GICD_ISENABLER<n>, Interrupt Set-Enable Registers, n = 0 - 31

The GICD_ISENABLER<n> characteristics are:

Purpose

Enables forwarding of the corresponding interrupt to the CPU interfaces.

Configuration

These registers are available in all GIC configurations. If GICD_CTLR.DS==0, these registers are Common.

The number of implemented GICD_ISENABLER<n> registers is (GICD_TYPER.ITLinesNumber+1). Registers are numbered from 0.

GICD_ISENABLER0 is Banked for each connected PE with GICR_TYPER.Processor_Number < 8.

Accessing GICD_ISENABLER0 from a PE with GICR_TYPER.Processor_Number > 7 is constrained unpredictable:

- Register is RAZ/WI.
- An unknown banked copy of the register is accessed.

Attributes

GICD_ISENABLER<n> is a 32-bit register.

Field descriptions

31	30	29	28	27	26
Set_enable_bit31	Set_enable_bit30	Set_enable_bit29	Set_enable_bit28	Set_enable_bit27	Set_enable_b

Set_enable_bit<x>, bit [x], for x = 31 to 0

For SPIs and PPIs, controls the forwarding of interrupt number 32n + x to the CPU interfaces. Reads and writes have the following behavior:

Set_enable_bit<x>	Meaning

0b0	If read, indicates that forwarding of the corresponding interrupt is disabled. If written, has no effect.
0b1	If read, indicates that forwarding of the corresponding interrupt is enabled. If written, enables forwarding of the corresponding interrupt. After a write of 1 to this bit, a subsequent read of this bit returns 1.

For SGIs, the behavior of this bit is implementation defined.

The reset behavior of this field is:

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

For INTID m, when DIV and MOD are the integer division and modulo operations:

- The corresponding GICD_ISENABLER<n> number, n, is given by n = m DIV 32.
- The offset of the required GICD_ISENABLER is (0x100 + (4*n)).
- The bit number of the required group modifier bit in this register is m MOD 32.

At start-up, and after a reset, a PE can use this register to discover which peripheral INTIDs the GIC supports. If GICD_CTLR.DS==0 in a system that supports EL3, the PE must do this for the Secure view of the available interrupts, and Non-secure software running on the PE must do this discovery after the Secure software has configured interrupts as Group 0/Secure Group 1 and Non-secure Group 1.

Accessing GICD_ISENABLER<n>

For SGIs and PPIs:

- When ARE is 1 for the Security state of an interrupt, the field for that interrupt is res0 and an implementation is permitted to make the field RAZ/WI in this case.

- Equivalent functionality is provided by GICR_ISENABLER0.

Bits corresponding to unimplemented interrupts are RAZ/WI.

When GICD_CTLR.DS==0, bits corresponding to Group 0 or Secure Group 1 interrupts are RAZ/WI to Non-secure accesses.

It is implementation defined whether implemented SGIs are permanently enabled, or can be enabled and disabled by writes to GICD_ISENABLER<n> and GICD_ICENABLER<n> where n=0.

For SPIs and PPIs, each bit controls the forwarding of the corresponding interrupt from the Distributor to the CPU interfaces.

GICD_ISENABLER<n> can be accessed through the memory-mapped interfaces:

Component	Frame	Offset	Instance
GIC Distributor	Dist_base	0x0100 + (4 * n)	GICD_ISENABLER<n>

Accesses on this interface are **RW**.

28/03/2023 16:02; 72747e43966d6b97dcbd230a1b3f0421d1ea3d94