

ICC_AP0R<n>_EL1, Interrupt Controller Active Priorities Group 0 Registers, n = 0 - 3

The ICC_AP0R<n>_EL1 characteristics are:

Purpose

Provides information about Group 0 active priorities.

Configuration

AArch64 System register ICC_AP0R<n>_EL1 bits [31:0] are architecturally mapped to AArch32 System register [ICC_AP0R<n>\[31:0\]](#).

This register is present only when FEAT_GICv3 is implemented. Otherwise, direct accesses to ICC_AP0R<n>_EL1 are undefined.

Attributes

ICC_AP0R<n>_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																															
IMPLEMENTATION DEFINED																															
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 [63:32]

Reserved, res0.

IMPLEMENTATION DEFINED, bits [31:0]

implementation defined.

The reset behavior of this field is:

- On a Warm reset, this field resets to 0.

The contents of these registers are implementation defined with the one architectural requirement that the value 0x00000000 is consistent with no interrupts being active.

Accessing ICC_AP0R<n>_EL1

Writing to these registers with any value other than the last read value of the register (or 0x00000000 when there are no Group 0 active priorities) might result in unpredictable behavior of the interrupt prioritization system, causing:

- Interrupts that should preempt execution to not preempt execution.
- Interrupts that should not preempt execution to preempt execution.

ICC_AP0R1_EL1 is only implemented in implementations that support 6 or more bits of priority. ICC_AP0R2_EL1 and ICC_AP0R3_EL1 are only implemented in implementations that support 7 or more bits of priority. Unimplemented registers are undefined.

Note

The number of bits of preemption is indicated by [ICH_VTR_EL2](#).PREbits.

Writing to the active priority registers in any order other than the following order will result in unpredictable behavior:

- ICC_AP0R<n>_EL1.
- Secure [ICC_AP1R<n>_EL1](#).
- Non-secure [ICC_AP1R<n>_EL1](#).

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

MRS <Xt>, ICC_AP0R<m>_EL1 ; Where m = 0-3

op0	op1	CRn	CRm	op2
0b11	0b000	0b1100	0b1000	0b1:m[1:0]

```
integer m = UInt(op2<1:0>);

if m == 1 && NUM_GIC_PRIORITY_BITS < 6 then
    UNDEFINED;
elseif (m == 2 || m == 3) && NUM_GIC_PRIORITY_BITS <
7 then
    UNDEFINED;
elseif 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.FIQ == '1' then
    UNDEFINED;
elseif ICC_SRE_EL1.SRE == '0' then
    AArch64.SystemAccessTrap(EL1, 0x18);
elseif EL2Enabled() && ICH_HCR_EL2.TALL0 == '1'
then
    AArch64.SystemAccessTrap(EL2, 0x18);
elseif EL2Enabled() && HCR_EL2.FMO == '1' then
    X[t, 64] = ICV_AP0R_EL1[m];
elseif HaveEL(EL3) && SCR_EL3.FIQ == '1' then
    if Halted() && EDSCR.SDD == '1' then
        UNDEFINED;
    else
        AArch64.SystemAccessTrap(EL3, 0x18);
    else
        X[t, 64] = ICC_AP0R_EL1[m];
elseif PSTATE.EL == EL2 then
    if Halted() && HaveEL(EL3) && EDSCR.SDD == '1'
&& boolean IMPLEMENTATION_DEFINED "EL3 trap priority
when SDD == '1' && SCR_EL3.FIQ == '1' then
    UNDEFINED;
elseif ICC_SRE_EL2.SRE == '0' then
    AArch64.SystemAccessTrap(EL2, 0x18);
elseif HaveEL(EL3) && SCR_EL3.FIQ == '1' then
    if Halted() && EDSCR.SDD == '1' then
        UNDEFINED;
    else
        AArch64.SystemAccessTrap(EL3, 0x18);
    else
        X[t, 64] = ICC_AP0R_EL1[m];
elseif PSTATE.EL == EL3 then
    if ICC_SRE_EL3.SRE == '0' then
        AArch64.SystemAccessTrap(EL3, 0x18);
    else
        X[t, 64] = ICC_AP0R_EL1[m];

```

MSR ICC_AP0R<m>_EL1, <Xt> ; Where m = 0-3

op0	op1	CRn	CRm	op2
0b11	0b000	0b1100	0b1000	0b1:m[1:0]

```

integer m = UInt(op2<1:0>);

if m == 1 && NUM_GIC_PRIORITY_BITS < 6 then
    UNDEFINED;
elseif (m == 2 || m == 3) && NUM_GIC_PRIORITY_BITS <
7 then
    UNDEFINED;
elseif 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.FIQ == '1' then
            UNDEFINED;
        elsif ICC_SRE_EL1.SRE == '0' then
            AArch64.SystemAccessTrap(EL1, 0x18);
        elsif EL2Enabled() && ICH_HCR_EL2.TALL0 == '1'
then
            AArch64.SystemAccessTrap(EL2, 0x18);
        elsif EL2Enabled() && HCR_EL2.FMO == '1' then
            ICV_AP0R_EL1[m] = X[t, 64];
        elsif HaveEL(EL3) && SCR_EL3.FIQ == '1' then
            if Halted() && EDSCR.SDD == '1' then
                UNDEFINED;
            else
                AArch64.SystemAccessTrap(EL3, 0x18);
            else
                ICC_AP0R_EL1[m] = 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.FIQ == '1' then
                UNDEFINED;
            elsif ICC_SRE_EL2.SRE == '0' then
                AArch64.SystemAccessTrap(EL2, 0x18);
            elsif HaveEL(EL3) && SCR_EL3.FIQ == '1' then
                if Halted() && EDSCR.SDD == '1' then
                    UNDEFINED;
                else
                    AArch64.SystemAccessTrap(EL3, 0x18);
                else
                    ICC_AP0R_EL1[m] = X[t, 64];
        elsif PSTATE.EL == EL3 then
            if ICC_SRE_EL3.SRE == '0' then
                AArch64.SystemAccessTrap(EL3, 0x18);
            else
                ICC_AP0R_EL1[m] = X[t, 64];

```

[AArch32
Registers](#)

[AArch64
Registers](#)

[AArch32
Instructions](#)

[AArch64
Instructions](#)

[Index by
Encoding](#)

[External
Registers](#)

28/03/2023 16:01; 72747e43966d6b97dcbd230a1b3f0421d1ea3d94

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