

## GICC\_IAR, CPU Interface Interrupt Acknowledge Register

The GICC\_IAR characteristics are:

### Purpose

Provides the INTID of the signaled interrupt. A read of this register by the PE acts as an acknowledge for the interrupt.

### Configuration

This register is present only when FEAT\_GICv3\_LEGACY is implemented. Otherwise, direct accesses to GICC\_IAR are res0.

This register is available in all configurations of the GIC. If [GICD\\_CTLR.DS](#)=0:

- This register is Common.
- [GICC\\_AIAR](#) is an alias of the Non-secure view of this register.

The format of the INTID is governed by whether affinity routing is enabled for a Security state.

### Attributes

GICC\_IAR is a 32-bit register.

### Field descriptions

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
RES0								INTID																							

#### Bits [31:24]

Reserved, res0.

#### INTID, bits [23:0]

The INTID of the signaled interrupt.

---

#### Note

INTIDs 1020-1023 are reserved and convey additional information such as spurious interrupts.

---

When affinity routing is not enabled:

- Bits [23:13] are res0.
- For SGIs, bits [12:10] identify the CPU interface corresponding to the source PE. For all other interrupts these bits are res0.

A read of this register returns the INTID of the highest priority pending interrupt for the CPU interface. The read returns a spurious INTID of 1023 if any of the following apply:

- Forwarding of interrupts by the Distributor to the CPU interface is disabled.
- Signaling of interrupts by the CPU interface to the connected PE is disabled.
- There are no pending interrupts on the CPU interface with sufficient priority for the interface to signal it to the PE.

When the GIC returns a valid INTID to a read of this register it treats the read as an acknowledge of that interrupt. In addition, it changes the interrupt status from pending to active, or to active and pending if the pending state of the interrupt persists. Normally, the pending state of an interrupt persists only if the interrupt is level-sensitive and remains asserted.

For every read of a valid INTID from GICC\_IAR, the connected PE must perform a matching write to [GICC\\_EOIR](#).

---

### **Note**

- Arm recommends that software preserves the entire register value read from this register, and writes that value back to [GICC\\_EOIR](#) on completion of interrupt processing.
  - For SPIs, although multiple target PEs might attempt to read this register at any time, only one PE can obtain a valid INTID. For more information, see 'Activation' in ARM® Generic Interrupt Controller Architecture Specification, GIC architecture version 3.0 and version 4.0 (ARM IHI 0069).
-

# Accessing GICC\_IAR

When [GICD\\_CTLR.DS](#)==1, if the highest priority pending interrupt is in Group 1, the special INTID 1022 is returned.

In GIC implementations that support two Security states, if the highest priority pending interrupt is in Group 0, Non-secure reads return the special INTID 1023.

In GIC implementations that support two Security states, if the highest priority pending interrupt is in Group 1, Secure reads return the special INTID 1022.

This register is used only when System register access is not enabled. When System register access is enabled:

- For AArch32 implementations, [ICC\\_IAR0](#) and [ICC\\_IAR1](#) provide equivalent functionality.
- For AArch64 implementations, [ICC\\_IAR0\\_EL1](#) and [ICC\\_IAR1\\_EL1](#) provide equivalent functionality.

When affinity routing is enabled for a Security state, it is a programming error to use memory-mapped registers to access the GIC.

## GICC\_IAR can be accessed through the memory-mapped interfaces:

Component	Offset	Instance
GIC CPU interface	0x000C	GICC_IAR

This interface is accessible as follows:

- When [GICD\\_CTLR.DS](#) == 0, accesses to this register are **RO**.
- When an access is Secure, accesses to this register are **RO**.
- When an access is Non-secure, accesses to this register are **RO**.

[AArch32  
Registers](#)

[AArch64  
Registers](#)

[AArch32  
Instructions](#)

[AArch64  
Instructions](#)

[Index by  
Encoding](#)

[External  
Registers](#)

28/03/2023 16:02; 72747e43966d6b97dcbd230a1b3f0421d1ea3d94

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