# SPMSCR_EL1, System Performance Monitors Secure Control Register

The SPMSCR_EL1 characteristics are:

## Purpose

Controls observability of Secure events by System PMU <s>, and optionally controls Secure attributes for message signaled interrupts and Non-secure access to the performance monitor registers.

## Configuration

This register is present only when FEAT_SPMU is implemented. Otherwise, direct accesses to SPMSCR_EL1 are undefined.

## Attributes

SPMSCR_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 45 44 43 42 41 40 39 38 37 | 36 | 35 34 33 | 32 |
|---|---|---|---|---|
| IMPLEMENTATION DEFINED | | | | |
| IMPL | RES0 | NAO | RES0 | SO |
| 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 |

#### IMPLEMENTATION DEFINED, bits [63:32]

implementation defined observation controls. Additional implementation defined bits to control certain types of filter or events.

#### IMPL, bit [31]

Indicates SPMSCR_EL1 is present.

The reset behavior of this field is:

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

Access to this field is **RO**.

**Bits [30:5]**

Reserved, res0.

**NAO, bit [4]**
**When System PMU <s> can count or monitor non-attributable events:**

Non-attributable Observation. Controls whether events or monitorable characteristics not attributable with any source can be monitored.

| NAO | Meaning |
|---|---|
| `0b0` | Events not attributable with any event source are not counted, unless overridden by SPMSCR_EL1.SO. |
| `0b1` | Counting non-attributable events is not prevented by this field. |

When both SPMROOTCR_EL3 and SPMSCR_EL1 are implemented, non-attributable events are counted only if both SPMROOTCR_EL3.NAO is 1 and SPMSCR_EL1.{NAO, SO} is nonzero.

SPMSCR_EL1.NAO has the opposite reset polarity to SPMROOTCR_EL3.NAO.

This field is optional if Root and Realm states are not implemented. When this field is not implemented, System PMU <s> behaves as if SPMSCR_EL1.NAO is 0, and whether events or monitorable characteristics not attributable with any source can be monitored is controlled by SPMSCR_EL1.SO.

The reset behavior of this field is:

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

**Otherwise:**

Reserved, res0.

**Bits [3:1]**

Reserved, res0.

**SO, bit [0]**

Secure Observation. Controls whether events or monitorable characteristics attributable to a Secure event source can be monitored.

| SO | Meaning |
|---|---|
| 0b0 | Events attributable to a Secure event source are not counted. |
| 0b1 | Events attributable to a Secure event source are counted. |

Also controls whether events or monitorable characteristics not attributable with any source can be monitored. See SPMSCR_EL1.NAO.

The reset behavior of this field is:

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

## Accessing SPMSCR_EL1

To access SPMSCR_EL1 for System PMU <s>, set SPMSELR_EL0.SYSPMUSEL to s.

SPMSCR_EL1 is undefined if accessed in Non-secure or Realm state.

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

```
MRS <Xt>, SPMSCR_EL1
```

| op0 | op1 | CRn | CRm | op2 |
|---|---|---|---|---|
| 0b10 | 0b111 | 0b1001 | 0b1110 | 0b111 |

```
if IsCurrentSecurityState(SS_NonSecure) ||
(IsFeatureImplemented(FEAT_RME) &&
IsCurrentSecurityState(SS_Realm)) then
    UNDEFINED;
elsif 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.EnPM2 == '0' then
        UNDEFINED;
    elsif EL2Enabled() &&
IsFeatureImplemented(FEAT_FGT2) && HaveEL(EL3) &&
SCR_EL3.FGTEn2 == '0' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    elsif EL2Enabled() &&
IsFeatureImplemented(FEAT_FGT2) &&
HDFGRTR2_EL2.nSPMSCR_EL1 == '0' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    elsif HaveEL(EL3) && MDCR_EL3.EnPM2 == '0' then
        if Halted() && EDSCR.SDD == '1' then
```

```
            UNDEFINED;
        else
            AArch64.SystemAccessTrap(EL3, 0x18);
    else
        X[t, 64] =
SPMSCR_EL1[UInt(SPMSELR_EL0.SYSPMUSEL)];
elsif PSTATE.EL == EL2 then
    if Halted() && HaveEL(EL3) && EDSCR.SDD == '1'
&& boolean IMPLEMENTATION_DEFINED "EL3 trap priority
when SDD == '1'" && MDCR_EL3.EnPM2 == '0' then
        UNDEFINED;
    elsif HaveEL(EL3) && MDCR_EL3.EnPM2 == '0' then
        if Halted() && EDSCR.SDD == '1' then
            UNDEFINED;
        else
            AArch64.SystemAccessTrap(EL3, 0x18);
    else
        X[t, 64] =
SPMSCR_EL1[UInt(SPMSELR_EL0.SYSPMUSEL)];
elsif PSTATE.EL == EL3 then
    X[t, 64] =
SPMSCR_EL1[UInt(SPMSELR_EL0.SYSPMUSEL)];
```

## MSR SPMSCR_EL1, <Xt>

| op0 | op1 | CRn | CRm | op2 |
|---|---|---|---|---|
| 0b10 | 0b111 | 0b1001 | 0b1110 | 0b111 |

```
if IsCurrentSecurityState(SS_NonSecure) ||
(IsFeatureImplemented(FEAT_RME) &&
IsCurrentSecurityState(SS_Realm)) then
    UNDEFINED;
elsif 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.EnPM2 == '0' then
        UNDEFINED;
    elsif EL2Enabled() &&
IsFeatureImplemented(FEAT_FGT2) && HaveEL(EL3) &&
SCR_EL3.FGTEn2 == '0' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    elsif EL2Enabled() &&
IsFeatureImplemented(FEAT_FGT2) &&
HDFGWTR2_EL2.nSPMSCR_EL1 == '0' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    elsif HaveEL(EL3) && MDCR_EL3.EnPM2 == '0' then
        if Halted() && EDSCR.SDD == '1' then
            UNDEFINED;
        else
            AArch64.SystemAccessTrap(EL3, 0x18);
    else
```

```
        SPMSCR_EL1[UInt(SPMSELR_EL0.SYSPMUSEL)] =
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.EnPM2 == '0' then
        UNDEFINED;
    elsif HaveEL(EL3) && MDCR_EL3.EnPM2 == '0' then
        if Halted() && EDSCR.SDD == '1' then
            UNDEFINED;
        else
            AArch64.SystemAccessTrap(EL3, 0x18);
    else
        SPMSCR_EL1[UInt(SPMSELR_EL0.SYSPMUSEL)] =
X[t, 64];
elsif PSTATE.EL == EL3 then
    SPMSCR_EL1[UInt(SPMSELR_EL0.SYSPMUSEL)] = X[t,
64];
```

AArch32 Registers | AArch64 Registers | AArch32 Instructions | AArch64 Instructions | Index by Encoding | External Registers

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