PMOVSCLR_EL0, Performance Monitors Overflow Flag Status Clear Register

The PMOVSCLR_EL0 characteristics are:

Purpose

Contains the state of the overflow bit for the Cycle Count Register, PMCCNTR_EL0, and each of the implemented event counters PMEVCNTR<n>_EL0. Writing to this register clears these bits.

Configuration

AArch64 System register PMOVSCLR_EL0 bits [31:0] are architecturally mapped to AArch32 System register PMOVSR[31:0].

AArch64 System register PMOVSCLR_EL0 bits [31:0] are architecturally mapped to External register PMU.PMOVSCLR_EL0[31:0].

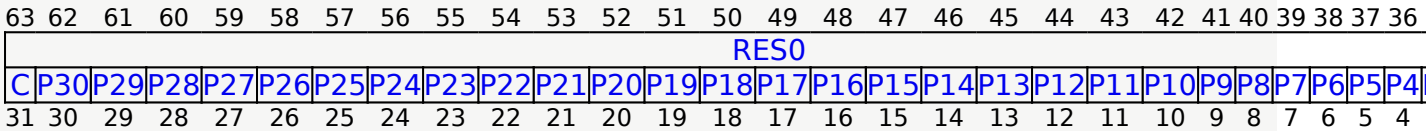
AArch64 System register PMOVSCLR_EL0 bits [63:32] are architecturally mapped to External register PMU.PMOVSCLR_EL0[63:32] when FEAT_PMUv3p9 is implemented or FEAT_PMUv3_EXT64 is implemented.

This register is present only when FEAT_PMUv3 is implemented. Otherwise, direct accesses to PMOVSCLR_EL0 are undefined.

Attributes

PMOVSCLR_EL0 is a 64-bit register.

Field descriptions



Bits [63:33]

Reserved, res0.

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

Unsigned overflow flag for fixed-function counter <m> clear. On writes, allows software to clear the unsigned overflow flag for fixed-function counter <m> to 0. On reads, returns the unsigned overflow flag for fixed-function counter <m> overflow status.

F<m>	Meaning
0b0	Fixed-function counter <m> has not overflowed.
0b1	Fixed-function counter <m> has overflowed.

PMOVSCLR_EL0.F0 holds the overflow 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 W1C.

The reset behavior of this field is:

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

Otherwise:

Reserved, res0.

C, bit [31]

Unsigned overflow flag for PMCCNTR_EL0 clear. On writes, allows software to clear the unsigned overflow flag for PMCCNTR_EL0 to 0. On reads, returns the unsigned overflow flag for PMCCNTR_EL0 overflow status.

C	Meaning
`0b0`	PMCCNTR_EL0 has not overflowed.
`0b1`	PMCCNTR_EL0 has overflowed.

PMCR_EL0.LC controls whether an overflow is detected from unsigned overflow of PMCCNTR_EL0[31:0] or unsigned overflow of PMCCNTR_EL0[63:0].

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}.
- Otherwise access to this field is W1C.

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

Unsigned overflow flag for PMEVCNTR<m>_EL0 clear. On writes, allows software to clear the unsigned overflow flag for PMEVCNTR<m>_EL0 to 0. On reads, returns the unsigned overflow flag for PMEVCNTR<m>_EL0 overflow status.

P<m>	Meaning

0b0	PMEVCNTR<m>_EL0 has not overflowed.
0b1	PMEVCNTR<m>_EL0 has overflowed.

If FEAT_PMUv3p5 is implemented, MDCR_EL2.HLP and PMCR_EL0.LP control whether an overflow is detected from unsigned overflow of PMEVCNTR<m>_EL0[31:0] or unsigned overflow of PMEVCNTR<m>_EL0[63:0].

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 W1C.

The reset behavior of this field is:

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

Accessing PMOVSCLR_EL0

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

MRS <Xt>, PMOVSCLR_EL0

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

```
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.PMOVS == '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] = PMOVSCLR_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.PMOVS == '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] = PMOVSCLR_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] = PMOVSCLR_EL0;
elsif PSTATE.EL == EL3 then
    X[t, 64] = PMOVSCLR_EL0;
```

MSR PMOVSCLR_EL0, <Xt>

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

```
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.PMOVS == '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
        PMOVSCLR_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.PMOVS == '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
        PMOVSCLR_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
            PMOVSCLR_EL0 = X[t, 64];
    elsif PSTATE.EL == EL3 then
        PMOVSCLR_EL0 = X[t, 64];
```

28/03/2023 16:01; 72747e43966d6b97dcbd230a1b3f0421d1ea3d94