# PMSNEVFR_EL1, Sampling Inverted Event Filter Register

The PMSNEVFR_EL1 characteristics are:

## Purpose

Controls sample filtering by events. The overall inverted filter is the logical OR of these filters. For example, if PMSNEVFR_EL1.E[3] and PMSNEVFR_EL1.E[5] are both set to 1, samples that have either event 3 (Level 1 unified or data cache refill) or event 5 (TLB walk) set to 1 are not recorded.

## Configuration

This register is present only when FEAT_SPEv1p2 is implemented. Otherwise, direct accesses to PMSNEVFR_EL1 are undefined.

## Attributes

PMSNEVFR_EL1 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| E[63] | E[62] | E[61] | E[60] | E[59] | E[58] | E[57] | E[56] | E[55] | E[54] | E[53] | E[52] | E[51] | E[50] | E[49] | E[48] | | |
| E[31] | E[30] | E[29] | E[28] | E[27] | E[26] | E[25] | E[24] | E[23] | E[22] | E[21] | E[20] | E[19] | E[18] | E[17] | E[16] | E[15] | E[14] |
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 |

### E[<x>], bit [x], for x = 63 to 48, 31 to 24, 15 to 12

E[<x>] is the event filter for implementation defined event <x>.

| E[<x>] | Meaning |
|---|---|
| `0b0` | Event <x> is ignored. |
| `0b1` | Do not record samples that have event <x> == 1. |

An implementation defined event might be recorded as a multi-bit field. In this case, the corresponding bits of PMSNEVFR_EL1 define an implementation defined filter for the event.

This bit is ignored by the PE when PMSFCR_EL1.FnE == 0.

The reset behavior of this field is:

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

When event <x> is not implemented, or filtering on event <x> is not supported, access to this field is **RAZ/WI**.

**Bits [47:32]**

Reserved, RAZ/WI.

**E[23], bit [23]**
**When FEAT_SPEv1p4 is implemented and event 23 is implemented:**

Data not snooped.

| E[23] | Meaning |
|---|---|
| `0b0` | Data snooped event is ignored. |
| `0b1` | Do not record samples that have the Data snooped event == 1. |

This field is ignored by the PE when PMSFCR_EL1.FnE == 0.

The reset behavior of this field is:

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

**Otherwise:**

Reserved, RAZ/WI.

**E[22], bit [22]**
**When FEAT_SPEv1p4 is implemented and event 22 is implemented:**

Not recently fetched.

| E[22] | Meaning |
|---|---|
| `0b0` | Recently fetched event is ignored. |
| `0b1` | Do not record samples that have the Recently fetched event == 1. |

This field is ignored by the PE when PMSFCR_EL1.FnE == 0.

The reset behavior of this field is:

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

**Otherwise:**

Reserved, RAZ/WI.

**E[21], bit [21]**
**When FEAT_SPEv1p4 is implemented and event 21 is implemented:**

Cache data not modified.

| E[21] | Meaning |
|---|---|
| `0b0` | Cache data modified event is ignored. |
| `0b1` | Do not record samples that have the Cache data modified event == 1. |

This field is ignored by the PE when PMSFCR_EL1.FnE == 0.

The reset behavior of this field is:

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

**Otherwise:**

Reserved, RAZ/WI.

**E[20], bit [20]**
**When FEAT_SPEv1p4 is implemented and event 20 is implemented:**

Level 2 data cache hit.

| E[20] | Meaning |
|---|---|
| `0b0` | Level 2 data cache miss event is ignored. |
| `0b1` | Do not record samples that have the Level 2 data cache miss event == 1. |

This field is ignored by the PE when PMSFCR_EL1.FnE == 0.

The reset behavior of this field is:

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

**Otherwise:**

Reserved, RAZ/WI.

**E[19], bit [19]**
**When FEAT_SPEv1p4 is implemented and event 19 is implemented:**

No level 2 data cache access.

| E[19] | Meaning |
|---|---|
| 0b0 | Level 2 data cache access event is ignored. |
| 0b1 | Do not record samples that have the Level 2 data cache access event == 1. |

This field is ignored by the PE when PMSFCR_EL1.FnE == 0.

The reset behavior of this field is:

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

**Otherwise:**

Reserved, RAZ/WI.

**E[18], bit [18]**
**When FEAT_SVE is implemented and FEAT_SPEv1p1 is implemented:**

Not empty predicate.

| E[18] | Meaning |
|---|---|
| 0b0 | Empty predicate event is ignored. |
| 0b1 | Do not record samples that have the Empty predicate event == 1. |

This field is ignored by the PE when PMSFCR_EL1.FnE == 0.

The reset behavior of this field is:

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

**Otherwise:**

Reserved, RAZ/WI.

**E[17], bit [17]**
**When FEAT_SVE is implemented and FEAT_SPEv1p1 is implemented:**

Not partial predicate.

| E[17] | Meaning |
|---|---|
| 0b0 | Partial predicate event is ignored. |
| 0b1 | Do not record samples that have the Partial predicate event == 1. |

This field is ignored by the PE when PMSFCR_EL1.FnE == 0.

The reset behavior of this field is:

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

**Otherwise:**

Reserved, RAZ/WI.

**E[16], bit [16]**
**When FEAT_TME is implemented:**

Not transactional.

| E[16] | Meaning |
|---|---|
| 0b0 | Transactional event is ignored. |
| 0b1 | Do not record samples that have the Transactional event == 1. |

This field is ignored by the PE when PMSFCR_EL1.FnE == 0.

The reset behavior of this field is:

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

**Otherwise:**

Reserved, RAZ/WI.

**E[11], bit [11]**
**When FEAT_SPEv1p1 is implemented:**

Aligned.

| E[11] | Meaning |
|---|---|
| 0b0 | Misalignment event is ignored. |
| 0b1 | Do not record samples that have the Misalignment event == 1. |

This field is ignored by the PE when PMSFCR_EL1.FnE == 0.

The reset behavior of this field is:

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

**Otherwise:**

Reserved, RAZ/WI.

**E[10], bit [10]**
**When (FEAT_SPEv1p4 is implemented or filtering on event 10 is optionally supported) and event 10 is implemented:**

No remote access.

| E[10] | Meaning |
|---|---|
| `0b0` | Remote access event is ignored. |
| `0b1` | Do not record samples that have the Remote access event == 1. |

This field is ignored by the PE when PMSFCR_EL1.FnE == 0.

The reset behavior of this field is:

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

**Otherwise:**

Reserved, RAZ/WI.

**E[9], bit [9]**
**When (FEAT_SPEv1p4 is implemented or filtering on event 9 is optionally supported) and event 9 is implemented:**

Last Level cache hit.

| E[9] | Meaning |
|---|---|
| `0b0` | Last Level cache miss event is ignored. |
| `0b1` | Do not record samples that have the Last Level cache miss event == 1. |

This field is ignored by the PE when PMSFCR_EL1.FnE == 0.

The reset behavior of this field is:

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

**Otherwise:**

Reserved, RAZ/WI.

**E[8], bit [8]**
**When (FEAT_SPEv1p4 is implemented or filtering on event 8 is optionally supported) and event 8 is implemented:**

No Last Level cache access.

| E[8] | Meaning |
|---|---|
| 0b0 | Last Level cache access event is ignored. |
| 0b1 | Do not record samples that have the Last Level cache access event == 1. |

This field is ignored by the PE when PMSFCR_EL1.FnE == 0.

The reset behavior of this field is:

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

**Otherwise:**

Reserved, RAZ/WI.

**E[7], bit [7]**

Correctly predicted.

| E[7] | Meaning |
|---|---|
| 0b0 | Mispredicted event is ignored. |
| 0b1 | Do not record samples that have the Mispredicted event == 1. |

This field is ignored by the PE when PMSFCR_EL1.FnE == 0.

The reset behavior of this field is:

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

**E[6], bit [6]**

Taken.

| E[6] | Meaning |
|---|---|

| | |
|---|---|
| 0b0 | Not taken event is ignored. |
| 0b1 | Do not record samples that have the Not taken event == 1. |

This field is ignored by the PE when PMSFCR_EL1.FnE == 0.

The reset behavior of this field is:

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

### E[5], bit [5]

TLB hit.

| E[5] | Meaning |
|---|---|
| 0b0 | TLB walk event is ignored. |
| 0b1 | Do not record samples that have the TLB walk event == 1. |

This field is ignored by the PE when PMSFCR_EL1.FnE == 0.

The reset behavior of this field is:

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

### E[4], bit [4]

**When FEAT_SPEv1p4 is implemented or filtering on event 4 is optionally supported:**

No TLB access.

| E[4] | Meaning |
|---|---|
| 0b0 | TLB access event is ignored. |
| 0b1 | Do not record samples that have the TLB access event == 1. |

This field is ignored by the PE when PMSFCR_EL1.FnE == 0.

The reset behavior of this field is:

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

**Otherwise:**

Reserved, RAZ/WI.

**E[3], bit [3]**

Level 1 data or unified cache hit.

| E[3] | Meaning |
|---|---|
| 0b0 | Level 1 data or unified cache refill event is ignored. |
| 0b1 | Do not record samples that have the Level 1 data or unified cache refill event == 1. |

This field is ignored by the PE when PMSFCR_EL1.FnE == 0.

The reset behavior of this field is:

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

**E[2], bit [2]**
**When FEAT_SPEv1p4 is implemented or filtering on event 2 is optionally supported:**

No Level 1 data cache access.

| E[2] | Meaning |
|---|---|
| 0b0 | Level 1 data cache access event is ignored. |
| 0b1 | Do not record samples that have the Level 1 data cache access event == 1. |

This field is ignored by the PE when PMSFCR_EL1.FnE == 0.

The reset behavior of this field is:

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

**Otherwise:**

Reserved, RAZ/WI.

**E[1], bit [1]**
**When the PE supports sampling of speculative instructions:**

Speculative.

| E[1] | Meaning |
|---|---|
| 0b0 | Architecturally executed event is ignored. |

| | |
|---|---|
| 0b1 | Do not record samples that have the Architecturally executed event == 1. |

This field is ignored by the PE when PMSFCR_EL1.FnE == 0.

The reset behavior of this field is:

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

**Otherwise:**

Reserved, RAZ/WI.

**Bit [0]**

Reserved, RAZ/WI.

## Accessing PMSNEVFR_EL1

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

### MRS <Xt>, PMSNEVFR_EL1

| op0 | op1 | CRn | CRm | op2 |
|---|---|---|---|---|
| 0b11 | 0b000 | 0b1001 | 0b1001 | 0b001 |

```
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.NSPB[0] == '0' ||
MDCR_EL3.NSPB[1] != SCR_EL3.NS ||
(IsFeatureImplemented(FEAT_RME) && MDCR_EL3.NSPBE !=
SCR_EL3.NSE)) then
        UNDEFINED;
    elsif Halted() && HaveEL(EL3) && EDSCR.SDD ==
'1' && boolean IMPLEMENTATION_DEFINED "EL3 trap
priority when SDD == '1'" && MDCR_EL3.EnPMSN == '0'
then
        UNDEFINED;
    elsif EL2Enabled() &&
IsFeatureImplemented(FEAT_FGT) && (!HaveEL(EL3) ||
SCR_EL3.FGTEn == '1') && HDFGRTR_EL2.nPMSNEVFR_EL1
== '0' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    elsif EL2Enabled() && MDCR_EL2.TPMS == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    elsif HaveEL(EL3) && (MDCR_EL3.NSPB[0] == '0' ||
MDCR_EL3.NSPB[1] != SCR_EL3.NS ||
(IsFeatureImplemented(FEAT_RME) && MDCR_EL3.NSPBE !=
SCR_EL3.NSE)) then
        if Halted() && EDSCR.SDD == '1' then
            UNDEFINED;
        else
            AArch64.SystemAccessTrap(EL3, 0x18);
    elsif HaveEL(EL3) && MDCR_EL3.EnPMSN == '0' then
        if Halted() && EDSCR.SDD == '1' then
            UNDEFINED;
        else
            AArch64.SystemAccessTrap(EL3, 0x18);
    elsif EL2Enabled() && HCR_EL2.<NV2,NV> == '11'
then
        X[t, 64] = NVMem[0x850];
    else
        X[t, 64] = PMSNEVFR_EL1;
elsif PSTATE.EL == EL2 then
    if Halted() && HaveEL(EL3) && EDSCR.SDD == '1'
&& boolean IMPLEMENTATION_DEFINED "EL3 trap priority
when SDD == '1'" && (MDCR_EL3.NSPB[0] == '0' ||
MDCR_EL3.NSPB[1] != SCR_EL3.NS ||
(IsFeatureImplemented(FEAT_RME) && MDCR_EL3.NSPBE !=
SCR_EL3.NSE)) then
        UNDEFINED;
    elsif Halted() && HaveEL(EL3) && EDSCR.SDD ==
'1' && boolean IMPLEMENTATION_DEFINED "EL3 trap
priority when SDD == '1'" && MDCR_EL3.EnPMSN == '0'
then
        UNDEFINED;
    elsif HaveEL(EL3) && (MDCR_EL3.NSPB[0] == '0' ||
MDCR_EL3.NSPB[1] != SCR_EL3.NS ||
(IsFeatureImplemented(FEAT_RME) && MDCR_EL3.NSPBE !=
SCR_EL3.NSE)) then
        if Halted() && EDSCR.SDD == '1' then
            UNDEFINED;
        else
            AArch64.SystemAccessTrap(EL3, 0x18);
    elsif HaveEL(EL3) && MDCR_EL3.EnPMSN == '0' then
        if Halted() && EDSCR.SDD == '1' then
            UNDEFINED;
        else
            AArch64.SystemAccessTrap(EL3, 0x18);
    else
        X[t, 64] = PMSNEVFR_EL1;
elsif PSTATE.EL == EL3 then
    X[t, 64] = PMSNEVFR_EL1;
```

## MSR PMSNEVFR_EL1, <Xt>

| op0 | op1 | CRn | CRm | op2 |
|---|---|---|---|---|

| | | | | |
|---|---|---|---|---|
| 0b11 | 0b000 | 0b1001 | 0b1001 | 0b001 |

```
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.NSPB[0] == '0' ||
MDCR_EL3.NSPB[1] != SCR_EL3.NS ||
(IsFeatureImplemented(FEAT_RME) && MDCR_EL3.NSPBE !=
SCR_EL3.NSE)) then
        UNDEFINED;
    elsif Halted() && HaveEL(EL3) && EDSCR.SDD ==
'1' && boolean IMPLEMENTATION_DEFINED "EL3 trap
priority when SDD == '1'" && MDCR_EL3.EnPMSN == '0'
then
        UNDEFINED;
    elsif EL2Enabled() &&
IsFeatureImplemented(FEAT_FGT) && (!HaveEL(EL3) ||
SCR_EL3.FGTEn == '1') && HDFGWTR_EL2.nPMSNEVFR_EL1
== '0' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    elsif EL2Enabled() && MDCR_EL2.TPMS == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    elsif HaveEL(EL3) && (MDCR_EL3.NSPB[0] == '0' ||
MDCR_EL3.NSPB[1] != SCR_EL3.NS ||
(IsFeatureImplemented(FEAT_RME) && MDCR_EL3.NSPBE !=
SCR_EL3.NSE)) then
        if Halted() && EDSCR.SDD == '1' then
            UNDEFINED;
        else
            AArch64.SystemAccessTrap(EL3, 0x18);
    elsif HaveEL(EL3) && MDCR_EL3.EnPMSN == '0' then
        if Halted() && EDSCR.SDD == '1' then
            UNDEFINED;
        else
            AArch64.SystemAccessTrap(EL3, 0x18);
    elsif EL2Enabled() && HCR_EL2.<NV2,NV> == '11'
then
        NVMem[0x850] = X[t, 64];
    else
        PMSNEVFR_EL1 = 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.NSPB[0] == '0' ||
MDCR_EL3.NSPB[1] != SCR_EL3.NS ||
(IsFeatureImplemented(FEAT_RME) && MDCR_EL3.NSPBE !=
SCR_EL3.NSE)) then
        UNDEFINED;
    elsif Halted() && HaveEL(EL3) && EDSCR.SDD ==
'1' && boolean IMPLEMENTATION_DEFINED "EL3 trap
priority when SDD == '1'" && MDCR_EL3.EnPMSN == '0'
then
        UNDEFINED;
    elsif HaveEL(EL3) && (MDCR_EL3.NSPB[0] == '0' ||
MDCR_EL3.NSPB[1] != SCR_EL3.NS ||
(IsFeatureImplemented(FEAT_RME) && MDCR_EL3.NSPBE !=
SCR_EL3.NSE)) then
```

```
        if Halted() && EDSCR.SDD == '1' then
            UNDEFINED;
        else
            AArch64.SystemAccessTrap(EL3, 0x18);
    elsif HaveEL(EL3) && MDCR_EL3.EnPMSN == '0' then
        if Halted() && EDSCR.SDD == '1' then
            UNDEFINED;
        else
            AArch64.SystemAccessTrap(EL3, 0x18);
    else
        PMSNEVFR_EL1 = X[t, 64];
elsif PSTATE.EL == EL3 then
    PMSNEVFR_EL1 = X[t, 64];
```

28/03/2023 16:02; 72747e43966d6b97dcbd230a1b3f0421d1ea3d94