-as-zero and ignores writes if all of the following are true:
  - FEAT_PMUv3p9 is implemented.
  - Accessed at EL0.
  - PMUSERENR_EL0.UEN == 1.
  - PMUACR_EL1.C == 0.
- This field ignores writes if all of the following are true:
  - FEAT_PMUv3p9 is implemented.
  - Accessed at EL0.
  - PMUSERENR_EL0.{UEN,CR} == {1,1}.

The reset behavior of this field is:

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

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

PMEVCNTR<m>_EL0 enable. On writes, allows software to enable PMEVCNTR<m>_EL0. On reads, returns the PMEVCNTR<m>_EL0 enable status.

P<m>	Meaning
`0b0`	PMEVCNTR<m>_EL0 disabled.
`0b1`	PMEVCNTR<m>_EL0 enabled.

Accessing this field has the following behavior:

- This field reads-as-zero and 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 == 1.
    - 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).
- This field ignores writes if all of the following are true:
  - FEAT_PMUv3p9 is implemented.
  - Accessed at EL0.
  - PMUSERENR_EL0.{UEN,ER} == {1,1}.
- Otherwise access to this field is W1S.

The reset behavior of this field is:

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

Accessing PMCNTENSET_EL0

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

MRS <Xt>, PMCNTENSET_EL0

op0	op1	CRn	CRm	op2
0b11	0b011	0b1001	0b1100	0b001

```
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.EN == '0' 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') && HDFGRTR_EL2.PMCNTEN ==
'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
            X[t, 64] = PMCNTENSET_EL0;
    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') && HDFGRTR_EL2.PMCNTEN == '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
            X[t, 64] = PMCNTENSET_EL0;
    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
            X[t, 64] = PMCNTENSET_EL0;
    elsif PSTATE.EL == EL3 then
        X[t, 64] = PMCNTENSET_EL0;
```

MSR PMCNTENSET_EL0, <Xt>

op0	op1	CRn	CRm	op2
0b11	0b011	0b1001	0b1100	0b001

```
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.EN == '0' 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.PMCNTEN ==
    '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
            PMCNTENSET_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.PMCNTEN == '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
            PMCNTENSET_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
            PMCNTENSET_EL0 = X[t, 64];
    elsif PSTATE.EL == EL3 then
        PMCNTENSET_EL0 = X[t, 64];
```

28/03/2023 16:02; 72747e43966d6b97dcbd230a1b3f0421d1ea3d94