

ICC_IGRPEN1_EL1, Interrupt Controller Interrupt Group 1 Enable register

The ICC_IGRPEN1_EL1 characteristics are:

Purpose

Controls whether Group 1 interrupts are enabled for the current Security state.

Configuration

This register is banked between ICC_IGRPEN1_EL1 and ICC_IGRPEN1_EL1_S and ICC_IGRPEN1_EL1_NS.

AArch64 System register ICC_IGRPEN1_EL1 bits [31:0] (ICC_IGRPEN1_EL1_S) are architecturally mapped to AArch32 System register [ICC_IGRPEN1\[31:0\]](#) (ICC_IGRPEN1_S).

AArch64 System register ICC_IGRPEN1_EL1 bits [31:0] (ICC_IGRPEN1_EL1_NS) are architecturally mapped to AArch32 System register [ICC_IGRPEN1\[31:0\]](#) (ICC_IGRPEN1_NS).

This register is present only when FEAT_GICv3 is implemented. Otherwise, direct accesses to ICC_IGRPEN1_EL1 are undefined.

Attributes

ICC_IGRPEN1_EL1 is a 64-bit register.

This register has the following instances:

- ICC_IGRPEN1_EL1, when EL3 is not implemented
- ICC_IGRPEN1_EL1_S, when EL3 is implemented
- ICC_IGRPEN1_EL1_NS, when EL3 is implemented

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																															Enable
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 [63:1]

Reserved, res0.

Enable, bit [0]

Enables Group 1 interrupts for the current Security state.

Enable	Meaning
0b0	Group 1 interrupts are disabled for the current Security state.
0b1	Group 1 interrupts are enabled for the current Security state.

Virtual accesses to this register update [ICH_VMCR_EL2.VENG1](#).

If EL3 is present:

- The Secure [ICC_IGRPEN1_EL1.Enable](#) bit is a read/write alias of the [ICC_IGRPEN1_EL3.EnableGrp1S](#) bit.
- The Non-secure [ICC_IGRPEN1_EL1.Enable](#) bit is a read/write alias of the [ICC_IGRPEN1_EL3.EnableGrp1NS](#) bit.

If the highest priority pending interrupt for that PE is a Group 1 interrupt using 1 of N model, then the interrupt will target another PE as a result of the Enable bit changing from 1 to 0.

The reset behavior of this field is:

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

Accessing ICC_IGRPEN1_EL1

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

MRS <Xt>, ICC_IGRPEN1_EL1

op0	op1	CRn	CRm	op2
0b11	0b000	0b1100	0b1100	0b111

```
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 == '1' then
        UNDEFINED;
    elsif ICC_SRE_EL1.SRE == '0' then
        AArch64.SystemAccessTrap(EL1, 0x18);
    elsif EL2Enabled() &&
    IsFeatureImplemented(FEAT_FGT) && (!HaveEL(EL3) ||
    SCR_EL3.FGTEn == '1') && HFGTR_EL2.ICC_IGRPENn_EL1
```

```

== '1' then
    AArch64.SystemAccessTrap(EL2, 0x18);
elseif EL2Enabled() && ICH_HCR_EL2.TALL1 == '1'
then
    AArch64.SystemAccessTrap(EL2, 0x18);
elseif EL2Enabled() && HCR_EL2.IMO == '1' then
    X[t, 64] = ICV_IGRPEN1_EL1;
elseif HaveEL(EL3) && SCR_EL3.IRQ == '1' then
    if Halted() && EDSCR.SDD == '1' then
        UNDEFINED;
    else
        AArch64.SystemAccessTrap(EL3, 0x18);
elseif HaveEL(EL3) then
    if SCR_EL3.NS == '0' then
        X[t, 64] = ICC_IGRPEN1_EL1_S;
    else
        X[t, 64] = ICC_IGRPEN1_EL1_NS;
else
    X[t, 64] = ICC_IGRPEN1_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 == '1' then
        UNDEFINED;
    elseif ICC_SRE_EL2.SRE == '0' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    elseif HaveEL(EL3) && SCR_EL3.IRQ == '1' then
        if Halted() && EDSCR.SDD == '1' then
            UNDEFINED;
        else
            AArch64.SystemAccessTrap(EL3, 0x18);
    elseif HaveEL(EL3) then
        if SCR_EL3.NS == '0' then
            X[t, 64] = ICC_IGRPEN1_EL1_S;
        else
            X[t, 64] = ICC_IGRPEN1_EL1_NS;
    else
        X[t, 64] = ICC_IGRPEN1_EL1;
elseif PSTATE.EL == EL3 then
    if ICC_SRE_EL3.SRE == '0' then
        AArch64.SystemAccessTrap(EL3, 0x18);
    else
        if SCR_EL3.NS == '0' then
            X[t, 64] = ICC_IGRPEN1_EL1_S;
        else
            X[t, 64] = ICC_IGRPEN1_EL1_NS;

```

MSR ICC_IGRPEN1_EL1, <Xt>

op0	op1	CRn	CRm	op2
0b11	0b000	0b1100	0b1100	0b111

```

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 == '1' then
            UNDEFINED;
        elsif ICC_SRE_EL1.SRE == '0' then
            AArch64.SystemAccessTrap(EL1, 0x18);
        elsif EL2Enabled() &&
        IsFeatureImplemented(FEAT_FGT) && (!HaveEL(EL3) ||
        SCR_EL3.FGTEn == '1') && HFGWTR_EL2.ICC_IGRPENn_EL1
        == '1' then
            AArch64.SystemAccessTrap(EL2, 0x18);
        elsif EL2Enabled() && ICH_HCR_EL2.TALL1 == '1'
        then
            AArch64.SystemAccessTrap(EL2, 0x18);
        elsif EL2Enabled() && HCR_EL2.IMO == '1' then
            ICV_IGRPEN1_EL1 = X[t, 64];
        elsif HaveEL(EL3) && SCR_EL3.IRQ == '1' then
            if Halted() && EDSCR.SDD == '1' then
                UNDEFINED;
            else
                AArch64.SystemAccessTrap(EL3, 0x18);
            elsif HaveEL(EL3) then
                if SCR_EL3.NS == '0' then
                    ICC_IGRPEN1_EL1_S = X[t, 64];
                else
                    ICC_IGRPEN1_EL1_NS = X[t, 64];
            else
                ICC_IGRPEN1_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 == '1' then
                UNDEFINED;
            elsif ICC_SRE_EL2.SRE == '0' then
                AArch64.SystemAccessTrap(EL2, 0x18);
            elsif HaveEL(EL3) && SCR_EL3.IRQ == '1' then
                if Halted() && EDSCR.SDD == '1' then
                    UNDEFINED;
                else
                    AArch64.SystemAccessTrap(EL3, 0x18);
            elsif HaveEL(EL3) then
                if SCR_EL3.NS == '0' then
                    ICC_IGRPEN1_EL1_S = X[t, 64];
                else
                    ICC_IGRPEN1_EL1_NS = X[t, 64];
            else
                ICC_IGRPEN1_EL1 = X[t, 64];
        elsif PSTATE.EL == EL3 then
            if ICC_SRE_EL3.SRE == '0' then
                AArch64.SystemAccessTrap(EL3, 0x18);
            else
                if SCR_EL3.NS == '0' then
                    ICC_IGRPEN1_EL1_S = X[t, 64];
                else
                    ICC_IGRPEN1_EL1_NS = 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.