

PIR_EL1, Permission Indirection Register 1 (EL1)

The PIR_EL1 characteristics are:

Purpose

Stage 1 Permission Indirection Register for privileged access of the EL1&0 translation regime.

Configuration

This register is present only when FEAT_S1PIE is implemented. Otherwise, direct accesses to PIR_EL1 are undefined.

Attributes

PIR_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
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 PIR_EL1

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

MRS <Xt>, PIR_EL1

op0	op1	CRn	CRm	op2
0b11	0b000	0b1010	0b0010	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'" && 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') && HFGTR_EL2.nPIR_EL1 == '0'
    then

```

```

        AArch64.SystemAccessTrap(EL2, 0x18);
    elsif HaveEL(EL3) && SCR_EL3.PIEEn == '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[0x2A0];
        else
            X[t, 64] = PIR_EL1;
    elsif PSTATE.EL == EL2 then
        if Halted() && HaveEL(EL3) && EDSCR.SDD == '1'
&& boolean IMPLEMENTATION_DEFINED "EL3 trap priority
when SDD == '1'" && SCR_EL3.PIEEn == '0' then
            UNDEFINED;
        elsif HaveEL(EL3) && SCR_EL3.PIEEn == '0' then
            if Halted() && EDSCR.SDD == '1' then
                UNDEFINED;
            else
                AArch64.SystemAccessTrap(EL3, 0x18);
        elsif HCR_EL2.E2H == '1' then
            X[t, 64] = PIR_EL2;
        else
            X[t, 64] = PIR_EL1;
    elsif PSTATE.EL == EL3 then
        X[t, 64] = PIR_EL1;

```

MSR_PIR_EL1, <Xt>

op0	op1	CRn	CRm	op2
0b11	0b000	0b1010	0b0010	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'" && SCR_EL3.PIEEn == '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.nPIR_EL1 == '0'
then
        AArch64.SystemAccessTrap(EL2, 0x18);
    elsif HaveEL(EL3) && SCR_EL3.PIEEn == '0' then
        if Halted() && EDSCR.SDD == '1' then
            UNDEFINED;
        else
            AArch64.SystemAccessTrap(EL3, 0x18);
    elsif EL2Enabled() && HCR_EL2.<NV2,NV1,NV> ==

```

```

'111' then
    NVMem[0x2A0] = X[t, 64];
else
    PIR_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
        PIR_EL2 = X[t, 64];
    else
        PIR_EL1 = X[t, 64];
elsif PSTATE.EL == EL3 then
    PIR_EL1 = X[t, 64];

```

MRS <Xt>, PIR_EL12

op0	op1	CRn	CRm	op2
0b11	0b101	0b1010	0b0010	0b011

```

if PSTATE.EL == EL0 then
    UNDEFINED;
elsif PSTATE.EL == EL1 then
    if EL2Enabled() && HCR_EL2.<NV2,NV1,NV> == '101'
    then
        X[t, 64] = NVMem[0x2A0];
    elsif EL2Enabled() && HCR_EL2.NV == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    else
        UNDEFINED;
elsif PSTATE.EL == EL2 then
    if HCR_EL2.E2H == '1' 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] = PIR_EL1;
    else
        UNDEFINED;
elsif PSTATE.EL == EL3 then
    if EL2Enabled() && !ELUsingAArch32(EL2) &&

```

```

HCR_EL2.E2H == '1' then
    X[t, 64] = PIR_EL1;
else
    UNDEFINED;

```

MSR_PIR_EL12, <Xt>

op0	op1	CRn	CRm	op2
0b11	0b101	0b1010	0b0010	0b011

```

if PSTATE.EL == EL0 then
    UNDEFINED;
elsif PSTATE.EL == EL1 then
    if EL2Enabled() && HCR_EL2.<NV2,NV1,NV> == '101'
    then
        NVMem[0x2A0] = X[t, 64];
    elsif EL2Enabled() && HCR_EL2.NV == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    else
        UNDEFINED;
    elsif PSTATE.EL == EL2 then
        if HCR_EL2.E2H == '1' 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
                    PIR_EL1 = X[t, 64];
            else
                UNDEFINED;
        elsif PSTATE.EL == EL3 then
            if EL2Enabled() && !ELUsingAArch32(EL2) &&
            HCR_EL2.E2H == '1' then
                PIR_EL1 = X[t, 64];
            else
                UNDEFINED;

```

[AArch32
Registers](#)

[AArch64
Registers](#)

[AArch32
Instructions](#)

[AArch64
Instructions](#)

[Index by
Encoding](#)

[External
Registers](#)

