

## ICV\_PMR\_EL1, Interrupt Controller Virtual Interrupt Priority Mask Register

The ICV\_PMR\_EL1 characteristics are:

### Purpose

Provides a virtual interrupt priority filter. Only virtual interrupts with a higher priority than the value in this register are signaled to the PE.

### Configuration

AArch64 System register ICV\_PMR\_EL1 bits [31:0] are architecturally mapped to AArch32 System register [ICV\\_PMR\[31:0\]](#).

This register is present only when FEAT\_GICv3 is implemented and EL2 is implemented. Otherwise, direct accesses to ICV\_PMR\_EL1 are undefined.

To allow software to ensure appropriate observability of actions initiated by GIC register accesses, the PE and CPU interface logic must ensure that writes to this register are self-synchronising. This ensures that no interrupts below the written PMR value will be taken after a write to this register is architecturally executed. For more information, see 'Observability of the effects of accesses to the GIC registers' in ARM® Generic Interrupt Controller Architecture Specification, GIC architecture version 3.0 and version 4.0 (ARM IHI 0069).

### Attributes

ICV\_PMR\_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
RES0																															
RES0																								Priority							
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

#### Bits [63:8]

Reserved, res0.

## Priority, bits [7:0]

The priority mask level for the virtual CPU interface. If the priority of a virtual interrupt is higher than the value indicated by this field, the interface signals the virtual interrupt to the PE.

The possible priority field values are as follows:

Implemented priority bits	Possible priority field values	Number of priority levels
[7:0]	0x00-0xFF (0-255), all values	256
[7:1]	0x00-0xFE (0-254), even values only	128
[7:2]	0x00-0xFC (0-252), in steps of 4	64
[7:3]	0x00-0xF8 (0-248), in steps of 8	32
[7:4]	0x00-0xF0 (0-240), in steps of 16	16

Unimplemented priority bits are RAZ/WI.

The reset behavior of this field is:

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

## Accessing ICV\_PMR\_EL1

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

MRS <Xt>, ICC\_PMR\_EL1

op0	op1	CRn	CRm	op2
0b11	0b000	0b0100	0b0110	0b000

```

if PSTATE.EL == EL0 then
    UNDEFINED;
elseif PSTATE.EL == EL1 then
    if Halted() && HaveEL(EL3) && EDSCR.SDD == '1'
    && boolean IMPLEMENTATION_DEFINED "EL3 trap priority
when SDD == '1'" && SCR_EL3.<IRQ,FIQ> == '11' then
        UNDEFINED;
    elseif ICC_SRE_EL1.SRE == '0' then
        AArch64.SystemAccessTrap(EL1, 0x18);
    elseif EL2Enabled() && ICH_HCR_EL2.TC == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    elseif EL2Enabled() && HCR_EL2.FMO == '1' then
        X[t, 64] = ICV_PMR_EL1;
    elseif EL2Enabled() && HCR_EL2.IMO == '1' then
        X[t, 64] = ICV_PMR_EL1;
    elseif HaveEL(EL3) && SCR_EL3.<IRQ,FIQ> == '11'
then
        if Halted() && EDSCR.SDD == '1' then
            UNDEFINED;
        else
            AArch64.SystemAccessTrap(EL3, 0x18);
        else
            X[t, 64] = ICC_PMR_EL1;
    elseif PSTATE.EL == EL2 then
        if Halted() && HaveEL(EL3) && EDSCR.SDD == '1'
    && boolean IMPLEMENTATION_DEFINED "EL3 trap priority
when SDD == '1'" && SCR_EL3.<IRQ,FIQ> == '11' then
            UNDEFINED;
        elseif ICC_SRE_EL2.SRE == '0' then
            AArch64.SystemAccessTrap(EL2, 0x18);
        elseif HaveEL(EL3) && SCR_EL3.<IRQ,FIQ> == '11'
then
            if Halted() && EDSCR.SDD == '1' then
                UNDEFINED;
            else
                AArch64.SystemAccessTrap(EL3, 0x18);
            else
                X[t, 64] = ICC_PMR_EL1;
    elseif PSTATE.EL == EL3 then
        if ICC_SRE_EL3.SRE == '0' then
            AArch64.SystemAccessTrap(EL3, 0x18);
        else
            X[t, 64] = ICC_PMR_EL1;

```

## MSR ICC\_PMR\_EL1, <Xt>

op0	op1	CRn	CRm	op2
0b11	0b000	0b0100	0b0110	0b000

```

if PSTATE.EL == EL0 then
    UNDEFINED;
elseif PSTATE.EL == EL1 then

```

```

        if Halted() && HaveEL(EL3) && EDSCR.SDD == '1'
        && boolean IMPLEMENTATION_DEFINED "EL3 trap priority
when SDD == '1'" && SCR_EL3.<IRQ,FIQ> == '11' then
            UNDEFINED;
        elsif ICC_SRE_EL1.SRE == '0' then
            AArch64.SystemAccessTrap(EL1, 0x18);
        elsif EL2Enabled() && ICH_HCR_EL2.TC == '1' then
            AArch64.SystemAccessTrap(EL2, 0x18);
        elsif EL2Enabled() && HCR_EL2.FMO == '1' then
            ICV_PMR_EL1 = X[t, 64];
        elsif EL2Enabled() && HCR_EL2.IMO == '1' then
            ICV_PMR_EL1 = X[t, 64];
        elsif HaveEL(EL3) && SCR_EL3.<IRQ,FIQ> == '11'
then
            if Halted() && EDSCR.SDD == '1' then
                UNDEFINED;
            else
                AArch64.SystemAccessTrap(EL3, 0x18);
            else
                ICC_PMR_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.<IRQ,FIQ> == '11' then
                UNDEFINED;
            elsif ICC_SRE_EL2.SRE == '0' then
                AArch64.SystemAccessTrap(EL2, 0x18);
            elsif HaveEL(EL3) && SCR_EL3.<IRQ,FIQ> == '11'
then
                if Halted() && EDSCR.SDD == '1' then
                    UNDEFINED;
                else
                    AArch64.SystemAccessTrap(EL3, 0x18);
                else
                    ICC_PMR_EL1 = X[t, 64];
        elsif PSTATE.EL == EL3 then
            if ICC_SRE_EL3.SRE == '0' then
                AArch64.SystemAccessTrap(EL3, 0x18);
            else
                ICC_PMR_EL1 = X[t, 64];

```

[AArch32  
Registers](#)

[AArch64  
Registers](#)

[AArch32  
Instructions](#)

[AArch64  
Instructions](#)

[Index by  
Encoding](#)

[External  
Registers](#)

28/03/2023 16:02; 72747e43966d6b97dcbdd230a1b3f0421d1ea3d94

Copyright © 2010-2023 Arm Limited or its affiliates. All rights reserved. This document is Non-Confidential.