Arm® CoreLink™ GIC-700 Generic Interrupt Controller

Software Developer Errata Notice

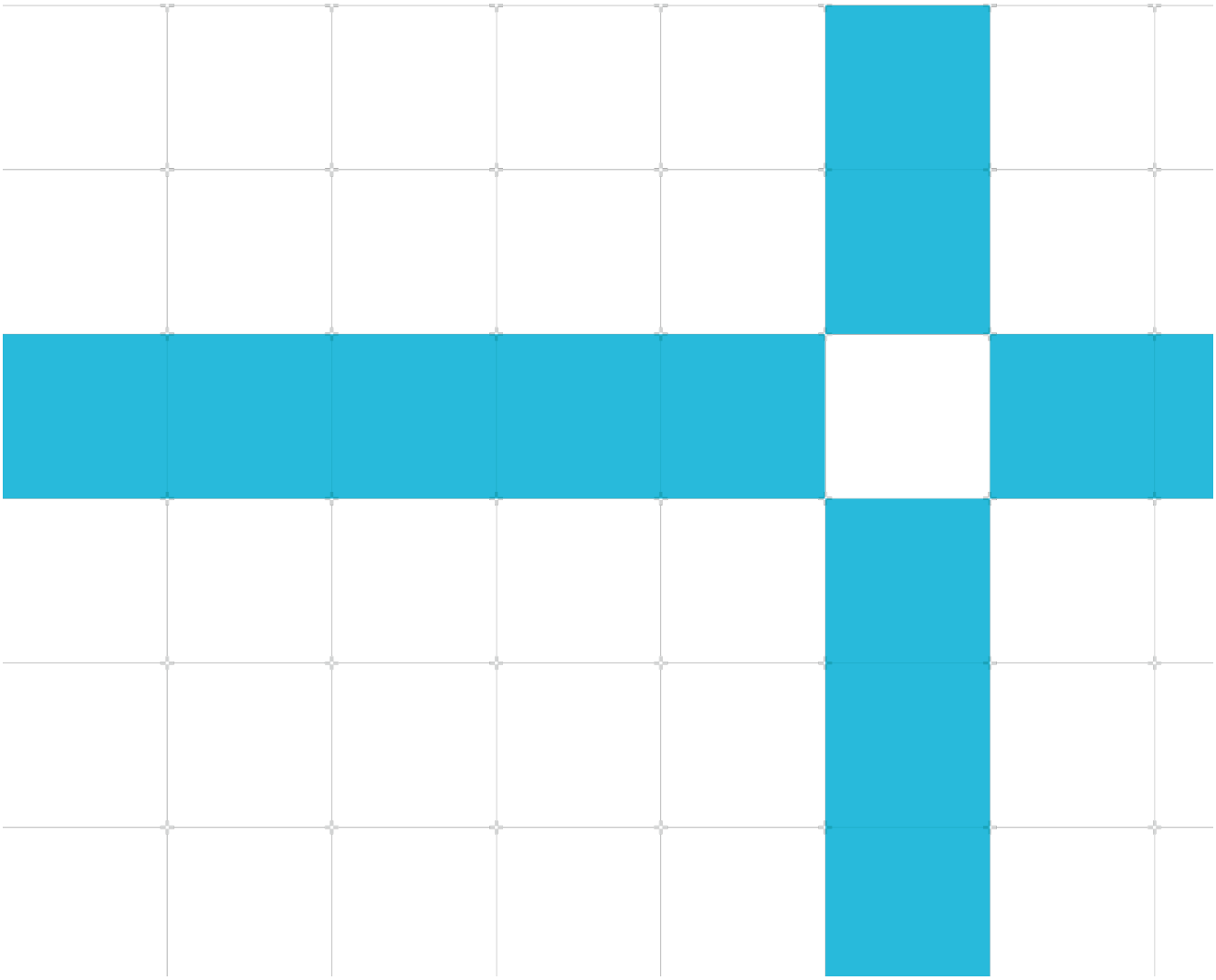
Date of issue: 25-May-2022

Non-Confidential

Document version: v5.0

Document ID: SDEN-1769194

This document contains all known errata since the r0p0 release of the product.



Non-confidential proprietary notice

Arm Limited. Company 02557590 registered in England.

110 Fulbourn Road, Cambridge, England CB1 9NJ.

(LES-PRE-20349)

Confidentiality status

This document is Non-Confidential. The right to use, copy and disclose this document may be subject to license restrictions in accordance with the terms of the agreement entered into by Arm and the party that Arm delivered this document to.

Unrestricted Access is an Arm internal classification.

Feedback

Arm welcomes feedback on this product and its documentation. To provide feedback on Arm® CoreLink™ GIC-700 Generic Interrupt Controller, create a ticket on **https://support.developer.arm.com**.

To provide feedback on the document, fill the following survey: **https://developer.arm.com/documentation-feedback-survey**.

Inclusive language commitment

Arm values inclusive communities. Arm recognizes that we and our industry have used language that can be offensive. Arm strives to lead the industry and create change.

If you find offensive language in this document, please email **terms@arm.com**.

Contents

Introduction

Scope

This document describes errata categorized by level of severity. Each description includes:

- The current status of the erratum.
- Where the implementation deviates from the specification and the conditions required for erroneous behavior to occur.
- The implications of the erratum with respect to typical applications.
- The application and limitations of a workaround where possible.

Categorization of errata

Errata are split into three levels of severity and further qualified as common or rare:

Category A	A critical error. No workaround is available or workarounds are impactful. The error is likely to be common for many systems and applications.
Category A (Rare)	A critical error. No workaround is available or workarounds are impactful. The error is likely to be rare for most systems and applications. Rare is determined by analysis, verification and usage.
Category B	A significant error or a critical error with an acceptable workaround. The error is likely to be common for many systems and applications.
Category B (Rare)	A significant error or a critical error with an acceptable workaround. The error is likely to be rare for most systems and applications. Rare is determined by analysis, verification and usage.
Category C	A minor error.

Change Control

Errata are listed in this section if they are new to the document, or marked as "updated" if there has been any change to the erratum text. Fixed errata are not shown as updated unless the erratum text has changed. The **errata summary table** identifies errata that have been fixed in each product revision.

25-May-2022: Changes in document version v5.0

ID	Status	Area	Category	Summary
2289578	New	Programmer	Category C	Read error reporting does not include rpoison in vTGT and PTS
2381241	New	Programmer	Category C	RAS Record 0 can incorrectly report overflow due to masked Deactivates.
2381076	New	Programmer	Category C	RSS Residency Completion TD event lost for coincident events

25-Jun-2021: Changes in document version v4.0

ID	Status	Area	Category	Summary
2141155	Updated	Programmer	Category B	SPI pipe failing to return SPIs to owning chip when all GICD group enables are dropped.
2169019	New	Programmer	Category B	Doorbell missed due to an LPI command
2172285	New	Programmer	Category B	GIC can fail to issue clear commands to recall interrupts
2195890	New	Programmer	Category B	LPIs may be held in Pending Table until specific external events happen
2032913	Updated	Programmer	Category C	Deadlock caused by response to wrong chip during cross-chip MOVALL with 3 or more chips
2121183	Updated	Programmer	Category C	ITS DCACHE and CCACHE PMU MISS EVENTS are pessimistic
2181357	New	Programmer	Category C	Interrupt may miss search after command on non-resident vPE
2185600	New	Programmer	Category C	Doorbell may not be masked by Activate resulting in duplicate doorbell

06-Apr-2021: Changes in document version v3.0

ID	Status	Area	Category	Summary
2141155	New	Programmer	Category B	SPI pipe failing to return SPIs to owning chip when all GICD group enables are dropped.
2032913	New	Programmer	Category C	Deadlock caused by response to wrong chip during cross-chip MOVALL with 3 or more chips
2121183	New	Programmer	Category C	ITS DCACHE and CCACHE PMU MISS EVENTS are pessimistic

12-Nov-2020: Changes in document version v2.0

ID	Status	Area	Category	Summary
1978030	New	Programmer	Category B	Deadlock caused by VMOVP commands forming a loop between chips
1985213	New	Programmer	Category B (rare)	Issues caused by repeated VMOVP commands of the same vPE in systems with 3 or more chips
2012501	New	Programmer	Category C	Unrecoverable DED error in TGT or VTGT may result in interrupts being left in PT
1957262	New	Programmer	Category C	Reported error data is corrupted for PPI RAM (error records 7&8)
1957948	New	Programmer	Category C	Target range check for MAPC/MOVALL/VMAPP/VMOVP ignores bits[51:48] of Rdbase field
1968775	New	Programmer	Category C	Unpredictable SW access to GICR_INVLPIR or GICR_INVALLR after VTGT DED error can cause vPE corruption
1980415	New	Programmer	Category C	vPE corruption due to vPT read error during VMOVI
1981453	New	Programmer	Category C	PMU filtering incorrect between virtual and physical interrupts
1991967	New	Programmer	Category C	Incorrect debug data on some GICx_ERRINSR registers
1995446	New	Programmer	Category C	Affinity3 field corrupted by cross-chip 32 bit writes to upper half of GICD_IROUTERn(E)

17-Jul-2020: Changes in document version v1.0

ID	Status	Area	Category	Summary
1809145	New	Programmer	Category B	Interrupt missed by search due to matching INV or INVALL
1882328	New	Programmer	Category B	LPI commands running in parallel with a matching INV might not take effect
1808803	New	Programmer	Category C	DED error in VICM RAM can cause data corruption after aborted flush
1801287	New	Programmer	Category C	ECC error can cause incorrect data to be logged for GICR_ERRINSR debug register access
1806339	New	Programmer	Category C	LPI Merge PMU incorrectly also counts LPI command matches
1859709	New	Programmer	Category C	Multiple concurrent ECC errors can cause LPI loss during GICD_ERRINSR debug register write
1871535	New	Programmer	Category C	GICT_ERR0ADDR.PADDR has non-standard 32-bit write behavior
1886083	New	Programmer	Category C	Reset issue can cause read to 0x0 during coincident GITS_SGIR write and VMAPP(V1A0) command to offline chip
1892861	New	Programmer	Category C	Errored RAM data not reported for VLPIs
1900358	New	Programmer	Category C	ClockGating and Q-Channel are too pessimistic

Errata summary table

The errata associated with this product affect the product versions described in the following table.

ID	Area	Category	Summary	Found in versions	Fixed in version
2195890	Programmer	Category B	LPIs may be held in Pending Table until specific external events happen	r0p0, r0p1, r1p0	r2p0
2172285	Programmer	Category B	GIC can fail to issue clear commands to recall interrupts	r0p0, r0p1, r1p0	r2p0
2169019	Programmer	Category B	Doorbell missed due to an LPI command	r0p0, r0p1, r1p0	r2p0
2141155	Programmer	Category B	SPI pipe failing to return SPIs to owning chip when all GICD group enables are dropped.	r0p0, r0p1, r1p0	r2p0
1978030	Programmer	Category B	Deadlock caused by VMOVP commands forming a loop between chips	r0p0, r0p1	r1p0
1882328	Programmer	Category B	LPI commands running in parallel with a matching INV might not take effect	r0p0	r0p1
1815972	Programmer	Category B	Interrupt missed by search due to matching INV or INVALL	r0p0	r0p1
1985213	Programmer	Category B (rare)	Issues caused by repeated VMOVP commands of the same vPE in systems with 3 or more chips	r0p0, r0p1	r1p0
2381076	Programmer	Category C	RSS Residency Completion TD event lost for coincident events	r0p0, r0p1, r1p0, r2p0	r2p1
2381241	Programmer	Category C	RAS Record 0 can incorrectly report overflow due to masked Deactivates.	r1p0, r2p0	r2p1
2289578	Programmer	Category C	Read error reporting does not include rpoison in vTGT and PTS	r0p0, r0p1, r1p0, r2p0	r2p1
2185600	Programmer	Category C	Doorbell may not be masked by Activate resulting in duplicate doorbell	r0p0, r0p1, r1p0	r2p0
2181357	Programmer	Category C	Interrupt may miss search after command on non-resident vPE	r0p0, r0p1, r1p0	r2p0
2121183	Programmer	Category C	ITS DCACHE and CCACHE PMU MISS EVENTS are pessimistic	r0p0, r0p1, r1p0	r2p0
2032913	Programmer	Category C	Deadlock caused by response to wrong chip during cross-chip MOVALL with 3 or more chips	r0p0, r0p1, r1p0	r2p0
1995446	Programmer	Category C	Affinity3 field corrupted by cross-chip 32 bit writes to upper half of GICD_IROUTERn(E)	r0p0, r0p1	r1p0
1991967	Programmer	Category C	Incorrect debug data on some GICx_ERRINSR registers	r0p0, r0p1	r1p0

ID	Area	Category	Summary	Found in versions	Fixed in version
1981453	Programmer	Category C	PMU filtering incorrect between virtual and physical interrupts	r0p0, r0p1	r1p0
1980415	Programmer	Category C	vPE corruption due to vPT read error during VMOVI	r0p0, r0p1	r1p0
1968775	Programmer	Category C	Unpredictable SW access to GICR_INVLPIR or GICR_INVALLR after VTGT DED error can cause vPE corruption	r0p0, r0p1	r1p0
1957948	Programmer	Category C	Target range check for MAPC/MOVALL/VMAPP/VMOVP ignores bits[51:48] of Rdbase field	r0p0, r0p1	r1p0
1957262	Programmer	Category C	Reported error data is corrupted for PPI RAM (error records 7&8)	r0p0, r0p1	r1p0
1912069	Programmer	Category C	Unrecoverable DED error in TGT or VTGT may result in interrupts being left in PT	r0p0, r0p1	r1p0
1900358	Programmer	Category C	ClockGating and Q-Channel are too pessimistic	r0p0	r0p1
1892861	Programmer	Category C	Errored RAM data not reported for VLPIs	r0p0, r0p1	r1p0
1886083	Programmer	Category C	Reset issue can cause read to 0x0 during coincident GITS_SGIR write and VMAPP(V1A0) command to offline chip	r0p0, r0p1	r1p0
1871535	Programmer	Category C	GICT_ERR0ADDR.PADDR has non-standard 32-bit write behavior	r0p0, r0p1	r1p0
1859709	Programmer	Category C	Multiple concurrent ECC errors can cause LPI loss during GICD_ERRINSR debug register write	r0p0, r0p1	r1p0
1806339	Programmer	Category C	LPI Merge PMU incorrectly also counts LPI command matches	r0p0, r0p1	r1p0
1801287	Programmer	Category C	ECC error can cause incorrect data to be logged for GICR_ERRINSR debug register access	r0p0, r0p1	r1p0
1733884	Programmer	Category C	DED error in VICM RAM can cause data corruption after aborted flush	r0p0, r0p1	r1p0

Errata descriptions

Category A

There are no errata in this category.

Category A (rare)

There are no errata in this category.

Category B

2195890
LPIs may be held in Pending Table until specific external events happen

Status

Affects: GIC-700
Fault Type: Programmer Cat B
Fault Status: Present in: r0p0, r0p1, r1p0 Fixed in: r2p0

Description

If the conditions described below occur, then LPIs may remain in the Pending Table until one of a number of external events occurs.

No LPIs are lost and this can happen on physical or virtual PEs but this erratum means they may not be delivered in a finite time.

Configurations impacted

Any configuration with lpi_support

Conditions

The following conditions must all be met:

- The LPI cache is overflowing to the Pending Tables.
- There are more than 4 pending enabled interrupts to a (v)PE within the same aligned block of 32 LPI IDs.
- There are no other blocks in the PT that have not yet been searched that have more than 4 pending enabled interrupts.
- A second instance of at least one of those 4 interrupts arrives and the GIC has not yet merged the instances.
- There are no more than 2 other interrupts waiting to be delivered to the same (v)PE.
- The CPU activates multiple LPIs faster than they are extracted from the PT (This time depends on GIC loading and memory access delays).

Implications

Interrupts for a (v)PE may be left in the Pending Table until one of the following events happens:

- The number of pending LPIs for the (v)PE reaches 3 at the same time (excluding ones already

searched in the PT).
- An LPI command is run (**INV**, **MOV** to the (v)PE) for any LPI ID (From r0p1 onwards).
- A residency change occurs - There are no implications for doorbell generation.

Workaround

For physical LPIs the workaround is to issue an **INV** using GICR_INVLPIR to an unused, in range LPI ID to retrigger the search.

This could be done periodically, for example, in line with a residency change, or as part of servicing LPIs. If using LPIs as the event, then the GICR_INVLPIR write could be issued after servicing every LPI. However, it only needs to be issued if:

- At least 4 interrupts in the block of 32 are enabled and mapped to the current PE or, if easier,
- At least 4 interrupts in the block of 32 are enabled and mapped to any PE.

Alternatively, GICR_INVLPIR writes after LPIs are not required if there is a subsequent LPI waiting to be delivered in the CPU interface (i.e. a LPI ID is in ICC_HPPIR1_EL1 at the time of the priority drop of the serviced LPI). This is only true if using a single LPI priority and not using the filtering described above.

No workaround is expected to be required for vPEs (unless pinned 1:1 to PEs and never scheduled in and out by the hypervisor) as the normal residency flow will cause the interrupts to be delivered and doorbells will be delivered correctly.

2172285
GIC can fail to issue clear commands to recall interrupts

Status

Affects: GIC-700
Fault Type: Programmer Cat B
Fault Status: Present in: r0p0, r0p1, r1p0 Fixed in: r2p0

Description

When certain events occur, the GIC is meant to recall an interrupt by either sending a different interrupt or sending a **`CLEAR`** command to the CPU. Recalling the interrupt allows the ITS command or associated poll to complete. The events that cause the GIC to recall an interrupt are any of the following:

- LPI command (**`MOV`**, **`CLR`**, **`INV`** (on enabled interrupt), **`INVDB`**) matching an LPI that is sent to a CPU.
- GICD_ICENABLERn matching an SPI that is sent to a CPU.
- GICD_CTLR Group enables drop matching any interrupt sent to a CPU.
- Making a vPE resident when a doorbell is requested but not yet serviced.

In some circumstances, the GIC fails to actively recall the interrupt and software is stalled while it waits for RWP to be zero, the ITS command queue to progress or the residency change to complete.

Configurations impacted

Any configuration with LPI support.

Conditions

This erratum can occur if either of the following sequences occur:

Sequence A

1. An SPI or doorbell is sent to a PE.
2. An LPI is sent to the same PE before the interrupt in 1) is activated. This could be because the LPI is of a higher priority, or the SPI/doorbell becomes disabled for any reason.
3. Another LPI is pending and enabled that has the same priority as the interrupt in step 2), or the priority of the interrupt in step 2) is **`0xF8`**.
4. An LPI command occurs that matches the LPI sent in step 2) before the interrupt in step 1) is released by the PE. If the interrupt in 1) is activated, this erratum does not occur.

This sequence stalls the software. The stall can be removed by either:

- Activation of the LPI being recalled.
- Any incoming LPI of a priority high enough to match or exceed the top 3 LPIs for the target PE.

- CPU group enable toggle on the target PE or GICD group enable toggle.

Sequence B

1. An LPI is sent to a PE.
2. An SPI or doorbell is sent to the same PE before the interrupt in 1) is activated. This could be because the SPI/doorbell is of a higher priority, or the LPI becomes disabled for any reason.
3. Another SPI or doorbell is pending and enabled that has the same priority as the interrupt in step 2) or the priority of the interrupt in step 2) is `0xF8`.
4. The interrupt in step 2) is recalled for any reason before the interrupt in step 1) is released by the PE. If the interrupt in 1) is activated, this erratum does not occur.

This sequence stalls the software. The stall can be removed by either:

- Activation of the SPI being recalled.
- Any incoming SPI or doorbell of priority higher than the top 3 SPI or LPI doorbells for this PE.
- CPU group enable toggle on the target PE or GICD group enable toggle.

Implications

If software never attempts to recall interrupts with interrupts masked in the CPU (PSTATE.I or priority masking) so that the interrupts being updated are naturally activated, then this erratum causes a performance impact. This is not expected to be significant.
If software attempts to recall interrupts with interrupts masked in the CPU (PSTATE.I or priority masking), then the software might hang while it waits for the operation to complete.

Workaround

There are several possible Workarounds:

1. Control bits

GICD_FCTLR.TPA bit corrects this issue and no further workarounds are required.

However, the TPA bit has the following disadvantages:

- It increases power by continually searching the TGT cache RAMs when there are interrupts pending for a CPU.
- It prevents the Q-Channel from working when there are enabled LPIs (through the Configuration table) in the GIC.
- It can only be turned when all GICR_CTLR.EnableLPI bits are 0, that is, when software initially programs the GIC.

2. Interrupt reprogramming

Software should not attempt to reprogram the GIC (that is, do not perform any operation that may result in recalling an interrupt from a PE) while interrupts are masked by PSTATE.I or priority masking.

Note this implies not using priority `0xF8` or lower after security shift for any interrupt type to ensure the priority mask can always be exceeded.

3. Alternatives

Software should use one of the appropriate workarounds for each situation below:

All causes below assume that software never uses priority `0xF8` or lower after security shift for any interrupt type to ensure the priority mask can always be exceeded.

For ITS commands that have reached a timeout
a) clearing the CPU NSG1 group enable if the command will be targeting the same PE and then restoring the original value once unblocked.
b) clearing the GICD_CTLR.NSG1Enable if the command is targeting a different or unknown PE and then restoring the original value once unblocked.

Note clearing GICD_CTLR.NSG1Enable may itself hit this errata, so RWP should not be polled as part of the workaround before re-enabling.

Note for virtual commands that cause doorbell recalls, the workaround needs to be applied after the return of the first interrupt in each scenario. This should occur in a timely manner and have completed well before the timeout has expired. If this is not the case, then setting and repeating the clear after a subsequent timeout should resolve.

For residency changes that have reached a timeout
a) toggling the CPU NSGI group enable and re-enable the original value once unblocked.

For GICD_ICENABLERn writes that have reached a timeout
a) use one of the workarounds for ITS commands.
b) repeat the GICD_ICENABLER write until RWP drops.

For GICD group enable changes:
Other than for specified workarounds, do not disable GICD Group enables while the corresponding CPU Group enables are on.

2169019
Doorbell missed due to an ITS command

Status

Affects: GIC-700
Fault Type: Programmer Cat B
Fault Status: Present in: r0p0, r0p1, r1p0 Fixed in: r2p0

Description

Doorbells might be missed if an ITS command or GICR INV*R register access removes the only source of a doorbell for a vPE and a new vLPI or vSGI arrives for the same vPE while the GIC services the command.

Configurations impacted

Any configuration with GICv4.1 support.

Conditions

This erratum can occur if the following sequence occurs:

1. There is a non-resident vPE with an enabled doorbell and Group 1 is enabled.
2. An interrupt arrives for the vPE which generates the doorbell.
3. Before the doorbell is serviced, an ITS (or INV register command) removes the interrupt.
4. Before the command completes, another interrupt arrives that would have generated a doorbell for the same vPE.
   Note, the LPI command does not complete until a subsequent **`SYNC`** command completes.

Note, **`INV`** commands that enable interrupts cannot cause this issue.

Implications

Doorbells are not generated for the vPE until a doorbell is generated for any other vPE in the system.

Workaround

There are two possible workarounds. If software uses a command that could remove an interrupt from a non-resident vPE, then either:

- Add an **`INVDB`** command after the LPI command or sequence of commands or replace **`SYNC`** with an **`INVDB`**. The **`INVDB`** regenerates the doorbell if necessary.

- Schedule the vPE to be run without the doorbell.

If using the GICR invalidate registers to mask LPIs while resident, no workaround is necessary.

2141155
RWP can fail to drop after clearing all 3 GICD_CTLR Group enable signals

Status

Affects: GIC-700
Fault Type: Programmer Cat B
Fault Status: Present in: r0p0, r0p1, r1p0 Fixed in: r2p0

Description

When software clears all the GICD_CTLR Group enable bits, the GICD_CTLR.RWP bit is meant to remain set to 1 until the new value propagates to all blocks of the GIC.

In multichip configurations, GICD_CTLR.RWP remains set to 1 while all SPIs to return to their own chip. However, under the conditions detailed below, some SPIs might not automatically return to the owning chip and that causes RWP to remain set to 1.

Note that clearing all GICD group enables is only expected during significant power state changes and not during normal operation.

Configurations impacted

Any multichip configuration with **`chip_count`** > 1

Conditions

The erratum might occur if the following conditions occur:

1. An SPI is targeted at a remote chip.
2. The remote chip services the SPI. The interrupt is active and not pending.
3. Software sets all GICD_CTLR group enables to 0.

Implications

Software might remain stuck polling the GICD_CTLR.RWP bit, which will not change to zero.

Workaround

Software implements a timeout when polling the RWP bit. When the timeout occurs, software can either:

- Issue Secure reads to GICD_ISACTIVERn for all SPIs.

- Toggle and restore GICR_CTLR.DPGx of CPU0 on each chip. Alternatively, Non-secure software can use the GICR_CTLR.DPG1NS bit to perform this procedure.

1978030
Deadlock caused by VMOVP commands forming a loop between chips

Status

Affects: GIC-700
Fault Type: Programmer Cat B
Fault Status: Present in: r0p0, r0p1 Fixed in: r1p0

Description

The ITS has an architected **VMOVP** command that moves a vPE from one PE to another.
GIC-700 uses a mode where it is only necessary to run a **VMOVP** command on a single ITS (GITS_TYPER.VMOVP = 1). Therefore, it is possible to run multiple **VMOVPs** at the same time.
The GIC can deadlock the system if software sets up **VMOVP** commands that form a loop between chips (even though each **VMOVP** operates on a different vPE).

Configurations impacted

Any multi-chip configuration with GICv4.1 support.

Conditions

This erratum can occur if the following conditions occur at the same time:

- Multiple **VMOVP** commands to multiple chips, where each chip has one vPE moving towards it and one vPE moving away from it. For example, **VMOVP** vPE0 from chip 0 to chip 1 and **VMOVP** vPE1 from chip 1 to chip 0.
  Note. The vPE must also have just moved onto the chip within a short (maximum of 16 cross-chip messages) but unobservable time-window.
- Each chip has sent enough vLPI or vSGI targting vPEs that are being moved away from other chips, so that all its cross-chip credits are in use.
- Each chip has a vLPI or vSGI targeting the vPE being moved away from it, that was present before the vPE moved to the other chip.

Implications

The system deadlocks because the **VMOVP** commands cannot complete and eventually acceptance of (v)LPIs ceases.

Workaround

The software should use one or more global semaphores, to ensure that no chip can have a **VMOVP** moving a vPE to it, at the same time as a vPE moves from it.
The simplest workaround, is to limit the system to running a single cross-chip **VMOVP** at a time. A **VMOVP** completes when a subsequent **SYNC** command completes on the same ITS.
If GITS_TYPER.SVPET = 2, then a cross-chip **VMOVP** is any **VMOVP** between two targets with different Affinity2 values.
If GITS_TYPER.SVPET = 1, then a cross-chip **VMOVP** is any **VMOVP** between two targets with different Affinity3 values.

1882328
LPI commands running in parallel with a matching INV might not take effect

Status

Affects: GIC-700
Fault Type: Programmer Cat B
Fault Status: Present in: r0p0 Fixed in: r0p1

Description

Any command that impacts interrupts in the Redistributor might not be applied correctly, if an `INV` for that same interrupt is still running.
Note that `SYNC` does not ensure a previous `INV` has completed.

Configurations impacted

GIC configurations that include LPI support, that is, when the `lpi_support` configuration parameter is set to 1.

Conditions

This erratum occurs when all the following conditions occur:

- An virtual or physical LPI is pending in the GIC
- An `INV` command is issued for the LPI
- An non-`INV` command such as MOV, CLR, DISCARD, or a flush, is issued for the same LPI before the LPI has finished executing.

Note a `SYNC` command following the `INV` does not guarantee that execution has completed.

Implications

Any command running in an indeterminate time window after a matching `INV` command, might not be applied to existing interrupts.
Commands are applied correctly to all new incoming interrupts.

Workaround

Firmware should set the GICD_FCTLR.MCD control bit. Setting MCD to 1, works around the problem and gives the same command performance as the GIC-600.

1815972
Interrupt missed by search due to matching INV or INVALL

Status

Affects: GIC-700
Fault Type: Programmer Cat B
Fault Status: Present in: r0p0 Fixed in: r0p1

Description

The GIC might fail to deliver a virtual or physical LPI that arrives during an `INV` or `INVALL` command that matches it.

Configurations impacted

GIC configurations that include LPI support, that is, when the `lpi_support` configuration parameter is set to 1.

Conditions

The conditions are:

1. An `INV` or `INVALL` command occurs that leaves an interrupt enabled.
2. A matching vLPI or LPI arrives while the GIC processes the command.
3. The vLPI or LPI was not already pending in the GIC before the command arrived.

There are multiple conditions involving cache fill levels, interrupt load, and system memory timings that impact the likelihood of hitting this erratum. However, they are not controllable in the GIC or from the system level.

Implications

If this issue occurs, then the interrupt might not be delivered in a finite time. This issue also includes future LPIs or the same ID that hit and merge with the missed interrupt in the LPI cache.
If the vLPI is the only interrupt for a vPE, then a default Doorbell is not generated even when programmed to do so.

Workaround

The following workarounds are only required if the LPI is enabled after the command. The workaround involves replacing `INV` and `INVALL` commands with alternative commands.
Software must:

- Not use the GICR_INVLPI and GICR_INVALL registers.
- Not use **INVALL** commands after mappings are programmed.

Software must then perform one of the following options:

Option 1: If changing properties is a rare occurrence, then a **CLEAR**/**INT** workaround can be applied as follows:

1. **CLEAR**
2. **SYNC**
3. **INT**

This workaround might result in a spurious interrupt when enabling the LPI.
If using enable as an active mask for the interrupt, then after the **SYNC** command software must check the source instead of regenerating the interrupt.

Option 2: Re-trigger the search by temporarily moving the interrupt to a different source as follows:

1. **INV**. This step is only necessary for physical LPIs because **MOVI** has an implicit invalidation for vLPIs.
2. **MOVI A–B**
3. **MOVI B–A**
4. **SYNC**

During this workaround, the intermediate target (B) might service the interrupt. Therefore, Arm recommends that:

- For vLPIs, the intermediate target (B) is a dummy vPE per VM, with no Doorbell enabled.
- For physical LPIs, software is tolerant of the interrupt arriving at the intermediate target (B).

Category B (rare)

1985213
Issues caused by repeated VMOVP commands of the same vPE in systems with 3 or more chips

Status

Affects: GIC-700
Fault Type: Programmer Cat B(rare)
Fault Status: Present in: r0p0, r0p1 Fixed in: r1p0

Description

The ITS has an architected **`VMOVP`** command that moves a vPE from one PE to another.
The GIC-700 is designed to use a mode where it is only necessary to run a **`VMOVP`** command on a single ITS (GITS_TYPER.VMOVP = 1) to successfully move a vPE.
However, if a second **`VMOVP`** for the same vPE is issued before the first **`VMOVP`** propagates around the system, then the following issues can occur:

- Deadlock of the ITS command queue
- vLPI or vSGI interrupt loss on the moved vPE.

Configurations impacted

Any multi-chip configuration with GICv4.1 support and 3 or more chips.
Note, it is not believed that any matching configurations exist using the affected versions. Please contact Arm Support if this is not the case.

Conditions

This erratum can occur if the following sequence occurs:

1. A **`VMOVP`** is run on a vPE to move a vPE between chips. A **`SYNC`** on the issuing ITS can also have occurred.
2. A virtual command/vLPI or vSGI arrives on a third chip for the same vPE.
3. A second **`VMOVP`** occurs, to move the vPE to a third chip before the first **`VMOVP`** fully propagates and is accepted by the third chip.

Implications

The system might deadlock or it might lose the vLPI or vSGI.

Workaround

Note, this workaround is not lightweight and it has a significant performance impact.
Software should ensure that:

- Only one virtual command per vPE can be run at a time across all ITS in the system. That is, ensure that **`SYNC`** commands are completed between commands to the same vPE.
- Before software moves a vPE onto a chip, it should perform the following sequence:
  a. Issue a **`VMOVP`** command for the vPE to move it onto its current location from an ITS on the destination chip.
  b. Use GICR_VINVCHIPR on the destination chip to invalidate all chip information for all chips other than the destination chip.
  c. Issue the **`VMOVP`** from any ITS where the vPE is mapped in the system.

Category C

2381076
RSS Residency Completion TD event lost for coincident events

Status

Affects: GIC-700
Fault Type: Programmer Cat C
Fault Status: Present in: r0p0, r0p1,r1p0,r2p0 Fixed in: r2p1

Description

The PMU has the option to count residency changes, however, it will only count one if multiple changes complete in the same cycle.

Conditions

Multiple Residency changes complete at the same time (e.g. after a cache search completes)

Configurations Impacted

Any with gicv41_support = 1.

Implications

The PMU will undercount.

Workaround

None.

2381241
RAS Record 0 can incorrectly report overflow due to masked Deactivates

Status

Affects: GIC-700
Fault Type: Programmer Cat C
Fault Status: Present in: r1p0,r2p0 Fixed in: r2p1

Description

The GIC can report out-of-range Deactivates on RAS record 0.
Reporting of these messages can be disabled using GICT_ERR0CTLR.DIS_DEACT.

If two of these events occur at the same when the error is masked, then the overflow bit is not correctly masked and it causes an overflow on a future error.

Conditions

An incorrect overflow is reported if the following sequence occurs:

1. Out-of-range Deactivate errors are masked.
2. Two CPUs perform out-of-range deactivates at exactly the same time.
3. One of the following errors (all due to bad Software) occurs when not masked:
    - `0x18` - SYN_SPI_BLOCK
    - `0x19` - SYN_SPI_OOR
    - `0x1B` - SYN_SPI_NO_DEST_1OFN

Configurations impacted

All.

Implications

Software is not able to identify if a real overflow has occurred.

Note that only the 3 syndromes mentioned above can inherit an incorrect overflow.

Non-secure software can create Deactivate events, so it could maliciously attempt to hit this erratum.

Workaround

None.

2289578
Read error reporting does not include rpoison in vTGT and PTS

Status

Affects: GIC-700
Fault Type: Programmer Cat C
Fault Status: Present in: r0p0, r0p1, r1p0,r2p0 Fixed in: r2p1

Description

If the GIC reads data from any of the following tables and the system returns data marked as poisoned then the data is treated as errored but not reported in the RAS registers.

1) Pending Tables (virtual or Physical)
2) LPI property tables when read for Doorbells (v4.1) or in blocks of 32 interrupts.

Conditions

The GIC cannot poison data itself so this can only happen if data is poisoned elsewhere in the memory system.

Configurations Impacted

Any configuration with lpi support

Implications

If Poisoned data is returned by the system:
1) In v4 systems doorbells will not be delivered if the property fetch returns poisoned data and this won't be reported.
2) Interrupts will not be delivered from the PT and this won't be reported.

Workaround

If possible react to the data poisoning if reported by the memory system.
ARPOISON can be masked using GICD_FCTLR2.ARP. This will cause the GIC to attempt to continue with the supplied data.

2185600
Doorbell may not be masked by Activate resulting in duplicate doorbell

Status

Affects: GIC-700
Fault Type: Programmer Cat C
Fault Status: Present in: r0p0, r0p1, r1p0 Fixed in: r2p0

Description

A vPE might receive two copies of the doorbell rather than the required one. This extra doorbell is likely (but not guaranteed) to be in close temporal proximity to the first, and could arrive after the vPE is made resident.
This spurious doorbell is not recalled by an `INVDB` command, residency change, or the removal of the pending virtual interrupts.
Changing the doorbell ID is not an expected usecase, however, if it is changed, or the vPE is unmapped and reallocated before the spurious doorbell is activated, then a doorbell with the new doorbell ID might not be generated.

Configurations impacted

Any configuration with GICv4.1 support.

Conditions

A doorbell is required for a vPE and one of the following occurs:

- Another interrupt arrives for the vPE
- The GIC is heavily loaded such that there are more than 3 SPIs or Doorbells pending, enabled and routed to any PE
- An `INVDB` occurs on any vPE
- A DED error occurs in the vTGT RAM

Implications

If the doorbell ID is not changed, then the occasional spurious doorbell only incurs a minor performance impact.
If the doorbell ID is changed or the vPE ID is reallocated faster than the doorbell is serviced, then it is possible that the old doorbell ID is used until it is activated. This behavior means that directly injected interrupts do not schedule the vPE, and it is not scheduled until some other event occurs such as a virtual timer or an SPI occurs.

Workaround

No workaround is expected to be required for the spurious doorbells because software should simply schedule the vPE on both doorbells. Note, software must still schedule the vPE if doorbells arrive during the non-residency handshake.

If software changes the doorbell on a vPEID, then it could use the following sequence before the next residency:

1. Generated an enabled vSGI to the vPE.
2. Issue a **`VMOVP`** with DB=1, to clear the doorbell mask.
3. Issue an **`INVDB`**.

2181357
Interrupt may miss search after command on non-resident vPE

Status

Affects: GIC-700
Fault Type: Programmer Cat C
Fault Status: Present in: r0p0, r0p1, r1p0 Fixed in: r2p0

Description

Interrupts for a vPE stored in memory might not be delivered when a vPE is made resident, if a command removes a vLPI from that vPE in the previous non-residency period.

Configurations impacted

Any configuration with GICv4.1 support.

Conditions

The following conditions must all be met:

1. A vLPI is being evicted from the LPI cache to the Pending table (main memory). This eviction could be due to the cache overflowing or to clear a space for higher priority interrupts.
2. The vPE is not resident
3. There are 3 vLPIs waiting for the vPE
4. An LPI command removes one of those 3 vLPIs

Implications

The expected interrupts that are stored in memory might not be delivered. There is no impact on doorbell generation.

Workaround

The issue cannot occur if the virtual CPU Group 1 enable is on (ICV_GROUPR1_EL1) before the vPE is made resident using the GICR_VPENDBASER register. This behavior is the expected use case.

Otherwise, software can resolve the issue by issuing an invalidate command (`INV`) for any in-range vLPI ID on the same vPE, either through an ITS command queue or the GICR_INVLPIR register. This can be done at any point after the vPE is made resident.

2121183
ITS_DID_MISS and ITS_COL_MISS PMU events are pessimistic

Status

Affects: GIC-700
Fault Type: Programmer Cat C
Fault Status: Present in: r0p0, r0p1, r1p0 Fixed in: r2p0

Description

The ITS_DID_MISS event will incorrectly count when an ITS translation hits in the VCACHE.
The ITS_COL_MISS event will incorrectly count when an ITS translation is virtual or hits a locked cache entry.

Configurations impacted

Any configuration with `lpi_support` == 1.

Conditions

The erratum occurs when the PMU is configured to count DCACHE or CCACHE misses.

Implications

The PMU events indicate significantly poorer cache usage than is actually happening.

Workaround

The ITS_DID_MISS value can be corrected by using multiple counters and the following formula:

- Device cache misses = ITS_DID_MISS − (ITS_LPI − ITS_VID_MISS)

The ITS_COL_MISS value can be corrected by using multiple counters and the following formula:

- Collection cache misses = ITS_COL_MISS − ITS_LPI(with a virtual filter)

If using locked entries, then also subtract ITS_LL_LPI.

2032913
Deadlock caused by response to wrong chip during cross-chip MOVALL with 3 or more chips

Status

Affects: GIC-700
Fault Type: Programmer Cat C
Fault Status: Present in: r0p0, r0p1, r1p0 Fixed in: r2p0

Description

If multiple **`MOVALL`** commands target the same chip at the same time, the GIC might send a response to the wrong chip for part of a **`MOVALL`** command.

Configurations impacted

Any configuration with 3 or more chips.

Conditions

This erratum can occur if the following conditions are met:

- Two ITS **`MOVALL`** commands are running on different chips and both commands target a third chip.
- Both PEs being moved have interrupts in the *Pending table* (PT).
- Both PEs attempt to send sections (blocks) of the PT to the third chip at the same time.
- There is significant backpressure of the cross-chip interface, so fewer than 6 messages are output in the time it takes to issue and complete 2 memory reads.

Implications

No known driver uses the **`MOVALL`** command. Instead, drivers use on a sequence of **`MOVI`** commands to correctly balance interrupt load. However, if the conditions above are met, then the GIC might deadlock.

Workaround

If a workaround is required, then software should only allow one **`MOVALL`** command to target a chip at any one time.
Note:

- Only **`MOVALL`** commands between targets on different chips need to be considered. Depending on the configuration, a chip is defined by either the affinity3 or affinity2 value.

- Linux does not use **MOVALL** commands, so no workaround is needed.

1995446
Affinity3 field corrupted by cross-chip 32 bit writes to upper half of GICD_IROUTERn(E)

Status

Affects: GIC-700
Fault Type: Programmer Cat C
Fault Status: Present in: r0p0, r0p1 Fixed in: r1p0

Description

The upper 32 bits of the GICD_IROUTERn(E) registers contain the Affinity3 field. Also, the whole register should be accessible from the GICD address space of any connected chip using 32-bit or 64-bit accesses.

However, using a 32-bit write to the upper half of a GICD_IROUTERn(E) for an SPI that is owned on a different chip, results in the Affinity3 field being set to zero rather than the programmed value.

Configurations impacted

GIC configurations that include multiple chips and an affinity3 width of greater than 0, that is, when:

- The **`chip_count`** configuration parameter is set to 1.
- The **`chip_affinity_select_level`** configuration parameter is set to 3.

Note that AArch32 systems do not support nonzero Affinity3 values, so this erratum only impacts AArch64 systems.

To trigger this erratum, software must have already completed multiple 64-bit accesses, to have successfully connected the chips.

Conditions

Software issues a 32-bit write to the upper half of GICD_IROUTERn(E) for an SPI that is owned on another chip.

Implications

Impacted SPIs are not routed to the expected CPUs. Depending on the system topology and programming, the SPI might be delivered to a CPU on chip 0 with matching Affinity2, Affinity1 and Affinity0, or the SPI might not be delivered.

Workaround

There are two possible workarounds:

- Always use 64-bit writes when setting the Affinity3 field in GICD_IROUTERn(E). 32-bit writes are acceptable when setting GICR_IROUTERn(E).IRM. This workaround is the expected behavior.
- Program GICD_IROUTER(n)E registers through the GICD or GICDA registers space, on the chip where the SPI is owned according to the GICD_CHIPR registers.

1991967
Incorrect debug data on some GICx<n>_ERRINSR registers

Status

Affects: GIC-700
Fault Type: Programmer Cat C
Fault Status: Present in: r0p0, r0p1 Fixed in: r1p0

Description

The GICx<n>_ERRINSR registers are present so that software can check ECC functionality and test software recovery paths.
The read data returned by the following registers is incorrect:

- GICR<n>ERRINSR (CI block RAMs)
- GICD_ERRINSR3 (SPI TGT RAM)
- GICD_ERRINSR4 (LPI TGT RAM)

There is no issue with the write functionality.

Configurations impacted

Configurations with ECC enabled on the corresponding RAM.

Conditions

Software attempts to read any of the registers that *Description* lists.

Implications

The registers can still be used for their intended purposes but the read data is invalid.

Workaround

Software should ignore the read data. Software can check the related RAS error records directly, for details of the inserted errors.

1981453
PMU filtering incorrect between virtual and physical interrupts

Status

Affects: GIC-700
Fault Type: Programmer Cat C
Fault Status: Present in: r0p0, r0p1 Fixed in: r1p0

Description

Target filters do not work correctly for some PMU events when trying to differentiate between virtual and physical interrupts.

Configurations impacted

All configurations, but more PMU events are impacted on configurations with LPI support.

Conditions

Software applies target filters to any of the following PMU events:

- Any LPS event
- SET_COMP
- VSET_COMP

Implications

The PMU count value is inaccurate and should be ignored.

Workaround

Do not apply target filters to those PMU events that the *Conditions* section describes.

1980415
vPE corruption due to vPT read error during VMOVI

Status

Affects: GIC-700
Fault Type: Programmer Cat C
Fault Status: Present in: r0p0, r0p1 Fixed in: r1p0

Description

Architecturally, the GIC expects to have free flowing access to the *virtual Pending Tables* (vPT) that are allocated to it using the ITS **`VMAPP`** commands.
During a **`VMOVI`** command, if a read error occurs when accessing the vPT of the destination vPE, then a different vPE might be corrupted if certain conditions are met. See *Conditions* for more information.

Configurations impacted

GIC configurations that include multiple chips, that is, when the **`chip_count`** configuration parameter is set to more than 1.

Conditions

This erratum can occur if all the following conditions are met at the same time:

- An ITS issues a **`VMOVI`** command.
- The source vPE of the **`VMOVI`** is mapped on a different chip.
- The destination vPE is not mapped to the same chip as the source vPE.
- The destination vPE is not mapped by any ITS on the chip where the source vPE is currently mapped.
- The GIC receives a read error or poison indication, when attempting to access the vPT of the destination vPE.
- The vPE configuration table pointed to by GICR_vPROPBASER is indirect, that is, when GICR_vPROPBASER.Indirect =1.

Implications

When the GIC updates the vPE Configuration table entry for the destination vPE, it uses an incorrect (random) first-level table entry for the vPE Configuration table. This behavior results in the corruption of a vPE that the **`VMOVI`** command did not target.

Workaround

Software should ensure that any tables that are allocated to the GIC are valid and always accessible, so that response errors do not occur.
However, either of the following options can also resolve the issue:

- Ensure that all vPEs are mapped on all chips. This mapping is expected given vSGI support.
- Use a single-level vPE Configuration table in GICR_vPROPBASER instead of indirect tables. This option is not recommended unless there are less than 512 vPEs.

1968775
Unpredictable SW access to GICR_INVLPIR or GICR_INVALLR after VTGT DED error can cause vPE corruption

Status

Affects: GIC-700
Fault Type: Programmer Cat C
Fault Status: Present in: r0p0, r0p1 Fixed in: r1p0

Description

If software attempts to invalidate vLPIs on a vPE using GICR_INVLPIR or GICR_INVALLR when the vPE is mapped to a different chip (affinity group as defined by GITS_TYPER.SVPET) and there has been a DED (uncorrectable) error in the VTGT cache, then the vPE might become corrupted.

Configurations impacted

Any configuration with GICv4.1 support and multi-chip support.

Conditions

The erratum can occur when the following occur concurrently:

1. An uncorrectable error in the VTGT cache and the automatic internal recovery sequence has not completed.
2. Software attempts to invalidate LPIs on a vPE using GICR_INVLPIR or GICR_INVALLR, when the vPE is not mapped on the chip (GITS_TYPER.SVPET group) being accessed.

Implications

The vPE that is targeted by a GICR_INVLPIR or GICR_INVALLR write, might be corrupted and interrupts lost.

Workaround

Software must only use GICR_INVLPIR or GICR_INVALLR on the GICR where the vPE is currently mapped.
Note, this behavior is the expectation of the architecture and software cannot rely on the GICR_INVLPIR or GICR_INVALLR registers of Redistributors (GICR) working, if the vPE is not currently mapped to the same Redistributor.

1957948 Target range check for MAPC/MOVALL/VMAPP/VMOVP ignores bits[51:48] of RDbase field

Status

Affects: GIC-700
Fault Type: Programmer Cat C
Fault Status: Present in: r0p0, r0p1 Fixed in: r1p0

Description

The ITS ignores bits[51:48] of the RDbase (target) field of **`MAPC`**/**`MOVALL`**/**`VMAPP`**/**`VMOVP`** commands.

Configurations impacted

Any configuration with LPI support, that is, the **`lpi_support`** configuration parameter is set to 1.

Conditions

Target field [51:48] of the RDbase field are nonzero during a **`MAPC`**, **`MOVALL`**, **`VMAPP`**, or **`VMOVP`** command.

Implications

A command issued with an out-of-range target might not be detected, and it might execute with a truncated TGT field.

Workaround

Software should only use legal targets that only use the lower bits of the RDbase field.

1957262
Reported error data is corrupted for PPI RAM (error records 7&8)

Status

Affects: GIC-700
Fault Type: Programmer Cat C
Fault Status: Present in: r0p0, r0p1 Fixed in: r1p0

Description

If a SEC (correctable) or DED (uncorrectable) error occurs in the PPI RAM in a GCI block, then the error is correctly recorded but the supplied data in GICT_ERR<n>MISC0 is incorrect.

Configurations impacted

Any configuration with **`ppi_ram_ecc`** = 1.

Conditions

A correctable or uncorrectable error occurs in any PPI RAM in a GCI block.

Implications

The supplied data is wrong and it must be interpreted as *Workaround* describes. The only lost information is the bit_location of a SEC error, which should not cause issues in operating or recovering from errors in the GIC.

Workaround

Software must interpret the GICT_ERR<n>MISC0.Data field as follows:

- PPI block - use as the R0 *Technical Reference Manual* (TRM) specifies.
- Bit_location - this field is corrupt so software must ignore it.
- Offset - this field is corrupt. For DED errors, software can read GICR_ICERRR0 and GICR_ICERRR1E to determine this field.
- SGI/Int - this bit is inverted as defined in the R0 TRM.
- Core - use bits[6:1] to determine the impacted PE.

1912069
Unrecoverable DED error in TGT or VTGT may result in interrupts being left in PT

Status

Affects: GIC-700
Fault Type: Programmer Cat C
Fault Status: Present in: r0p0, r0p1 Fixed in: r1p0

Description

If a DED error occurs in the TGT(LPI) or VTGT cache when the *Pending Table* (PT) is in use for a particular PE, then interrupts in the PT for the vPE or PE might not be delivered until one of the following events occurs:

- an **`INVALL`** for the PE
- a queue of three interrupts is found for the PE or vPE. These interrupts could be either new or in the LPI cache from before the error.

Configurations impacted

Configurations that support LPIs.

Conditions

This erratum can occur when:

- there are 3 (v)LPIs pending for a (v)PE
- there are less than 3 (v)LPIs for the same (v)PE in the LPI cache
- there are interrupts in the PTs for the same v(PE)
- a DED error occurs in the TGT or vTGT cache
- interrupt processing rate suddenly decreases so that there are never 3 pending vLPIs for the vPE again.

Implications

The DED error might result in the loss of (v)LPIs from the system and so Arm expects that the most likely system scenario is to reset.
However, if software attempts to continue operation despite the loss of interrupts, then under the specified conditions some more (v)LPIs for the same (v)PE might not be delivered.

Workaround

If software does not perform a reset, to recover from the DED error, then it must either:

- perform an **`INVALL`** on the specified (v)PE
- follow the procedure to flush the LPI caches by using GICR_WAKER.Sleep and then wake up the GIC.

1900358
ClockGating and Q-Channel are too pessimistic

Status

Affects: GIC-700
Fault Type: Programmer Cat C
Fault Status: Present in: r0p0 Fixed in: r0p1

Description

The GIC can keep its hierarchical clock gates running more than is necessary including when all CPUs are asleep.
This in turn leads to unnecessary QDENY responses and a refusal to enter low_power mode on the GICD Q-Channel.

Configurations impacted

All.

Configurations without LPI support, that is, when the `lpi_support` configuration parameter is set to 0 are not impacted when all CPUs are asleep.

Conditions

This erratum occurs when any of the following are true:

- SPIs which must be targeted at a CPU with the relevant CPU group enabled
- vLPIs for each resident vPE irrespective of the CPU group enables and sleep state of the CPU.
- vSGIs for each resident vPE irrespective of the CPU group enables and sleep state of the CPU.
- LPIs including Doorbells irrespective of the CPU group enables and sleep state of the CPU.

Implications

The GIC will reject Q-Channel low_power requests more often than it should, resulting in poorer clock gating and higher power consumption.

The GIC will reject low_power Q-Channel requests with QDENY responses if any of the conditions are met.

Workaround

There is no work around to the extra denies during normal operation.

If the conditions are met when attempting to enter low power mode when CPUs are asleep, then the GIC will raise the necessary wake_requests to the power controller. Software should respond to these and ensure the GIC is empty before reattempting the power down request again.

1892861
Errored RAM data not reported for vLPIs

Status

Affects: GIC-700
Fault Type: Programmer Cat C
Fault Status: Present in: r0p0, r0p1 Fixed in: r1p0

Description

For uncorrectable (DED) errors, the error records of the vLPI RAMs do not report the corrupted data in the GICT_ERR<n>MISC1 registers.

Configurations impacted

Any configuration with GICv4.1 support.

Conditions

This erratum occurs when a DED error occurs in the VRES or VSTR RAMs.

Implications

For a DED error, any vLPI information stored in these RAMs is lost and the expected use case is to restart the device.
The reported data was intended to allow software to attempt to derive the lost IDs.

The error counts and addresses are correctly reported.

Workaround

There is no workaround.

1886083
Reset issue can cause read to 0x0 during coincident GITS_SGIR write and VMAPP(V1A0) command to an offline chip

Status

Affects: GIC-700
Fault Type: Programmer Cat C
Fault Status: Present in: r0p0, r0p1 Fixed in: r1p0

Description

There is a non-reset flop in the Distributor which could cause the GIC to issue a read to address `0x0` if software attempts a `VMAPP(V1A0)` command to an offline-chip.

Software is never expected to issue commands that target an offline chip.

Configurations impacted

Any multi-chip configuration with GICv4.1 support.

Conditions

This erratum occurs when the following events occur at the same time:

- A `VMAPP(V1A0)` command targeting an offline chip that is the first command sent to a Redistributor from an ITS on the same chip after reset.
- A vSGI write that is targeting the same vPE as the `VMAPP`

Implications

The GIC reads from an illegal region, resulting in unpredictable behavior that depends on the returned data.

Workaround

There are 2 possible workarounds:

- Software must not issue commands to offline chips or
- Software should issue any non-MAP virtual ITS command such as `VINVALL`, straight after turning the ITS on.

1871535
GICT_ERR0ADDR.PADDR has non-standard 32-bit write behavior

Status

Affects: GIC-700
Fault Type: Programmer Cat C
Fault Status: Present in: r0p0, r0p1 Fixed in: r1p0

Description

ADDR field of GICT_ERR0ADDR.PADDR does not update correctly with 32-bit register writes.

Configurations impacted

All configurations with an AXI slave interface address width greater than 32 bits, that is, when the **`axis_addr_width`** configuration parameter is set to greater than 32.

Conditions

This erratum occurs when software uses 32-bit access to update GICT_ERR0ADDR.PADDR

Implications

The PADDR field of GICT_ERR0ADDR is in the bottom **`axis_addr_width`** bits of the register.
If the register is written with a 32-bit access to the lower half of the register, then any bits of PADDR above bit[31] are cleared to 0.
If the register is written with a 32-bit access to the upper half of the register, then none of the PADDR[51:32] field is updated.

Workaround

There is no known reason to write to GICT_ERR0ADDR.PADDR. However, if software wants to write to the upper 32 bits of the register (other than to clear to 0), then it must use a 64-bit write access.

1859709
Multiple concurrent ECC errors can cause LPI loss during GICD_ERRINSR debug register write

Status

Affects: GIC-700
Fault Type: Programmer Cat C
Fault Status: Present in: r0p0, r0p1 Fixed in: r1p0

Description

If multiple ECC errors occur during a GICD_ERRINSRn debug register write for the PT Map Cache then the pending table of a processor or vPE in the map cache might become corrupted, resulting in loss of interrupts on that processor or vPE.

Configurations impacted

GIC configurations that include both:

1. LPI support, that is, when the **`lpi_support`** configuration parameter is set to 1.
2. PT RAM ECC, that is when the **`pts_ram_ecc`** configuration parameter is set to 1.

Conditions

This erratum occurs when all the following conditions occur within 7 clock cycles:

- An uncorrectable error (DED) is detected on line B of the pending table map cache.
- Software uses a GICD_ERRINSRn register, to attempt to inject an error into line A of the pending table map cache, where B = A +1.
- A correctable error (SEC) is detected on line A of the pending table map cache.

Implications

The ECC errors are detected and reported but the recovery process might corrupt the pending table of the processor or vPE in the cache, resulting in a loss of LPIs or vLPIs on that target.

Workaround

Do not use the GICD_ERRINSRn registers for the pending table map cache during runtime, unless software intends to reset the system on detection of a DED error.

1806339
LPI Merge PMU incorrectly also counts LPI command matches

Status

Affects: GIC-700
Fault Type: Programmer Cat C
Fault Status: Present in: r0p0, r0p1 Fixed in: r1p0

Description

The LPI Merge event is meant to count when LPIs are recieved that already match LPIs pending in the LPI cache.
It also incorrectly counts when LPI commands match an interrupt.

Configurations impacted

GIC configurations that include LPI support, that is, when the `lpi_support` configuration parameter set to 1.

Conditions

This erratum occurs when all the following conditions occur:

- The PMU is being used to count LPI Merge events
- LPI commands are run that match the same interrupts

Implications

If LPI commands match the PMU filter then the PMU overcounts.

Workaround

Only use the LPI MERGE PMU event when not also running LPI commands that match the PMU filters.

1801287
ECC error can cause incorrect data to be logged for GICR_ERRINSR debug register access

Status

Affects: GIC-700
Fault Type: Programmer Cat C
Fault Status: Present in: r0p0, r0p1 Fixed in: r1p0

Description

If a real error occurs in the GCI RAM a few cycles before the GIC processes a debug write to GICR_ERRINSR, then the GICR_ERRINSR might report the error data for the previous error in the GICT error records.

Configurations impacted

GIC configurations that include GCI RAM, that is, when the **`ppi_ram_ecc`** configuration parameter is set to 1.

Conditions

This erratum occurs when the following conditions occur in a very short time window:

1. An ECC error in the GCI RAM
2. Software cleares the error record for the GCI RAM error.
3. A debug register access to GICR_ERRINSR is processed on the same RAM.

Implications

The error record reports the correct number of errors but the reported error and location might be incorrect.

Workaround

Arm expects that software is tolerant of unexpected data being reported for GIC<x>ERRINSR register writes, and that it is able to clear the error record and retry the error insertion because there is always the possibility that another error could occur.

Error recovery should not be affected because only inserted errors are reported incorrectly. Therefore, no extra software workaround is expected.

1733884
DED error in VICM RAM can cause data corruption after aborted flush

Status

Affects: GIC-700
Fault Type: Programmer Cat C
Fault Status: Present in: r0p0, r0p1 Fixed in: r1p0

Description

If the VICM receives a memory flush request that is removed at the end of an outstanding load, then any vPE that has un-recovered DED errors is not loaded correctly from memory.

Configurations impacted

Any configuration with GICv4.1 support.

Conditions

This erratum occurs when all the following events occur in order:

- A DED error has occured in the VICM RAM
- The VICM is loading in vPE info as a result of a GICR_VPROPBASER.Valid = 1 or GITS_CTLR.Enabled = 1 write
- The VICM is told to flush everything by setting all GICR_VPROPBASER.Valid = 0 and GITS_CTLR.Enabled = 0.
- The VICM is told to load vPE info by writing any GICR_VPROPBASER.Valid = 1 or GITS_CTLR.Enabled = 1 without waiting for completion of the flush.

Implications

Errored vPE might be corrupted. The VpeID which had a DED error is reported in the GICT registers.

Workaround

Do not abort flush requests. Always poll for completion of the flush before re-enabling.
The expected use case is:

- If writing GICR_VPROPBASER.Valid = 0, then poll GICR_CTLR.RWP
- If writing GITS_CTLR.Enabled = 0, then poll GITS_CTLR.Quiescent