ICH_VMCR_EL2, Interrupt Controller Virtual Machine Control Register

The ICH_VMCR_EL2 characteristics are:

Purpose

Enables the hypervisor to save and restore the virtual machine view of the GIC state.

Configuration

AArch64 System register ICH_VMCR_EL2 bits [31:0] are architecturally mapped to AArch32 System register ICH_VMCR[31:0].

This register is present only when FEAT_GICv3 is implemented and (EL2 is implemented or EL3 is implemented). Otherwise, direct accesses to ICH_VMCR_EL2 are undefined.

If EL2 is not implemented, this register is res0 from EL3.

This register has no effect if EL2 is not enabled in the current Security state.

Attributes

ICH_VMCR_EL2 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

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
VPMR	VBPR0	VBPR1	RES0	VEOIM	RES0	VCBPR	VFIQEn	VAckCtl	VENG1	VENG0

Bits [63:32]

Reserved, res0.

VPMR, bits [31:24]

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

This field is an alias of ICV_PMR_EL1.Priority.

VBPR0, bits [23:21]

Virtual Binary Point Register, Group 0. Defines the point at which the priority value fields split into two parts, the group priority field and the subpriority field. The group priority field determines Group 0 interrupt preemption, and also determines Group 1 interrupt preemption if ICH_VMCR_EL2.VCBPR == 1.

This field is an alias of ICV_BPR0_EL1.BinaryPoint.

The minimum value of this field is determined by ICH_VTR_EL2.PREbits. An attempt to program the binary point field to a value less than the minimum value sets the field to the minimum value.

VBPR1, bits [20:18]

Virtual Binary Point Register, Group 1. Defines the point at which the priority value fields split into two parts, the group priority field and the subpriority field. The group priority field determines Group 1 interrupt preemption if ICH_VMCR_EL2.VCBPR == 0.

This field is an alias of ICV_BPR1_EL1.BinaryPoint.

This field is always accessible to EL2 accesses, regardless of the setting of the ICH_VMCR_EL2.VCBPR field.

For Non-secure writes, the minimum value of this field is the minimum value of ICH_VMCR_EL2.VBPR0 plus one.

For Secure writes, the minimum value of this field is the minimum value of ICH_VMCR_EL2.VBPR0.

An attempt to program the binary point field to a value less than the minimum value sets the field to the minimum value.

Bits [17:10]

Reserved, res0.

VEOIM, bit [9]

Virtual EOI mode. Controls whether a write to an End of Interrupt register also deactivates the virtual interrupt:

VEOIM	Meaning

0b0	ICV_EOIR0_EL1 and ICV_EOIR1_EL1 provide both priority drop and interrupt deactivation functionality. Accesses to ICV_DIR_EL1 are unpredictable.
0b1	ICV_EOIR0_EL1 and ICV_EOIR1_EL1 provide priority drop functionality only. ICV_DIR_EL1 provides interrupt deactivation functionality.

This bit is an alias of ICV_CTLR_EL1.EOImode.

Bits [8:5]

Reserved, res0.

VCBPR, bit [4]

Virtual Common Binary Point Register. Possible values of this bit are:

VCBPR	Meaning
0b0	ICV_BPR1_EL1 determines the preemption group for virtual Group 1 interrupts.
0b1	Reads of ICV_BPR1_EL1 return ICV_BPR0_EL1 plus one, saturated to `0b111`. Writes to ICV_BPR1_EL1 are ignored.

This field is an alias of ICV_CTLR_EL1.CBPR.

VFIQEn, bit [3]

Virtual FIQ enable. Possible values of this bit are:

VFIQEn	Meaning
0b0	Group 0 virtual interrupts are presented as virtual IRQs.
0b1	Group 0 virtual interrupts are presented as virtual FIQs.

This bit is an alias of GICV_CTLR.FIQEn.

In implementations where the Non-secure copy of ICC_SRE_EL1.SRE is always 1, this bit is res1.

VAckCtl, bit [2]

Virtual AckCtl. Possible values of this bit are:

VAckCtl	Meaning
`0b0`	If the highest priority pending interrupt is Group 1, a read of GICV_IAR or GICV_HPPIR returns an INTID of 1022.
`0b1`	If the highest priority pending interrupt is Group 1, a read of GICV_IAR or GICV_HPPIR returns the INTID of the corresponding interrupt.

This bit is an alias of GICV_CTLR.AckCtl.

This field is supported for backwards compatibility with GICv2. Arm deprecates the use of this field.

In implementations where the Non-secure copy of ICC_SRE_EL1.SRE is always 1, this bit is res0.

VENG1, bit [1]

Virtual Group 1 interrupt enable. Possible values of this bit are:

VENG1	Meaning
`0b0`	Virtual Group 1 interrupts are disabled.
`0b1`	Virtual Group 1 interrupts are enabled.

This bit is an alias of ICV_IGRPEN1_EL1.Enable.

VENG0, bit [0]

Virtual Group 0 interrupt enable. Possible values of this bit are:

VENG0	Meaning
`0b0`	Virtual Group 0 interrupts are disabled.
`0b1`	Virtual Group 0 interrupts are enabled.

This bit is an alias of ICV_IGRPEN0_EL1.Enable.

Accessing ICH_VMCR_EL2

When EL2 is using System register access, EL1 using either System register or memory-mapped access must be supported.

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

MRS <Xt>, ICH_VMCR_EL2

op0	op1	CRn	CRm	op2
0b11	0b100	0b1100	0b1011	0b111

```
if PSTATE.EL == EL0 then
    UNDEFINED;
elsif PSTATE.EL == EL1 then
    if EL2Enabled() && HCR_EL2.<NV2,NV> == '11' then
        X[t, 64] = NVMem[0x4C8];
    elsif EL2Enabled() && HCR_EL2.NV == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    else
        UNDEFINED;
elsif PSTATE.EL == EL2 then
    if ICC_SRE_EL2.SRE == '0' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    else
        X[t, 64] = ICH_VMCR_EL2;
elsif PSTATE.EL == EL3 then
    if ICC_SRE_EL3.SRE == '0' then
        AArch64.SystemAccessTrap(EL3, 0x18);
    else
        X[t, 64] = ICH_VMCR_EL2;
```

MSR ICH_VMCR_EL2, <Xt>

op0	op1	CRn	CRm	op2
0b11	0b100	0b1100	0b1011	0b111

```
if PSTATE.EL == EL0 then
    UNDEFINED;
elsif PSTATE.EL == EL1 then
    if EL2Enabled() && HCR_EL2.<NV2,NV> == '11' then
        NVMem[0x4C8] = X[t, 64];
    elsif EL2Enabled() && HCR_EL2.NV == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    else
        UNDEFINED;
elsif PSTATE.EL == EL2 then
    if ICC_SRE_EL2.SRE == '0' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    else
        ICH_VMCR_EL2 = X[t, 64];
```

```
elsif PSTATE.EL == EL3 then
    if ICC_SRE_EL3.SRE == '0' then
        AArch64.SystemAccessTrap(EL3, 0x18);
    else
        ICH_VMCR_EL2 = X[t, 64];
```

28/03/2023 16:02; 72747e43966d6b97dcbd230a1b3f0421d1ea3d94