SPMACCESSR_EL2, System Performance Monitors Access Register (EL2)

The SPMACCESSR_EL2 characteristics are:

Purpose

Controls access to System PMUs from EL1 and EL0.

Configuration

This register is present only when FEAT_SPMU is implemented. Otherwise, direct accesses to SPMACCESSR_EL2 are undefined.

Attributes

SPMACCESSR_EL2 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
P31		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		P3		P2		P1		P0	
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

P<m>, bits [2m+1:2m], for m = 31 to 0

System PMU <m> access. Controls access to System PMU <m>.

P<m>	Meaning
0b00	MRS read and MSR write System register accesses to System PMU <m> at EL1 and EL0 are trapped to EL2, unless the instruction generates a higher priority exception.
0b01	MSR write System register accesses to System PMU <m> at EL1 and EL0 are trapped to EL2, unless the instruction generates a higher priority exception.
0b11	This control does not cause any instructions to be trapped.

All other values are reserved.

The registers trapped by this control are:

AArch64: SPMCFGR_EL1, SPMCGCR<n>_EL1, SPMCNTENCLR_EL0, SPMCNTENSET_EL0, SPMCR_EL0, SPMDEVAFF_EL1, SPMDEVARCH_EL1, SPMEVCNTR<n>_EL0, SPMEVFILT2R<n>_EL0, SPMEVFILTR<n>_EL0, SPMEVTYPER<n>_EL0, SPMIIDR_EL1, SPMINTENCLR_EL1, SPMINTENSET_EL1, SPMOVSCLR_EL0, SPMOVSSET_EL0, and SPMSCR_EL1.

This field is ignored by the PE when EL2 is not implemented or disabled in the current Security state.

The reset behavior of this field is:

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

Accessing SPMACCESSR_EL2

When FEAT_VHE is implemented, and HCR_EL2.E2H is 1, without explicit synchronization, accesses from EL2 using the register name SPMACCESSR_EL2 or SPMACCESSR_EL1 are not guaranteed to be ordered with respect to accesses using the other register name.

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

MRS <Xt>, SPMACCESSR_EL2

op0	op1	CRn	CRm	op2
0b10	0b100	0b1001	0b1101	0b011

```
if PSTATE.EL == EL0 then
    UNDEFINED;
elsif PSTATE.EL == EL1 then
    if EL2Enabled() && HCR_EL2.NV == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    else
        UNDEFINED;
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] = SPMACCESSR_EL2;
elsif PSTATE.EL == EL3 then
    X[t, 64] = SPMACCESSR_EL2;
```

MSR SPMACCESSR_EL2, <Xt>

op0	op1	CRn	CRm	op2
0b10	0b100	0b1001	0b1101	0b011

```
if PSTATE.EL == EL0 then
    UNDEFINED;
elsif PSTATE.EL == EL1 then
    if EL2Enabled() && HCR_EL2.NV == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    else
        UNDEFINED;
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
        SPMACCESSR_EL2 = X[t, 64];
elsif PSTATE.EL == EL3 then
    SPMACCESSR_EL2 = X[t, 64];
```

MRS <Xt>, SPMACCESSR_EL1

op0	op1	CRn	CRm	op2
0b10	0b000	0b1001	0b1101	0b011

```
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.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.nSPMACCESSR_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);
    elsif EL2Enabled() && HCR_EL2.<NV2,NV1,NV> ==
'111' then
        X[t, 64] = NVMem[0x8E8];
    else
        X[t, 64] = SPMACCESSR_EL1;
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);
    elsif HCR_EL2.E2H == '1' then
        X[t, 64] = SPMACCESSR_EL2;
    else
        X[t, 64] = SPMACCESSR_EL1;
elsif PSTATE.EL == EL3 then
    X[t, 64] = SPMACCESSR_EL1;
```

MSR SPMACCESSR_EL1, <Xt>

op0	op1	CRn	CRm	op2
0b10	0b000	0b1001	0b1101	0b011

```
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.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.nSPMACCESSR_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);
    elsif EL2Enabled() && HCR_EL2.<NV2,NV1,NV> ==
'111' then
        NVMem[0x8E8] = X[t, 64];
    else
        SPMACCESSR_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.EnPM2 == '0' then
        UNDEFINED;
    elsif HaveEL(EL3) && MDCR_EL3.EnPM2 == '0' then
        if Halted() && EDSCR.SDD == '1' then
            UNDEFINED;
        else
            AArch64.SystemAccessTrap(EL3, 0x18);
    elsif HCR_EL2.E2H == '1' then
        SPMACCESSR_EL2 = X[t, 64];
    else
        SPMACCESSR_EL1 = X[t, 64];
elsif PSTATE.EL == EL3 then
    SPMACCESSR_EL1 = X[t, 64];
```

28/03/2023 16:02; 72747e43966d6b97dcbd230a1b3f0421d1ea3d94