PIRE0_EL2, Permission Indirection Register 0 (EL2)

The PIRE0_EL2 characteristics are:

Purpose

Stage 1 Permission Indirection Register for unprivileged access of the EL2&0 translation regime.

Configuration

This register is present only when FEAT_S1PIE is implemented. Otherwise, direct accesses to PIRE0_EL2 are undefined.

Attributes

PIRE0_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
Perm15	Perm14	Perm13	Perm12	Perm11	Perm10	Perm9	Perm8
Perm7	Perm6	Perm5	Perm4	Perm3	Perm2	Perm1	Perm0
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

Perm<m>, bits [4m+3:4m], for m = 15 to 0

Represents Stage 1 Base Permissions.

Perm<m>	Meaning
0b0000	No access, Overlay applied.
0b0001	Read, Overlay applied.
0b0010	Execute, Overlay applied.
0b0011	Read and Execute, Overlay applied.
0b0100	Reserved - treated as No access, Overlay applied.
0b0101	Read and Write, Overlay applied.
0b0110	Read, Write and Execute, Overlay applied.
0b0111	Read, Write and Execute, Overlay applied.

0b1000	Read, Overlay not applied.
0b1001	Read, GCS Read and GCS Write, Overlay not applied.
0b1010	Read and Execute, Overlay not applied.
0b1011	Reserved - treated as No access, Overlay not applied.
0b1100	Read and Write, Overlay not applied.
0b1101	Reserved - treated as No access, Overlay not applied.
0b1110	Read, Write and Execute, Overlay not applied.
0b1111	Reserved - treated as No access, Overlay not applied.

This field is permitted to be cached in a TLB.

When Stage 1 Indirect Permission mechanism is disabled, this register is ignored.

The reset behavior of this field is:

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

Accessing PIRE0_EL2

When FEAT_VHE is implemented, and HCR_EL2.E2H is 1, without explicit synchronization, accesses from EL2 using the register name PIRE0_EL2 or PIRE0_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>, PIRE0_EL2

op0	op1	CRn	CRm	op2
0b11	0b100	0b1010	0b0010	0b010

```
if PSTATE.EL == EL0 then
    UNDEFINED;
elsif PSTATE.EL == EL1 then
    if EL2Enabled() && HCR_EL2.<NV2,NV> == '11' then
        X[t, 64] = NVMem[0x298];
    elsif 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'" && SCR_EL3.PIEn == '0' then
        UNDEFINED;
    elsif HaveEL(EL3) && SCR_EL3.PIEn == '0' then
        if Halted() && EDSCR.SDD == '1' then
            UNDEFINED;
        else
            AArch64.SystemAccessTrap(EL3, 0x18);
    else
        X[t, 64] = PIRE0_EL2;
elsif PSTATE.EL == EL3 then
    X[t, 64] = PIRE0_EL2;
```

MSR PIRE0_EL2, <Xt>

op0	op1	CRn	CRm	op2
0b11	0b100	0b1010	0b0010	0b010

```
if PSTATE.EL == EL0 then
    UNDEFINED;
elsif PSTATE.EL == EL1 then
    if EL2Enabled() && HCR_EL2.<NV2,NV> == '11' then
        NVMem[0x298] = X[t, 64];
    elsif 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'" && SCR_EL3.PIEn == '0' then
        UNDEFINED;
    elsif HaveEL(EL3) && SCR_EL3.PIEn == '0' then
        if Halted() && EDSCR.SDD == '1' then
            UNDEFINED;
        else
            AArch64.SystemAccessTrap(EL3, 0x18);
    else
        PIRE0_EL2 = X[t, 64];
elsif PSTATE.EL == EL3 then
    PIRE0_EL2 = X[t, 64];
```

MRS <Xt>, PIRE0_EL1

op0	op1	CRn	CRm	op2
0b11	0b000	0b1010	0b0010	0b010

```
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'" && SCR_EL3.PIEn == '0' then
        UNDEFINED;
    elsif EL2Enabled() && HCR_EL2.TRVM == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    elsif EL2Enabled() &&
IsFeatureImplemented(FEAT_FGT) && (!HaveEL(EL3) ||
SCR_EL3.FGTEn == '1') && HFGRTR_EL2.nPIRE0_EL1 ==
'0' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    elsif HaveEL(EL3) && SCR_EL3.PIEn == '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[0x290];
    else
        X[t, 64] = PIRE0_EL1;
elsif PSTATE.EL == EL2 then
    if Halted() && HaveEL(EL3) && EDSCR.SDD == '1'
&& boolean IMPLEMENTATION_DEFINED "EL3 trap priority
when SDD == '1'" && SCR_EL3.PIEn == '0' then
        UNDEFINED;
    elsif HaveEL(EL3) && SCR_EL3.PIEn == '0' then
        if Halted() && EDSCR.SDD == '1' then
            UNDEFINED;
        else
            AArch64.SystemAccessTrap(EL3, 0x18);
    elsif HCR_EL2.E2H == '1' then
        X[t, 64] = PIRE0_EL2;
    else
        X[t, 64] = PIRE0_EL1;
elsif PSTATE.EL == EL3 then
    X[t, 64] = PIRE0_EL1;
```

MSR PIRE0_EL1, <Xt>

op0	op1	CRn	CRm	op2
0b11	0b000	0b1010	0b0010	0b010

```
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'" && SCR_EL3.PIEn == '0' then
        UNDEFINED;
    elsif EL2Enabled() && HCR_EL2.TVM == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    elsif EL2Enabled() &&
IsFeatureImplemented(FEAT_FGT) && (!HaveEL(EL3) ||
SCR_EL3.FGTEn == '1') && HFGWTR_EL2.nPIRE0_EL1 ==
'0' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    elsif HaveEL(EL3) && SCR_EL3.PIEn == '0' then
        if Halted() && EDSCR.SDD == '1' then
            UNDEFINED;
        else
            AArch64.SystemAccessTrap(EL3, 0x18);
    elsif EL2Enabled() && HCR_EL2.<NV2,NV1,NV> ==
'111' then
        NVMem[0x290] = X[t, 64];
    else
        PIRE0_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'" && SCR_EL3.PIEn == '0' then
        UNDEFINED;
    elsif HaveEL(EL3) && SCR_EL3.PIEn == '0' then
        if Halted() && EDSCR.SDD == '1' then
            UNDEFINED;
        else
            AArch64.SystemAccessTrap(EL3, 0x18);
    elsif HCR_EL2.E2H == '1' then
        PIRE0_EL2 = X[t, 64];
    else
        PIRE0_EL1 = X[t, 64];
elsif PSTATE.EL == EL3 then
    PIRE0_EL1 = X[t, 64];
```


28/03/2023 16:02; 72747e43966d6b97dcbd230a1b3f0421d1ea3d94