Chapter 9 Generic Interrupt Controller (GIC)

9.1 Overview

The generic interrupt controller(GIC) in this device has two interfaces, the distributor interface connects to the interrupt source, and the CPU interface connects to Cortex-A7.

It supports the following features:

- Supports 128hardware interrupt inputs
- Masking of any interrupts
- Prioritization of interrupts
- Distribution of the interrupts to the target Cortex-A7 processor(s)
- Generation of interrupts by software
- Supports Security Extensions

9.2 Block Diagram

Fig.12-1 shows the block diagram of GIC.

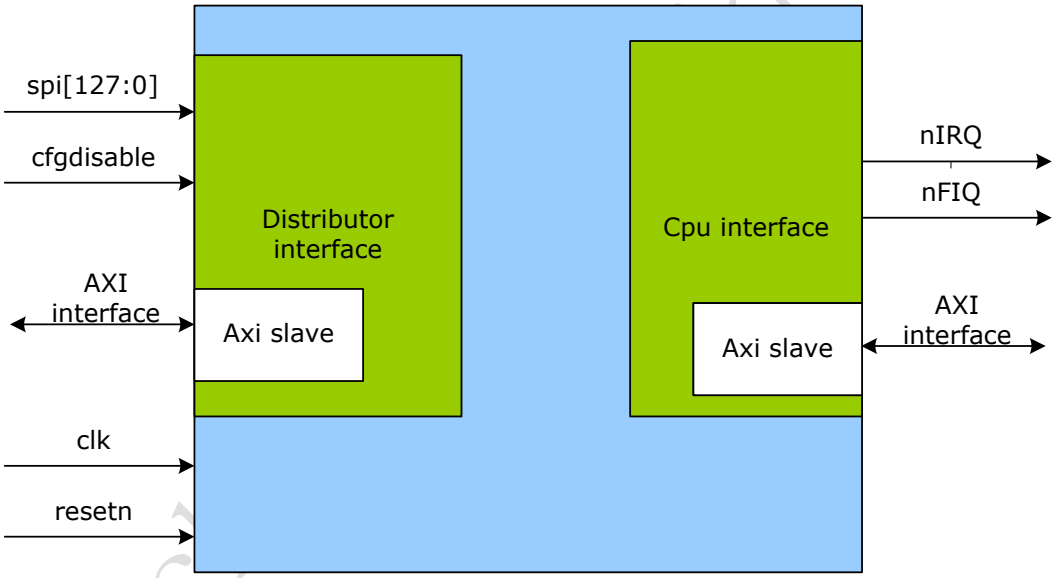

Fig. 9-1Block diagram of GIC

The diagram shows that GIC has two AXI interfaces independently and has two base address for these two interfaces.

9.3 Function Description

This GIC architecture splits logically into a Distributor block and one CPU interface block, as Figure 12-1 shows.

Distributor

This performs interrupt prioritization and distribution to the CPU interface that connect to the processor in the system.

CPU interface

CPU interface performs priority masking and preemption handling for a connected processor in the system.

9.3.1 The Distributor

The Distributor centralizes all interrupt sources, determines the priority of each interrupt, and for CPU interface dispatches the interrupt with the highest priority to the interface for priority masking and preemption handling.

The Distributor provides a programming interface for:

- Globally enabling the forwarding of interrupts to the CPU interface
- Enabling or disabling each interrupt
- Setting the priority level of each interrupt
- Setting the target processor list of each interrupt
- Setting each peripheral interrupt to be level-sensitive or edge-triggered
- If the GIC implements the Security Extensions, setting each interrupt as either
- Secure or Non-secure
- Sending a Software-generated interrupt (SGI) to processor.
- Visibility of the state of each interrupt
- A mechanism for software to set or clear the pending state of a peripheral interrupt.

Interrupt ID

Interrupts from sources are identified using ID numbers. CPU interface can see up to 160 interrupts.

The GIC assigns interrupt these 128 ID numbers as follows:

- Interrupt numbers ID32-ID127 are used for SPIs(shared peripheral interrupts).
- ID0-ID15 are used for SGI.
- ID16-ID31 are used for Private peripheral interrupt (PPI).

The GIC architecture reserves interrupt ID numbers 1022-1023 for special purposes.

ID1022

The GIC returns this value to a processor in response to an interrupt acknowledge only when the following apply:

- The interrupt acknowledge is a Secure read
- The highest priority pending interrupt is Non-secure
- The AckCtl bit in the Secure ICCICR is set to 0
- The priority of the interrupt is sufficient for it to be signalled to the processor.

Interrupt ID 1022 informs secure software that there is a Non-secure interrupt of sufficient priority to be signalled to the processor, that must be handled by Non-secure software. In this situation the secure software might alter its schedule to permit Non-secure software to handle the interrupt, to minimize the interrupt latency.

ID1023

This value is returned to a processor, in response to an interrupt acknowledge, if there is no pending interrupt with sufficient priority for it to be signalled to the processor.

On a processor that implements the Security Extensions, Secure software treats values of 1022 and 1023 as spurious interrupts.

9.3.2 CPU interface

CPU interface block provides the interface for a processor that operates with the GIC. CPU interface provides a programming interface for:

- Enabling the signal of interrupt requests by the CPU interface
- Acknowledging an interrupt
- Indicating completion of the processing of an interrupt
- Setting an interrupt priority mask for the processor
- Defining the preemption policy for the processor
- Determining the highest priority pending interrupt for the processor.

When enabled, CPU interface takes the highest priority pending interrupt for its connected processor and determines whether the interrupt has sufficient priority for it to signal the interrupt request to the processor.

To determine whether to signal the interrupt request to the processor the CPU interface considers the interrupt priority mask and the preemption settings for the processor. At any time, the connected processor can read the priority of its highest priority active interrupt from a CPU interface register.

The processor acknowledges the interrupt request by reading the CPU interface Interrupt Acknowledge register. The CPU interface returns one of:

The ID number of the highest priority pending interrupt, if that interrupt is of sufficient priority to generate an interrupt exception on the processor. This is the normal response to an interrupt acknowledge.

Exceptionally, an ID number that indicates a spurious interrupt.

When the processor acknowledges the interrupt at the CPU interface, the Distributor changes the status of the interrupt from pending to either active, or active and pending. At this point the CPU interface can signal another interrupt to the processor, to preempt interrupts that are active on the processor. If there is no pending interrupt with sufficient priority for signaling to the processor, the interface de-asserts the interrupt request signal to the processor.

When the interrupt handler on the processor has completed the processing of an interrupt, it writes to the CPU interface to indicate interrupt completion. When this happens, the distributor changes the status of the interrupt either:

- From active to inactive
- From active and pending to pending.

9.3.3 Interrupt handling state machine

The distributor maintains a state machine for each supported interrupt on CPU interface. Following figure shows an instance of this state machine, and the possible state transitions.

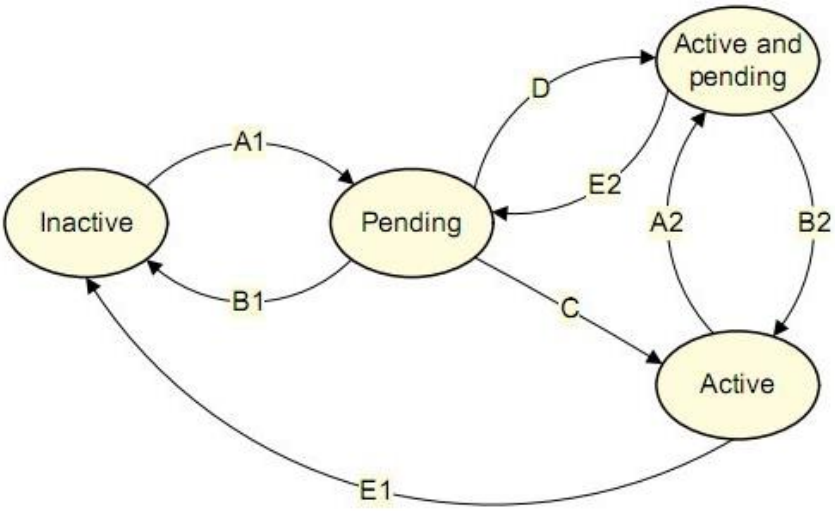

Fig. 9-2 GIC Interrupt handling state machine

Transition A1 or A2, add pending status

For an SGI:

- Occurs on a write to an ICDSGIR that specifies the processor as a target.
- If the GIC implements the Security Extensions and the write to the ICDSGIR is Secure, the transition occurs only if the security configuration of the specified SGI, for the CPU interface, corresponds to the ICDSGIR.SATT bit value.

For an SPI, occurs if either:

- a peripheral asserts an interrupt signal
- software writes to an ICDISPR.

Transition B1 or B2, remove pending status

Not applicable to SGIs:

- a pending SGI must transition through the active state, or reset, to remove its pending status.
- an active and pending SGI must transition through the pending state, or reset, to remove its pending status.

For an SPI, occurs if either:

- the level-sensitive interrupt is pending only because of the assertion of an input signal, and that signal is deasserted
- the interrupt is pending only because of the assertion of an edge-triggered interrupt signal, or a write to an ICDISPR, and software writes to the corresponding ICDICPR.

Transition C

If the interrupt is enabled and of sufficient priority to be signalled to the processor, occurs when software reads from the ICCIAR.

Transition D

For an SGI, occurs if the associated SGI is enabled and the Distributor forwards it to the CPU interface at the same time that the processor reads the ICCIAR to acknowledge a previous instance of the SGI. Whether this transition occurs

depends on the timing of the read of the ICCIAR relative to the reforwarding of the SGI.

For an SPI:

- Occurs if all the following apply:
  - The interrupt is enabled.
  - Software reads from the ICCIAR. This read adds the active state to the interrupt.
  - For a level-sensitive interrupt, the interrupt signal remains asserted. This is usually the case, because the peripheral does not deassert the interrupt until the processor has serviced the interrupt.
- For an edge-triggered interrupt, whether this transition occurs depends on the timing of the read of the ICCIAR relative to the detection of the reassertion of the interrupt. Otherwise the read of the ICCIAR causes transition C, possibly followed by transition A2.

Transition E1 or E2, remove active status

Occurs when software writes to the ICCEOIR.

9.4 Register Description

9.4.1 GIC Distributor interface register summary

Name	Offset	Size	Reset	Description
GICD_ICDDCR	0x000	W	0x0	Distributor Control Register
GICD_ICDICTR	0x004	W		Interrupt Controller Type Register
GICD_ICDIIDR	0x008	W		Distributor Implementer Identification Register
GICD_ICDISR	0x080	W		Interrupt Security Registers
-	-	-	-	reserved
GICD_ICDISER	0x100-0x17C	W		Interrupt Set-Enable Registers
GICD_ICDICER	0x180-0x1FC	W		Interrupt Clear-Enable Registers
GICD_ICDISPR	0x200-0x27C	W	0x0	Interrupt Set-Pending Registers
GICD_ICDICPR	0x280-0x2FC	W	0x0	Interrupt Clear-Pending Registers
GICD_ICDABR	0x300-0x37C	W	0x0	Active Bit Registers
-	-	-	-	reserved
GICD_ICDIPR	0x400-0x7F8	B	0x0	Interrupt Priority Registers
-	-	-	-	reserved
GICD_ICDIPTR	0x800-0x81C	B		Interrupt Processor Targets
-	-	-	-	reserved
GICD_ICDICFR	0xC00-0xCFC	W		Interrupt Configuration Registers
-	-	-	-	Reserved
GICD_ICPPISR		W		PPI Status Register
GICD_ICSPISR	0xD04-0xD1C	W		SPI Status Registers
-	-	-	-	Reserved
GICD_ICDSGIR	0xF00	W		Software Generated Interrupt Register

Notes:
*Size: **B** – Byte (8 bits) access, **HW** – Half WORD (16 bits) access, **W** –WORD (32 bits) access*

9.4.2 GIC Distributor interface detail register description

GICD_ICDDCR
Address:Operational Base+0x0
Distributor Control Register

Bit	Attr	Reset Value	Description
31:1	-	-	reserved
1	RW	0x0	1'b0: disables all Non-secure interrupts control bits in the distributor from changing state because of any external stimulus change that occurs on the corresponding SPI or PPI signals 1'b1: enables the distributor to update register locations for Non-secure interrupts
0	RW	0x0	1'b0: disables all Secure interrupt control bits in the distributor from changing state because of any external stimulus change that occurs on the corresponding SPI or PPI signals. 1'b1: enables the distributor to update register locations for Secure interrupts.

GICD_ICDICTR
Address:Operational Base+0x4
Interrupt Controller Type Register

Bit	Attr	Reset Value	Description
31:11	-	-	reserved
15:11	R	0x0	Returns the number of Lockable Shared Peripheral Interrupts (LSPIs) that the controller contains. The encoding is: 5'b11111: 31 LSPIs, that are the interrupts of IDs 32-62. When CFGSDISABLE is HIGH, the interrupt controller prevents writes to any register location that controls the operating state of an LSPI.
10	R	0x1	Indicates whether the GIC implements the Security Extensions. 1'b0: Security Extensions not implemented. 1'b1: Security Extensions implemented
9:8	-	-	reserved
7:5	R	0x0	Indicates the number of implemented CPU interface. The number of implemented CPU interface is one more than the value of this field, for example if this field is 3'b011, there are four CPU interface. In this product ,only one CPU interface is implemented.
4:0	R	0x2	Indicates the mAXImum number of interrupts that the GIC supportsa. If the value of this field is N, the mAXImum number of interrupts is 32(N+1). The interrupt ID range is from 0 to one less than the number of IDs. For example:

			5'b00011: Up to 128 interrupt lines, interrupt IDs 0-127. The mAXImum number of interrupts is 1020 (5'b11111).

GICD_ICDIIDR

Address:Operational Base+0x8

Distributor Implementer Identification Register

Bit	Attr	Reset Value	Description
31:24	R	0x0	product identifier.
23:20	-	-	reserved
19:16	R	0x0	variant number. Typically, this field is used to distinguish product variants, or major revisions of a product
15:12	R	0x0	revision number. Typically, this field is used to distinguish minor revisions of a product
11:0	R	0x0	Contains the JEP106 code of the company that implemented the GIC Distributor:a Bits [11:8]: The JEP106 continuation code of the implementer. Bits [7]: Always 0. Bits [6:0]: The JEP106 identity code of the implementer.

GICD_ICDISR

Address:Operational Base+0x80

Interrupt Security Registers

Bit	Attr	Reset Value	Description
31:0	RW	0x0	For each bit: 1'b0: The corresponding interrupt is Secure. 1'b1: The corresponding interrupt is Non-secure.

For interrupt ID N, when DIV and MOD are the integer division and modulo operations:

- the corresponding ICDISR number, M, is given by M = N DIV 32
- the offset of the required ICDISR is (0x080 + (4*M))
- the bit number of the required Security status bit in this register is N MOD 32.

GICD_ICDISER

Address:Operational Base+0x100

Interrupt Set-Enable Registers

Bit	Attr	Reset Value	Description
31:0	RW		For SPIs, for each bit: Reads 1'b0: The corresponding interrupt is disabled. 1'b1: The corresponding interrupt is enabled. Writes 1'b0: Has no effect. 1'b1: Enables the corresponding interrupt. A subsequent read of this bit returns the value 1.

For interrupt ID N, when DIV and MOD are the integer division and modulo operations:

- the corresponding ICDISER number, M, is given by M = N DIV 32
- the offset of the required ICDISER is (0x100 + (4*M))

• the bit number of the required Set-enable bit in this register is N MOD 32.

GICD_ICDICER
Address:Operational Base+0x180
Interrupt Clear-Enable Registers

Bit	Attr	Reset Value	Description
31:0	RW	0x0	For SPI, for each bit: Reads 1'b0: The corresponding interrupt is disabled. 1'b1: The corresponding interrupt is enabled. Writes 1'b0: Has no effect. 1'b1: Disables the corresponding interrupt. A subsequent read of this bit returns the value 0.

For interrupt ID N, when DIV and MOD are the integer division and modulo operations:
• the corresponding ICDICER number, M, is given by M = N DIV 32
• the offset of the required ICDICER is (0x180 + (4*M))
• the bit number of the required Clear-enable bit in this register is N MOD 32.

GICD_ICDISPR
Address:Operational Base+0x200
Interrupt Set-Pending Registers

Bit	Attr	Reset Value	Description
31:0	RW	0x0	For each bit: Reads 1'b0: The corresponding interrupt is not pending on any processor. 1'b1: • For SGIs, the corresponding interrupt is pending on this processor. • For SPIs, the corresponding interrupt is pending on at least one processor. Writes For SPIs: 1'b0: Has no effect. 1'b1: The effect depends on whether the interrupt is edge-triggered or level-sensitive: Edge-triggered Changes the status of the corresponding interrupt to: • pending if it was previously inactive • active and pending if it was previously active.Has no effect if the interrupt is already pending. Level sensitive If the corresponding interrupt is not pendinga, changes the status of the corresponding interrupt to: • pending if it was previously inactive • active and pending if it was previously active.

			If the interrupt is already pending: • because of a write to the ICDISPR, the write has no effect • because the corresponding interrupt signal is asserted, the write has no effect on the status of the interrupt, but the interrupt remains pendinga if the interrupt signal is deasserted.

For interrupt ID N, when DIV and MOD are the integer division and modulo operations:
• the corresponding ICDISPR number, M, is given by M = N DIV 32
• the offset of the required ICDISPR is (0x200 + (4*M))
• the bit number of the required Set-pending bit in this register is N MOD 32.

GICD_ICDICPR
Address:Operational Base+0x280
Interrupt Clear-Pending Registers

Bit	Attr	Reset Value	Description
31:0	RW	0x0	For each bit: Reads 1'b0: The corresponding interrupt is not pending on any processor 1'b1: • For SGIs, the corresponding interrupt is pendinga on this processor. • For SPIs, the corresponding interrupt is pendinga on aleast one processor. Writes For SPIs: 1'b0: Has no effect. 1'b1: The effect depends on whether the interrupt is edge-triggeredor level-sensitive: Edge-triggered Changes the status of the corresponding interrupto: • inactive if it was previously pending • active if it was previously active and pending.Has no effect if the interrupt is not pending. Level-sensitive If the corresponding interrupt is pendinga only because of a write to the ICDISPR, the write changes the status of the interrupt to: • inactive if it was previously pending • active if it was previously active and pending.Otherwise the interrupt remains pending if the interrupt signal remains asserted.

For interrupt ID N, when DIV and MOD are the integer division and modulo operations:
• the corresponding ICDICPR number, M, is given by M = N DIV 32
• the offset of the required ICDICPR is (0x280 + (4*M))

• the bit number of the required Set-pending bit in this register is N MOD 32.

GICD_ICDABR
Address:Operational Base+0x300
Active Bit Registers

Bit	Attr	Reset Value	Description
31:0	R		For each bit: 1'b0: Corresponding interrupt is not active. 1'b1: Corresponding interrupt is active.

For interrupt ID N, when DIV and MOD are the integer division and modulo operations:
• the corresponding ICDABR number, M, is given by M = N DIV 32
• the offset of the required ICDABR is (0x300 + (4*M))
• the bit number of the required Active bit in this register is N MOD 32.

GICD_ICDIPR
Address:Operational Base+0x400
Interrupt Priority Registers

Bit	Attr	Reset Value	Description
7:0	RW	0x0	The lower the value, the greater the priority of the corresponding interrupt.

For interrupt ID N:
• the corresponding ICDIPR number, M, is given by M = N
• the offset of the required ICDIPR is (0x400 + M)

GICD_ICDIPTR
Address:Operational Base+0x800
Interrupt Processor Targets Registers

Bit	Attr	Reset Value	Description
7:0	RW	0x1	This register is not used. As in ourproduct ,there is only one processor.

GICD_ICDICFR
Address:Operational Base+0xc00
Interrupt Configuration Registers

Bit	Attr	Reset Value	Description
2F+1	RW	0x0	F=0,1,2,3….15 The encoding is: 1'b0: Corresponding interrupt is level-sensitive. 1'b1: Corresponding interupt is edge-triggered.

For interrupt ID N, when DIV and MOD are the integer division and modulo operations:
• the corresponding ICDICFR number, M, is given by M = N DIV 16
• the offset of the required ICDIPTR is (0xC00 + (4*M))
• the required Priority field in this register, F, is given by F = N MOD 16, where field 0 refers to register bits [1:0], field 1 refers to bits [3:2], and so on, up to field 15 refers to bits [31:30]

GICD_ICDSGIR
Address:Operational Base+0xf00
Software Generated Interrupt Register

Bit	Attr	Reset Value	Description

31:26	-	-	reserved
25:24	W	0x0	2'b00: Send the interrupt to the CPU interface specified in the CPUTargetListfielda. 2'b01: Send the interrupt to all CPU interface except the CPU interface that requested the interrupt. 2'b10: Send the interrupt only to the CPU interface that requested the interrupt. 2'b11: Reserved
23:16	W	0x0	When TargetList Filter = 2'bb00, defines the CPU interface the Distributor must send the interrupt to. Each bit of CPUTargetList[7:0] refers to the corresponding CPU interface, for example CPUTargetList[0] corresponds to CPU interface 0. Setting a bit to 1 sends the interrupt to the corresponding interface.
15	W	0x0	If the GIC implements the Security Extensions, this field is writable only using a Secure access. Any Non-secure write to the ICDSGIR issues an SGI only if the specified SGI is programmed as Non-secure, regardless of the value of bit [15] of the write. Specifies the required security value of the SGI: 1'b0: Send the SGI specified in the SGIINTID field to a specified CPU interface only if the SGI is configured as Secure on that interface. 1'b1: Send the SGI specified in the SGIINTID field to a specified CPU interface only if the SGI is configured as Non-secure on that interface
14:4	-	-	reserved
3:0	W	0x0	The Interrupt ID of the SGI to send to the specified CPU interface. The value of this field is the Interrupt ID, in the range 0-15, for example a value of 4'b0011 specifies Interrupt ID 3

9.4.3 GIC CPU interface register summary

Name	Offset	Size	Reset Value	Description
GICC_ICCICR	0x00	W	0x0	CPU Interface Control Register
GICC_ICCPMR	0x04	W	0x0	Interrupt Priority Mask Register
GICC_ICCBPR	0x08	W	0x0	Binary Point Register
GICC_ICCIAR	0x0C	W	0x3ff	Interrupt Acknowledge Register
GICC_ICCEOIR	0x10	W	-	End of Interrupt Register
GICC_ICCRPR	0x14	W	0xff	Running Priority Register
GICC_ICCHPIR	0x18	W	0x3ff	Highest Pending Interrupt Register
GICC_ICCABPR	0x1C	W	0x0	Aliased Binary Point

				Register
GICC_ICCIIDR	0xFC	W	0x0	CPU Interface Identification Register

Notes:
*Size: **B** – Byte (8 bits) access, **HW** – Half WORD (16 bits) access, **W** –WORD (32 bits) access*

9.4.4 GIC CPU interface detail register description

GICC_ICCICR
Address:Operational Base+0x0
CPU Interface Control Register

Bit	Attr	Reset Value	Description
31:5	-	-	reserved
4	RW	0x0	Controls whether the CPU interface uses the Secure or Non-secure Binary Point Register for preemption. 1'b0: To determine any preemption, use: • the Secure Binary Point Register for Secure interrupts • the Non-secure Binary Point Register for Non-secure interrupts. 1'b1: To determine any preemption use the Secure Binary Point Register for both Secure and Non-secure interrupts.
3	RW	0x0	Controls whether the GIC signals Secure interrupts to a target processor using the FIQ or the IRQ signal. 1'b0: Signal Secure interrupts using the IRQ signal. 1'b1: Signal Secure interrupts using the FIQ signal. The GIC always signals Non-secure interrupts using the IRQ signal.
2	RW	0x0	Controls whether a Secure read of the ICCIAR, when the highest priority pending interrupt is Non-secure, causes the CPU interface to acknowledge the interrupt. 1'b0: If the highest priority pending interrupt is Non-secure, a Secure read of the ICCIAR returns an Interrupt ID of 1022. The read does not acknowledge the interrupt, and the pending status of the interrupt is unchanged. 1'b1: If the highest priority pending interrupt is Non-secure, a Secure read of the ICCIAR returns the Interrupt ID of the Non-secure interrupt. The read acknowledges the interrupt, and the status of the interrupt becomes active, or active and pending.

1	RW	0x0	An alias of the Enable bit in the Non-secure ICCICR. This alias bit means Secure software can enable the signalling of Non-secure interrupts. 1'b0: Disable signalling of Non-secure interrupts. 1'b1: Enable signalling of Non-secure interrupts
0	RW	0x0	Global enable for the signalling of Secure interrupts by the CPU interface to the connected processors. 1'b0: Disable signalling of Secure interrupts. 1'b1: Enable signalling of Secure interrupts

GICC_ICCPMR

Address:Operational Base+0x4

Interrupt Priority Mask Register

Bit	Attr	Reset Value	Description
31:8	-	-	reserved
7:0	RW	0x0	The priority mask level for the CPU interface. If the priority of an interrupt is higher than the value indicated by this field, the interface signals the interrupt to the processor. If the GIC supports fewer than 256 priority levels then some bits are read as zero(RAZ)/write ignore(WI), as follows: 128 supported levels Bit [0] = 1'b0. 64 supported levels Bit [1:0] = 2'b00. 32 supported levels Bit [2:0] = 3'b000. 16 supported levels Bit [3:0] = 4'b0000

GICC_ICCBPR

Address:Operational Base+0x8

Binary Point Register

Bit	Attr	Reset Value	Description
31:3	-	-	reserved
2:0	RW	0x0	The value of this field controls how the 8-bit interrupt priority field is split into a group priority field, used to determine interrupt preemption, and a subpriority field.

GICC_ICCIAR

Address:Operational Base+0xc

Interrupt Acknowledge Register

Bit	Attr	Reset Value	Description
31:13	-	-	reserved
12:10	RO	0x0	For SGIs in a multiprocessor implementation, this field identifies the processor that requested the interrupt. It returns the number of the CPU interface that made the request, for example a value of 3 (3'b011) means the request was generated by a write to the IDCSFGIR on CPU interface 3. For all other interrupts this field is RAZ.

9:0	RO	0x0	The interrupt ID.

GICC_ICCEOIR
Address:Operational Base+0x10
End of Interrupt Register

Bit	Attr	Reset Value	Description
31:13	-	-	reserved
12:10	WO	0x0	On a multiprocessor implementation, on completion of the processing of an SGI, this field contains the CPUID value from the corresponding ICCIAR access.
9:0	WO	0x0	The ACKINTID value from the corresponding ICCIAR access.

GICC_ICCRPR
Address:Operational Base+0x14
Running Priority Register

Bit	Attr	Reset Value	Description
31:8	-	-	reserved
7:0	RO	0x0	The priority value of the highest priority interrupt that is active on the CPU interface.

GICC_ICCABPR
Address:Operational Base+0x18
Aliased Binary Point Register

Bit	Attr	Reset Value	Description
31:3	-	-	reserved
2:0	RW	0x0	Provides an alias of the Non-secure ICCBPR.

GICC_ICCHPIR
Address:Operational Base+0x1c
Highest Pending Interrupt Register

Bit	Attr	Reset Value	Description
31:10	-	-	reserved
9:0	R	0x0	The interrupt ID of the highest priority pending interrupt.

GICC_ICCIIDR
Address:Operational Base+0xfc
CPU Interface Identification Register

Bit	Attr	Reset Value	Description
31:20	R	0x390	An IMPLEMENTATION DEFINED product identifier.
19:16	R	0x1	For an implementation that complies with this specification, the value is 0x1
15:12	R	0x2	An IMPLEMENTATION DEFINED revision number for the CPU interface.
11:0	R	0x43B	Contains the JEP106 code of the company that implemented the GIC CPU interface:b Bits [11:8]: The JEP106 continuation code of the implementer. Bit [7]: Always 0.

			Bits [6:0]: The JEP106 identity code of the implementer.

9.5 Interface Description

Both distributor interface and CPU interface are secure accessed only after reset.

When the signal cfgsdisable is HIGH, it enhances the security of the GIC by preventing write accesses to security-critical configuration registers.This signal is low after reset, it can be configured through TZPC registers.

9.6 Application Notes

9.6.1 General handling of interrupts

The GIC operates on interrupts as follows:

1. The GIC determines whether each interrupt is enabled. An interrupt that is not enabled has no further effect on the GIC.(Enables an interrupt by writing to the appropriate ICDISER bit, disables an interrupt by writing to the appropriate ICDICER bit)

2. For each enabled interrupt that is pending, the Distributor determines the targeted processor.

3. For processor, the Distributor determines the highest priority pending interrupt, based on the priority information it holds for each interrupt, and forwards the interrupt to the CPU interface.

4. The CPU interface compares the interrupt priority with the current interrupt priority for the processor, determined by a combination of the Priority Mask Register, the current preemption settings, and the highest priority active interrupt for the processor. If the interrupt has sufficient priority, the GIC signals an interrupt exception request to the processor.

5. When the processor takes the interrupt exception, it reads the ICCIAR in its CPU interface to acknowledge the interrupt. This readreturns an Interrupt ID that the processor uses to select the correct interrupt handler. When it recognizes this read, the GIC changes the state of the interrupt:

- If the pending state of the interrupt persists when the interrupt becomes active, or if the interrupt is generated again, from pending to active and pending.
- Otherwise, from pending to active

6.When the processor has completed handling the interrupt, it signals this completion by writing to the ICCEOIR in the GIC

Generating an SGI

A processor generates an SGI by writing to an ICDSGIR.

9.6.2 Interrupt prioritization

Software configures interrupt prioritization in the GIC by assigning a priority value to each interrupt source. Priority values are 8-bit unsigned binary.

In this product, GIC implements 64 priority levels. So only the highest 6 bits are valid, the lower 2 bits read as zero.

In the GIC prioritization scheme, lower numbers have higher priority, that is, the lower the assigned priority value the higher the priority of the interrupt. The highest interrupt priority always has priority field value 0.

The ICDIPRs hold the priority value for each supported interrupt. To determine the number of priority bits implemented write 0xFF to an ICDIPR priority field and read back the value stored.

Preemption

A CPU interface supports forwarding of higher priority pending interrupts to a target processor before an active interrupt completes. A pending interrupt is only forwarded if it has a higher priority than all of:

- the priority of the highest priority active interrupt on the target processor, the running priority for the processor, see Running Priority Register (ICCRPR) .
- The priority mask, see Priority masking.
- The priority group, see Priority grouping.

Preemption occurs at the time when the processor acknowledges the new interrupt, and starts to service it in preference to the previously active interrupt or the currently running process. When this occurs, the initial active interrupt is said to have been preempted. Starting to service an interrupt while another interrupt is still active is sometimes described as interrupt nesting.

Priority masking

The ICCPMR for a CPU interface defines a priority threshold for the target processor, see Interrupt Priority Mask Register. The GIC only signals pending interrupts with a higher priority than this threshold value to the target processor. A value of zero, the register reset value, masks all interrupts to the associated processor.

The GIC always masks an interrupt that has the largest supported priority field value. This provides an additional means of preventing an interrupt being signalled to any processor.

Priority grouping

Priority grouping splits each priority value into two fields, the group priority and the subpriority fields. The GIC uses the group priority field to determine whether a pending interrupt has sufficient priority to preempt a currently active interrupt.

The binary point field in the ICCBPR controls the split of the priority bits into the two parts. This 3-bit field specifies how many of the least significant bits of the 8-bit interrupt priority field are excluded from the group priority field, as following table shows.

Binary point value	Group priority field	Subpriority field	Field with binary point
0	[7:1]	[0]	ggggggg.s
1	[7:2]	[1:0]	gggggg.ss
2	[7:3]	[2:0]	ggggg.sss
3	[7:4]	[3:0]	gggg.ssss
4	[7:5]	[4:0]	ggg.sssss
5	[7:6]	[5:0]	gg.ssssss
6	[7]	[6:0]	g.sssssss
7	No preemption	[7:0]	.sssssss

Where multiple pending interrupts share the same group priority, the GIC uses the subpriority field to resolve the priority within a group.

9.6.3 The effect of the Security Extensions on interrupt handling

If a GIC CPU interface implements the Security Extensions, it provides two interrupt output signals, IRQ and FIQ:

- The CPU interface always uses the IRQ exception request for Non-secureinterrupts
- Software can configure the CPU interface to use either IRQ or FIQ exception requests for Secure interrupts.

Security Extensions support

Software can detect support for the Security Extensions by reading the ICDICTR.SecurityExtn bit, see Interrupt Controller Type Register (ICDICTR).

Secure software makes Secure writes to the ICDISRs to configure each interrupt as Secure or Non-secure, see Interrupt Security Registers (ICDISRn).

In addition:

- The banking of registers provides independent control of Secure and
- Non-secure interrupts.
- The Secure copy of the ICCICR has additional fields to control the processing of Secure and Non-secure interrupts, see CPU Interface Control Register (ICCICR) These fields are:
  - the SBPR bit, that affects the preemption of Non-secure interrupts.
  - the FIQEn bit, that controls whether the interface signals Secure interrupt to the processor using the IRQ or FIQ interrupt exception requests.
  - the AckCtl bit, that affects the acknowledgment of Non-secure interrupts.
  - the EnableNS bit, that controls whether Non-secure interrupts are signaled to the processor, and is an alias of the Enable bit in the Non-secure ICCICR.
- The Non-secure copy of the ICCBPR is aliased as the ICCABPR, see Aliased
- Binary Point Register (ICCABPR). This is a Secure register, meaning it is only
- accessible by Secure accesses.

Effect of the Security Extensions on interrupt acknowledgement

When a processor takes an interrupt, it acknowledges the interrupt by reading the ICCIAR. A read of the ICCIAR always acknowledges the highest priority pending interrupt for the processor performing the read.

If the highest priority pending interrupt is a Secure interrupt, the processor must make a Secure read of the ICCIAR to acknowledge it.

By default, the processor must make a Non-secure read of the ICCIAR to acknowledge a Non-secure interrupt. If he AckCtl bit in the Secure ICCICR is set to 1 the processor can make a Secure read of the ICCIAR to acknowledge a Non-secure interrupt.

If the read of the ICCIAR does not match the security of the interrupt, taking account of the AckCtl bit value for a Non-secure interrupt, the ICCIAR read does

not acknowledge any interrupt and returns the value:

- 1022 for a Secure read when the highest priority interrupt is Non-secure
- 1023 for a Non-secure read when the highest priority interrupt is Secure.