

## PMOVSSET\_EL0, Performance Monitors Overflow Flag Status Set Register

The PMOVSSET\_EL0 characteristics are:

### Purpose

Sets the state of the overflow bit for the Cycle Count Register, [PMCCNTR\\_EL0](#), and each of the implemented event counters [PMEVCNTR<n>\\_EL0](#).

### Configuration

AArch64 System register PMOVSSET\_EL0 bits [31:0] are architecturally mapped to AArch32 System register [PMOVSSET\[31:0\]](#).

AArch64 System register PMOVSSET\_EL0 bits [31:0] are architecturally mapped to External register [PMU.PMOVSSET\\_EL0\[31:0\]](#).

AArch64 System register PMOVSSET\_EL0 bits [63:32] are architecturally mapped to External register [PMU.PMOVSSET\\_EL0\[63:32\]](#) when FEAT\_PMUv3\_EXT64 is implemented.

This register is present only when FEAT\_PMUv3 is implemented. Otherwise, direct accesses to PMOVSSET\_EL0 are undefined.

### Attributes

PMOVSSET\_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:**

Unsigned overflow flag for fixed-function counter <m> set. On writes, allows software to set the unsigned overflow flag for fixed-function counter <m> to 1. On reads, returns the unsigned overflow flag for fixed-function counter <m> overflow status.

<b>F&lt;m&gt;</b>	<b>Meaning</b>
0b0	Fixed-function counter <m> has not overflowed.
0b1	Fixed-function counter <m> has overflowed.

PMOVSSET\_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 W1S.

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\\_ELO](#) set. On writes, allows software to set the unsigned overflow flag for [PMCCNTR\\_ELO](#) to 1. On reads, returns the unsigned overflow flag for [PMCCNTR\\_ELO](#) overflow status.

C	Meaning
0b0	<a href="#">PMCCNTR_ELO</a> has not overflowed.
0b1	<a href="#">PMCCNTR_ELO</a> has overflowed.

[PMCR\\_ELO](#).LC controls whether an overflow is detected from unsigned overflow of [PMCCNTR\\_ELO](#)[31:0] or unsigned overflow of [PMCCNTR\\_ELO](#)[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\\_ELO](#).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\\_ELO](#).{UEN,CR} == {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.

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

Unsigned overflow flag for [PMEVCNTR<m>\\_ELO](#) set. On writes, allows software to set the unsigned overflow flag for [PMEVCNTR<m>\\_ELO](#) to 1. On reads, returns the unsigned overflow flag for [PMEVCNTR<m>\\_ELO](#) overflow status.

P<m>	Meaning
------	---------

0b0	<a href="#">PMEVCNTR&lt;m&gt;_EL0</a> has not overflowed.
0b1	<a href="#">PMEVCNTR&lt;m&gt;_EL0</a> 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 W1S.

The reset behavior of this field is:

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

## Accessing PMOVSSET\_EL0

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

MRS <Xt>, PMOVSSET\_EL0

op0	op1	CRn	CRm	op2
0b11	0b011	0b1001	0b1110	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;
    elseif PMUSERENR_EL0.EN == '0' then
        if EL2Enabled() && HCR_EL2.TGE == '1' then
            AArch64.SystemAccessTrap(EL2, 0x18);
        else
            AArch64.SystemAccessTrap(EL1, 0x18);
        elseif EL2Enabled() && HCR_EL2.<E2H,TGE> != '11'
&& IsFeatureImplemented(FEAT_FGT) && (!HaveEL(EL3)
|| SCR_EL3.FGTEn == '1') && HDFGRTR_EL2.PMOVS == '1'
then
    AArch64.SystemAccessTrap(EL2, 0x18);
    elseif EL2Enabled() && MDCR_EL2.TPM == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    elseif HaveEL(EL3) && MDCR_EL3.TPM == '1' then
        if Halted() && EDSCR.SDD == '1' then
            UNDEFINED;
        else
            AArch64.SystemAccessTrap(EL3, 0x18);
    else
        X[t, 64] = PMOVSSET_EL0;
elseif 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;
    elseif EL2Enabled() &&
IsFeatureImplemented(FEAT_FGT) && (!HaveEL(EL3) ||
SCR_EL3.FGTEn == '1') && HDFGRTR_EL2.PMOVS == '1'
then
        AArch64.SystemAccessTrap(EL2, 0x18);
    elseif EL2Enabled() && MDCR_EL2.TPM == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    elseif HaveEL(EL3) && MDCR_EL3.TPM == '1' then
        if Halted() && EDSCR.SDD == '1' then
            UNDEFINED;
        else
            AArch64.SystemAccessTrap(EL3, 0x18);
    else
        X[t, 64] = PMOVSSET_EL0;
elseif 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;
    elseif HaveEL(EL3) && MDCR_EL3.TPM == '1' then
        if Halted() && EDSCR.SDD == '1' then
            UNDEFINED;
        else
            AArch64.SystemAccessTrap(EL3, 0x18);
    else
        X[t, 64] = PMOVSSET_EL0;
elseif PSTATE.EL == EL3 then
    X[t, 64] = PMOVSSET_EL0;

```

## MSR PMOVSSET\_EL0, <Xt>

op0	op1	CRn	CRm	op2
0b11	0b011	0b1001	0b1110	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
                PMOVSSET_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
                PMOVSSET_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
            PMOVSSET_EL0 = X[t, 64];
    elsif PSTATE.EL == EL3 then
        PMOVSSET_EL0 = X[t, 64];
```

---

[AArch32  
Registers](#)

[AArch64  
Registers](#)

[AArch32  
Instructions](#)

[AArch64  
Instructions](#)

[Index by  
Encoding](#)

[External  
Registers](#)

28/03/2023 16:02; 72747e43966d6b97dcbd230a1b3f0421d1ea3d94

Copyright Â© 2010-2023 Arm Limited or its affiliates. All rights reserved. This document is Non-Confidential.