SPMDEVARCH_EL1, System Performance Monitors Device Architecture Register

The SPMDEVARCH_EL1 characteristics are:

Purpose

Provides discovery information for System PMU <s>.

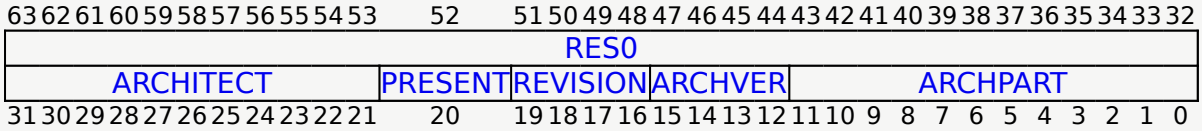
Configuration

This register is present only when FEAT_SPMU is implemented. Otherwise, direct accesses to SPMDEVARCH_EL1 are undefined.

Attributes

SPMDEVARCH_EL1 is a 64-bit register.

Field descriptions



Bits [63:32]

Reserved, res0.

ARCHITECT, bits [31:21]

Architect. Defines the architect of the component. Bits [31:28] are the JEP106 continuation code (JEP106 bank ID, minus 1) and bits [27:21] are the JEP106 ID code.

This field has an implementation defined value.

Access to this field is **RO**.

PRESENT, bit [20]

DEVARCH present. Defines that SPMDEVARCH_EL1 register is present.

PRESENT	Meaning

0b0	Device Architecture information not present.
0b1	Device Architecture information present.

If SPMDEVARCH_EL1 is not present, the register is res0.

REVISION, bits [19:16]

Revision. Defines the architecture revision of the component.

This field has an implementation defined value.

Access to this field is **RO**.

ARCHVER, bits [15:12]

Architecture Version. Defines the architecture version of the component.

SPMDEVARCH_EL1.ARCHVER and SPMDEVARCH_EL1.ARCHPART are also defined as a single field, SPMDEVARCH_EL1.ARCHID, so that SPMDEVARCH_EL1.ARCHVER is SPMDEVARCH_EL1.ARCHID[15:12].

This field has an implementation defined value.

Access to this field is **RO**.

ARCHPART, bits [11:0]

Architecture Part. Defines the architecture of the component.

SPMDEVARCH_EL1.ARCHVER and SPMDEVARCH_EL1.ARCHPART are also defined as a single field, SPMDEVARCH_EL1.ARCHID, so that SPMDEVARCH_EL1.ARCHPART is SPMDEVARCH_EL1.ARCHID[11:0].

This field has an implementation defined value.

Access to this field is **RO**.

Accessing SPMDEVARCH_EL1

To access SPMDEVARCH_EL1 for System PMU <s>, set SPMSELR_EL0.SYSPMUSEL to s.

SPMDEVARCH_EL1 reads-as-zero if the System PMU selected by SPMSELR_EL0.SYSPMUSEL is not implemented.

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

MRS <Xt>, SPMDEVARCH_EL1

op0	op1	CRn	CRm	op2
0b10	0b000	0b1001	0b1101	0b101

```
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.nSPMID == '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] =
SPMDEVARCH_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] =
SPMDEVARCH_EL1[UInt(SPMSELR_EL0.SYSPMUSEL)];
elsif PSTATE.EL == EL3 then
    X[t, 64] =
SPMDEVARCH_EL1[UInt(SPMSELR_EL0.SYSPMUSEL)];
```