PMUSERENR_EL0, Performance Monitors User Enable Register

The PMUSERENR_EL0 characteristics are:

Purpose

Enables or disables EL0 access to the Performance Monitors.

Configuration

AArch64 System register PMUSERENR_EL0 bits [31:0] are architecturally mapped to AArch32 System register PMUSERENR[31:0].

This register is present only when FEAT_PMUv3 is implemented. Otherwise, direct accesses to PMUSERENR_EL0 are undefined.

Attributes

PMUSERENR_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 35 34 33 32
RES0

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	3	2	1	0
RES0	TID	IR	UEN	ER	CR	SW	EN

Bits [63:7]

Reserved, res0.

TID, bit [6]
When FEAT_PMUv3p9 is implemented:

Trap ID registers. Traps EL0 read access to common event identification registers.

TID	Meaning
`0b0`	Accesses to PMCEID<n>_EL0 and PMCEID<n> are not trapped by this mechanism.
`0b1`	EL0 read accesses to PMCEID<n>_EL0 and PMCEID<n> are trapped.

In AArch64 state, the register accesses affected by this control are:

- `MRS` reads of PMCEID0_EL0 and PMCEID1_EL0.

In AArch32 state, the register accesses affected by this control are:

- `MRC` reads of PMCEID0, PMCEID1, PMCEID2, and PMCEID3.

When trapped, reads generate an exception to EL1, or to EL2 when EL2 is implemented and enabled for the current Security state and HCR_EL2.TGE is 1, and:

- AArch64 `MRS` reads are reported using EC syndrome value `0x18`.
- AArch32 `MRC` reads are reported using EC syndrome value `0x03`.

The reset behavior of this field is:

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

Otherwise:

Reserved, res0.

IR, bit [5]
When FEAT_PMUv3_ICNTR is implemented:

Instruction counter Read-only.

When PMUSERENR_EL0.UEN is 1, EL0 reads of the instruction counter and EL0 writes to PMZR_EL0 are enabled by PMUSERENR_EL0.UEN, unless trapped by another control, and PMUSERENR_EL0.IR controls the behavior of EL0 writes to the instruction counter and PMZR_EL0.

IR	Meaning
`0b0`	Permitted EL0 writes are not affected by this mechanism.
`0b1`	Permitted EL0 writes to the instruction counter and PMZR_EL0.F0 are ignored.

In AArch64 state, when PMUSERENR_EL0.UEN is 1, `MSR` writes to PMZR_EL0 and PMICNTR_EL0 are affected by this control.

Ignored writes are not trapped and do not generate an exception.

This field is ignored by the PE when PMUSERENR_EL0.UEN == 0.

The reset behavior of this field is:

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

Otherwise:

Reserved, res0.

UEN, bit [4]
When FEAT_PMUv3p9 is implemented:

User Enable, with access controlled by PMUACR_EL1.

Enables EL0 read/write access to PMU registers, other than PMCR_EL0. Software can restrict permitted accesses using PMUSERENR_EL0.{IR,ER,CR} and PMUACR_EL1.

UEN	Meaning
`0b0`	If FEAT_PMUv3_ICNTR is implemented, then EL0 accesses to PMICFILTR_EL0, PMICFILTR_EL0, PMICNTR_EL0, and PMICNTR_EL0 are trapped. EL0 accesses to other specified PMU registers are not affected by this control.
`0b1`	EL0 accesses to PMU registers are enabled, unless trapped by another control. The behavior of permitted accesses is controlled by PMUSERENR_EL0.{IR,ER,CR,SW} and PMUACR_EL1.

In AArch64 state, the register accesses affected by this control are:

- `MRS` or `MSR` accesses to the following registers:
  - PMCCFILTR_EL0, PMCCNTR_EL0, PMCNTENCLR_EL0, PMCNTENSET_EL0, PMEVCNTR<n>_EL0, PMEVTYPER<n>_EL0, PMOVSCLR_EL0, PMOVSSET_EL0, PMSELR_EL0, PMXEVCNTR_EL0, and PMXEVTYPER_EL0.
  - If FEAT_PMUv3_ICNTR is implemented, PMICFILTR_EL0 and PMICNTR_EL0.
- `MRS` reads of PMCEID0_EL0 and PMCEID1_EL0.
- `MSR` writes to PMSWINC_EL0 and PMZR_EL0.

In AArch32 state, the register accesses affected by this control are:

- `MRC` or `MCR` accesses to PMCCFILTR, PMCCNTR, PMCNTENCLR, PMCNTENSET, PMEVCNTR<n>, PMEVTYPER<n>, PMOVSR, PMOVSSET, PMSELR, PMXEVCNTR, and PMXEVTYPER.

- `MRC` reads of PMCEID0, PMCEID1, PMCEID2, and PMCEID3.

- MCR writes to PMSWINC.
-
- MRRC or MCRR accesses to PMCCNTR.

When trapped, reads and writes generate an exception to EL1, or to EL2 when EL2 is implemented and enabled for the current Security state and HCR_EL2.TGE is 1, and:

- AArch64 MRS and MSR accesses are reported using EC syndrome value 0x18.
- AArch32 MRC and MCR accesses are reported using EC syndrome value 0x03.
- AArch32 MRRC and MCRR accesses are reported using EC syndrome value 0x04.

This field is ignored by the PE and treated as zero when EL1 is using AArch32.

The reset behavior of this field is:

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

Otherwise:

Reserved, res0.

ER, bit [3]
When FEAT_PMUv3p9 is implemented:

Event counters Read enable or Read-only.

When PMUSERENR_EL0.{UEN,EN} is {0,0}, PMUSERENR_EL0.ER enables EL0 reads of the event counters and EL0 reads and writes of the select register.

When PMUSERENR_EL0.UEN is 1, EL0 reads of the event counters and EL0 writes to PMZR_EL0 are enabled by PMUSERENR_EL0.UEN, unless trapped by another control, and PMUSERENR_EL0.ER controls the behavior of EL0 writes to the event counters and PMZR_EL0.

ER	Meaning
0b0	When PMUSERENR_EL0.UEN == 0, EL0 reads of the event counters and EL0 reads and writes of the select register are disabled, unless enabled by PMUSERENR_EL0.EN. When PMUSERENR_EL0.UEN == 1, permitted EL0 writes are not affected by this mechanism.

`0b1`	When PMUSERENR_EL0.UEN == 0, EL0 reads of the event counters and EL0 reads and writes of the select register are enabled, unless trapped by another control. When PMUSERENR_EL0.UEN == 1, permitted EL0 writes to the event counters and PMZR_EL0.P[30:0] are ignored.

In AArch64 state, the register accesses affected by this control are:

- When PMUSERENR_EL0.{UEN,EN} is {0,0}:
    - `MRS` reads of PMEVCNTR<n>_EL0 and PMXEVCNTR_EL0.
    - `MRS` and `MSR` accesses to PMSELR_EL0.
- When PMUSERENR_EL0.UEN is 1, `MSR` writes to PMZR_EL0, PMEVCNTR<n>_EL0 and PMXEVCNTR_EL0.

In AArch32 state, the register accesses affected by this control are:

- When PMUSERENR_EL0.{UEN,EN} is {0,0}:
    - `MRC` reads of PMEVCNTR<n> and PMXEVCNTR.
    - `MRC` and `MCR` accesses to PMSELR.
- When PMUSERENR_EL0.UEN is 1, `MCR` writes to PMEVCNTR<n> and PMXEVCNTR.

When disabled, reads and writes generate an exception to EL1, or to EL2 when EL2 is implemented and enabled for the current Security state and HCR_EL2.TGE is 1, and:

- AArch64 `MRS` and `MSR` accesses are reported using EC syndrome value `0x18`.
- AArch32 `MRC` and `MCR` accesses are reported using EC syndrome value `0x03`.

Ignored writes are not trapped and do not generate an exception.

This field is ignored by the PE when PMUSERENR_EL0.{UEN,EN} == {0,1}.

The reset behavior of this field is:

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

Otherwise:

Event counters Read enable.

When PMUSERENR_EL0.EN is 0, PMUSERENR_EL0.ER enables EL0 reads of the event counters and EL0 reads and writes of the select register.

ER	Meaning
`0b0`	EL0 reads of the event counters and EL0 reads and writes of the select register are disabled, unless enabled by PMUSERENR_EL0.EN.
`0b1`	EL0 reads of the event counters and EL0 reads and writes of the select register are enabled, unless trapped by another control.

In AArch64 state, the register accesses affected by this control are:

- `MRS` reads of PMEVCNTR<n>_EL0 and PMXEVCNTR_EL0.
- `MRS` and `MSR` accesses to PMSELR_EL0.

In AArch32 state, the register accesses affected by this control are:

- `MRC` reads of PMEVCNTR<n> and PMXEVCNTR.
- `MRC` and `MCR` accesses to PMSELR.

When disabled, reads and writes generate an exception to EL1, or to EL2 when EL2 is implemented and enabled for the current Security state and HCR_EL2.TGE is 1, and:

- AArch64 `MRS` and `MSR` accesses are reported using EC syndrome value `0x18`.
- AArch32 `MRC` and `MCR` accesses are reported using EC syndrome value `0x03`.

This field is ignored by the PE when PMUSERENR_EL0.EN == 1.

The reset behavior of this field is:

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

CR, bit [2]
When FEAT_PMUv3p9 is implemented:

Cycle counter Read enable or Read-only.

When PMUSERENR_EL0.{UEN,EN} is {0,0}, PMUSERENR_EL0.CR enables EL0 reads of the cycle counter.

When PMUSERENR_EL0.UEN is 1, EL0 reads of the cycle counter and EL0 writes to PMZR_EL0 are enabled by PMUSERENR_EL0.UEN, unless trapped by another control, and PMUSERENR_EL0.CR controls the behavior of EL0 writes to the cycle counter and PMZR_EL0.

CR	Meaning
`0b0`	When PMUSERENR_EL0.UEN == 0, EL0 reads of the cycle counter are disabled, unless enabled by PMUSERENR_EL0.EN. When PMUSERENR_EL0.UEN == 1, permitted EL0 writes are not affected by this mechanism.
`0b1`	When PMUSERENR_EL0.UEN == 0, EL0 reads of the cycle counter are enabled, unless trapped by another control. When PMUSERENR_EL0.UEN == 1, permitted EL0 writes to the cycle counter and PMZR_EL0.C are ignored.

In AArch64 state, the register accesses affected by this control are:

- When PMUSERENR_EL0.{UEN,EN} is {0,0}, `MRS` reads of PMCCNTR_EL0.
- When PMUSERENR_EL0.UEN is 1, `MSR` writes to PMZR_EL0 and PMCCNTR_EL0.

In AArch32 state, the register accesses affected by this control are:

- When PMUSERENR_EL0.{UEN,EN} is {0,0}:
    - `MRC` reads of PMCCNTR.
    - `MRRC` reads of PMCCNTR.
- When PMUSERENR_EL0.UEN is 1:
    - `MCR` writes to PMCCNTR.
    - `MCRR` writes to PMCCNTR.

When disabled, reads generate an exception to EL1, or to EL2 when EL2 is implemented and enabled for the current Security state and HCR_EL2.TGE is 1, and:

- AArch64 `MRS` reads are reported using EC syndrome value `0x18`.

- AArch32 `MRC` reads are reported using EC syndrome value `0x03`.
- AArch32 `MRRC` reads are reported using EC syndrome value `0x04`.

Ignored writes are not trapped and do not generate an exception.

This field is ignored by the PE when PMUSERENR_EL0.{UEN,EN} == {0,1}.

The reset behavior of this field is:

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

Otherwise:

Cycle counter Read enable.

When PMUSERENR_EL0.EN is 0, PMUSERENR_EL0.CR enables EL0 reads of the cycle counter.

CR	Meaning
`0b0`	EL0 reads of the cycle counter are disabled, unless enabled by PMUSERENR_EL0.EN.
`0b1`	EL0 reads of the cycle counter are enabled, unless trapped by another control.

In AArch64 state, the register accesses affected by this control are:

- `MRS` reads of PMCCNTR_EL0.

In AArch32 state, the register accesses affected by this control are:

- `MRC` reads of PMCCNTR.
- `MRRC` reads of PMCCNTR.

When disabled, reads generate an exception to EL1, or to EL2 when EL2 is implemented and enabled for the current Security state and HCR_EL2.TGE is 1, and:

- AArch64 `MRS` reads are reported using EC syndrome value `0x18`.
- AArch32 `MRC` reads are reported using EC syndrome value `0x03`.
- AArch32 `MRRC` reads are reported using EC syndrome value `0x04`.

This field is ignored by the PE when PMUSERENR_EL0.EN == 1.

The reset behavior of this field is:

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

SW, bit [1]
When FEAT_PMUv3p9 is implemented:

Software increment register Write enable.

When PMUSERENR_EL0.UEN is 0, PMUSERENR_EL0.SW enables EL0 writes to the Software increment register.

When PMUSERENR_EL0.UEN is 1, EL0 writes to the Software increment register are enabled by PMUSERENR_EL0.UEN, unless trapped by another control, and PMUSERENR_EL0.SW controls the behavior of EL0 writes to the Software increment register.

SW	Meaning
`0b0`	When PMUSERENR_EL0.UEN == 0, EL0 writes to the Software increment register are disabled, unless enabled by PMUSERENR_EL0.EN. When PMUSERENR_EL0.UEN == 1, permitted EL0 writes are not affected by this mechanism.
`0b1`	When PMUSERENR_EL0.UEN == 0, EL0 writes to the Software increment register are enabled, unless trapped by another control. When PMUSERENR_EL0.UEN == 1, permitted EL0 writes to the Software increment register ignore the value of PMUACR_EL1.

In AArch64 state, the register accesses affected by this control are:

- `MSR` writes to PMSWINC_EL0.

In AArch32 state, the register accesses affected by this control are:

- `MCR` writes to PMSWINC.

When disabled, writes generate an exception to EL1, or to EL2 when EL2 is implemented and enabled for the current Security state and HCR_EL2.TGE is 1, and:

- AArch64 `MSR` writes are reported using EC syndrome value `0x18`.
- AArch32 `MCR` writes are reported using EC syndrome value `0x03`.

This field is ignored by the PE when PMUSERENR_EL0.{UEN,EN} == {0,1}.

The reset behavior of this field is:

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

Otherwise:

Software increment register Write enable.

When PMUSERENR_EL0.EN is 0, PMUSERENR_EL0.SW enables EL0 writes to the Software increment register.

SW	Meaning
`0b0`	EL0 writes to the Software increment register are disabled, unless enabled by PMUSERENR_EL0.EN.
`0b1`	EL0 writes to the Software increment register are enabled, unless trapped by another control.

In AArch64 state, the register accesses affected by this control are:

- `MSR` writes to PMSWINC_EL0.

In AArch32 state, the register accesses affected by this control are:

- `MCR` writes to PMSWINC.

When disabled, writes generate an exception to EL1, or to EL2 when EL2 is implemented and enabled for the current Security state and HCR_EL2.TGE is 1, and:

- AArch64 `MSR` writes are reported using EC syndrome value `0x18`.
- AArch32 `MCR` writes are reported using EC syndrome value `0x03`.

This field is ignored by the PE when PMUSERENR_EL0.EN == 1.

The reset behavior of this field is:

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

EN, bit [0]

Enable.

Enables EL0 read/write access to PMU registers, other than the instruction counter.

EN	Meaning
`0b0`	EL0 accesses to the specified PMU System registers are trapped, unless enabled by PMUSERENR_EL0.{UEN,ER,CR,SW}.
`0b1`	EL0 accesses to the specified PMU System registers are enabled, unless trapped by another control.

In AArch64 state, the register accesses affected by this control are:

- `MRS` or `MSR` accesses to PMCCFILTR_EL0, PMCCNTR_EL0, PMCNTENCLR_EL0, PMCNTENSET_EL0, PMCR_EL0, PMEVCNTR<n>_EL0, PMEVTYPER<n>_EL0, PMOVSCLR_EL0, PMOVSSET_EL0, PMSELR_EL0, PMXEVCNTR_EL0, and PMXEVTYPER_EL0.
- `MRS` reads of PMCEID0_EL0 and PMCEID1_EL0.
- `MSR` writes to the following registers:
    - PMSWINC_EL0.
    - If FEAT_PMUv3p9 is implemented, PMZR_EL0.

Note

When FEAT_PMUv3_ICNTR is implemented, this field does not affect `MRS` and `MSR` accesses to PMICNTR_EL0 and PMICFILTR_EL0.

In AArch32 state, the register accesses affected by this control are:

- `MRC` or `MCR` accesses to PMCCFILTR, PMCCNTR, PMCNTENCLR, PMCNTENSET, PMCR, PMEVCNTR<n>, PMEVTYPER<n>, PMOVSR, PMOVSSET, PMSELR, PMXEVCNTR, and PMXEVTYPER.
- `MRC` reads of the following registers:
    - PMCEID0 and PMCEID1.
    - If FEAT_PMUv3p1 is implemented, PMCEID2 and PMCEID3.
- `MCR` writes to PMSWINC.
- `MRRC` or `MCRR` accesses to PMCCNTR.

When trapped, reads and writes generate an exception to EL1, or to EL2 when EL2 is implemented and enabled for the current Security state and HCR_EL2.TGE is 1, and:

- AArch64 `MRS` and `MSR` accesses are reported using EC syndrome value `0x18`.
- AArch32 `MRC` and `MCR` accesses are reported using EC syndrome value `0x03`.
- AArch32 `MRRC` and `MCRR` accesses are reported using EC syndrome value `0x04`.

This field is ignored by the PE when FEAT_PMUv3p9 is implemented and PMUSERENR_EL0.UEN == 1.

The reset behavior of this field is:

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

Accessing PMUSERENR_EL0

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

`MRS <Xt>, PMUSERENR_EL0`

op0	op1	CRn	CRm	op2
0b11	0b011	0b1001	0b1110	0b000

```
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 EL2Enabled() && HCR_EL2.<E2H,TGE> != '11'
&& IsFeatureImplemented(FEAT_FGT) && (!HaveEL(EL3)
|| SCR_EL3.FGTEn == '1') &&
HDFGRTR_EL2.PMUSERENR_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
        X[t, 64] = PMUSERENR_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.PMUSERENR_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
        X[t, 64] = PMUSERENR_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] = PMUSERENR_EL0;
elsif PSTATE.EL == EL3 then
    X[t, 64] = PMUSERENR_EL0;
```

MSR PMUSERENR_EL0, <Xt>

op0	op1	CRn	CRm	op2
0b11	0b011	0b1001	0b1110	0b000

```
if PSTATE.EL == EL0 then
    UNDEFINED;
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.PMUSERENR_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
            PMUSERENR_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
        PMUSERENR_EL0 = X[t, 64];
elsif PSTATE.EL == EL3 then
    PMUSERENR_EL0 = X[t, 64];
```

28/03/2023 16:02; 72747e43966d6b97dcbd230a1b3f0421d1ea3d94