PMEVCNTSVR<n>_EL1, Performance Monitors Event Count Saved Value Register <n>, n = 0 - 30

The PMEVCNTSVR<n>_EL1 characteristics are:

Purpose

Captures the PMU Event counter <n>, PMEVCNTR<n>_EL0.

Configuration

AArch64 System register PMEVCNTSVR<n>_EL1 bits [63:0] are architecturally mapped to External register PMU.PMEVCNTSVR<n>_EL1[63:0].

This register is present only when FEAT_PMUv3_SS is implemented. Otherwise, direct accesses to PMEVCNTSVR<n>_EL1 are undefined.

Attributes

PMEVCNTSVR<n>_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
EVCNT																															
EVCNT																															
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

EVCNT, bits [63:0]

Sampled Event Count. The value of PMEVCNTR<n>_EL0 at the last successful Capture event.

The reset behavior of this field is:

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

Accessing PMEVCNTSVR<n>_EL1

If <n> is greater-than-or-equal-to the number of implemented event counters, then direct reads of PMEVCNTSVR<n>_EL1 are undefined.

Otherwise, direct reads of PMEVCNTSVR<n>_EL1 generate a Trap exception to EL2 when all of the following are true:

- <n> is greater-than-or-equal-to the number of snapshot registers accessible at the current Exception level.
- EL2 is implemented and enabled in the current Security state.
- The access is from EL1.

Note

If EL2 is implemented and enabled in the current Security state, MDCR_EL2.HPMN identifies the number of accessible snapshot registers at EL1. Otherwise, the number of accessible snapshot registers is the number of implemented event counters. See MDCR_EL2.HPMN for more details.

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

MRS <Xt>, PMEVCNTSVR<m>_EL1 ; Where m = 0-30

op0	op1	CRn	CRm	op2
0b10	0b000	0b1110	0b10:m[4:3]	m[2:0]

```
integer m = UInt(CRm<1:0>:op2<2:0>);

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.EnPMSS == '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.nPMSSDATA == '0' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    elsif HaveEL(EL3) && MDCR_EL3.EnPMSS == '0' then
        if Halted() && EDSCR.SDD == '1' then
            UNDEFINED;
        else
            AArch64.SystemAccessTrap(EL3, 0x18);
```

```
    else
        X[t, 64] = PMEVCNTSVR_EL1[m];
elsif PSTATE.EL == EL2 then
    if Halted() && HaveEL(EL3) && EDSCR.SDD == '1'
&& boolean IMPLEMENTATION_DEFINED "EL3 trap priority
when SDD == '1'" && MDCR_EL3.EnPMSS == '0' then
        UNDEFINED;
    elsif HaveEL(EL3) && MDCR_EL3.EnPMSS == '0' then
        if Halted() && EDSCR.SDD == '1' then
            UNDEFINED;
        else
            AArch64.SystemAccessTrap(EL3, 0x18);
    else
        X[t, 64] = PMEVCNTSVR_EL1[m];
elsif PSTATE.EL == EL3 then
    X[t, 64] = PMEVCNTSVR_EL1[m];
```

28/03/2023 16:01; 72747e43966d6b97dcbd230a1b3f0421d1ea3d94