ICC_RPR_EL1, Interrupt Controller Running Priority Register

The ICC_RPR_EL1 characteristics are:

Purpose

Indicates the Running priority of the CPU interface.

Configuration

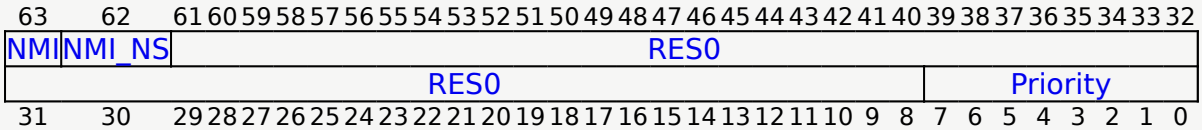
AArch64 System register ICC_RPR_EL1 performs the same function as AArch32 System register ICC_RPR.

This register is present only when FEAT_GICv3 is implemented. Otherwise, direct accesses to ICC_RPR_EL1 are undefined.

Attributes

ICC_RPR_EL1 is a 64-bit register.

Field descriptions



NMI, bit [63]
When FEAT_GICv3_NMI is implemented:

Indicates whether the running priority is from a NMI.

NMI	Meaning

0b0	When GICD_CTLR.DS==1, there are no Active NMIs, or all Active NMIs have undergone priority drop. When GICD_CTLR.DS==0: • For Non-secure and Realm reads, there are no Active Non-secure Group 1 NMIs, or all Active Non-secure Group 1 NMIs have undergone priority drop. • For Secure and Root reads, there are no Active Secure Group 1 NMIs, or all Active Secure Group 1 NMIs have undergone priority drop.
0b1	When GICD_CTLR.DS==1, there is an Active NMI. When GICD_CTLR.DS==0: • For Non-secure and Realm reads, there is an Active Non-secure Group 1 NMI. • For Secure and Root reads, there is an Active Secure Group 1 NMI.

Otherwise:

Reserved, res0.

NMI_NS, bit [62]
When FEAT_GICv3_NMI is implemented and EL3 is implemented:

Indicates whether the running priority is from a Non-secure NMI.

NMI_NS	Meaning
0b0	There are no Active Non-secure Group 1 NMIs, or all Active Non-secure Group 1 NMIs have undergone priority drop.
0b1	There is an Active Non-secure Group 1 NMI which has not undergone priority drop.

Otherwise:

Reserved, res0.

Bits [61:8]

Reserved, res0.

Priority, bits [7:0]

The current running priority on the CPU interface. This is the group priority of the current active interrupt.

The group priority of a Secure NMI, or NMI when GICD_CTLR.DS is 1, is `0x00`. The group priority of a Non-secure NMI is `0x80`, saturated to `0x00` for Non-secure reads.

If there are no active interrupts on the CPU interface, or all active interrupts have undergone a priority drop, the value returned is the Idle priority.

The priority returned is the group priority as if the BPR for the current Exception level and Security state was set to the minimum value of BPR for the number of implemented priority bits.

Note

If 8 bits of priority are implemented the group priority is bits[7:1] of the priority.

Accessing ICC_RPR_EL1

Software cannot determine the number of implemented priority bits from a read of this register.

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

MRS <Xt>, ICC_RPR_EL1

op0	op1	CRn	CRm	op2
0b11	0b000	0b1100	0b1011	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.<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
        X[t, 64] = ICV_RPR_EL1;
    elsif EL2Enabled() && HCR_EL2.IMO == '1' then
        X[t, 64] = ICV_RPR_EL1;
    elsif 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_RPR_EL1;
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
        X[t, 64] = ICC_RPR_EL1;
elsif PSTATE.EL == EL3 then
    if ICC_SRE_EL3.SRE == '0' then
        AArch64.SystemAccessTrap(EL3, 0x18);
    else
        X[t, 64] = ICC_RPR_EL1;
```


28/03/2023 16:01; 72747e43966d6b97dcbd230a1b3f0421d1ea3d94