Level-2 Cache Issues H1 2023

Technical Note

2023-07-16

Doc. No GRLIB-TN-0021

Issue 1.2

CHANGE RECORD

Issue	Date	Section / Page	Description
Draft 1.0	2023-07-03	All	Draft with limited distribution
1.0	2023-07-03	1.1 and 1.2 2 4 9 11	Correction of cross references to section 10. Editorial changes. Clarify Expanded Summary Renamed issue 2 Renamed issue 7 Updated
1.1	2023-07-04	10.3	Remove statement about draft status
1.2	2023-07-16	7, 10.1, 10.3 10 9.3	Updated workaround for issue 5. Remove the work TENTATIVE from heading Expanded workaround

TABLE OF CONTENTS

1 OVERVIEW

1.1 Scope

This document describes seven issues that affect GRLIB Level-2 cache controller (L2C) IP core. The Level-2 cache is also included in the GR740 SoC.

This document assumes that users have access to the L2C documentation. For IP core users, this documentation is available in the GRLIB IP Core User's Manual (GRIP) included in the GRLIB IP Library. For GR740 users, this documentation is available in the GR740 Data Sheet and User's Manual, the latest version is available from https://www.gaisler.com/GR740

1.2 Affected versions

At the time of writing there is no GRLIB release that corrects all issues in the L2C and all GRLIB releases with the L2C IP core are affected. An update of this document will be released once a GRLIB release with corrections is available.

1.3 Affected components

The following Frontgrade components are affected by the errata: GR740.

1.4 Distribution

Users of affected products are free to use the material in this document in their own documents and to redistribute this document. Please contact us for inquires on other distribution.

The latest version of this document is available from https://www.gaisler.com/notes

1.5 Contact

For questions on this document, please contact our support at support@gaisler.com. When requesting support, include the part name if the question is a specific device or the full GRLIB IP library package name if the question relates to a GRLIB IP library license.

2 SUMMARY

To understand the L2C issues and proposed workarounds, it is necessary to be familiar with the different types of accesses that are created internally in L2C because of accesses to the L2C on the on-chip bus. The different types of accesses are described in subsections below.

The descriptions in this document refer to L2C memory mapped registers and fields within these registers. In this case the annotation <*register*>.<field> is used. For example, L2CACCC.DBPWS is the DBPWS field in the L2CACCC (L2C Access Control) register.

Several of the issues are affected by SPLIT responses that are used instead of wait states to allow other entities to make use of the on-chip bus. When use of SPLIT responses is enabled, the L2C may answer to an incoming access with AMBA SPLIT and proceed to perform miss-processing for the access. Since the L2C can accept additional incoming accesses, the L2C is able to serve cache hits while doing miss-processing in this case. After a master has received a SPLIT response, it is prevented from accessing the bus until the L2C issues a SPLIT complete, also referred to as “un-split”. The L2C use of SPLIT responses is controlled via the L2CACCC.SPLIT register field.

Sections 3 to 9 contain descriptions of the issues. These are followed by section 10 that describes how to combine the workarounds. Section 11 provides information on affected Frontgrade Gaisler software packages.

2.1 Cached memory access

Normal cacheable accesses (in the case of the GR740, to the address range 0x0 - 0x7FFFFFFF) are not affected by these L2-cache issues.

2.2 Uncached memory access

A cache access (in the GR740, to the address range 0x0 - 0x7FFFFFFF) is uncached in the cases listed below. This is from the perspective of the L2C, processor MMU page cacheability does not matter.

- An access to an address range defined uncached by the MTRR registers.
- An access defined uncachable by the AMBA HPROT bus signal. (In the GR740, the only way to make HPROT signal an uncachable access is to configure this functionality in the IOMMU for fetching protection data from main memory. The behaviour is controlled via the HPROT field in the IOMMU control register).
- An access when the L2C is disabled.
- An access to an L2C memory address range outside of the locked ways, when the entire cache is locked (used as an on-chip memory)

2.3 IO area access

The IO area is used to create a bridge to map registers and plug&play information behind the cache. In the GR740 this area is mapped to address range 0xFFE00000 - 0xFFEFFFFF and is used for accessing the configuration registers for the memory controller and the memory scrubber.

2.4 L2C register access

The L2C registers are the memory-mapped control and status registers for the L2C. These are mapped to address range 0xF0000000 - 0xF03FFFFF in the GR740 component.

3 ISSUE 1: CACHE FLUSH CAN CAUSE REORDERING OF WRITE ACCESSES

3.1 Technical description

Flushing cache lines in the L2C via the L2C register interface can cause write accesses, directly followed the register access, to be reordered.

The following condition is required to trigger the issue.

- L2C is busy handling a cache miss.
- Register access to flush (write-back) a cache line (or the entire cache). The flush will be queued due to the ongoing cache miss handling.
- Memory writes (generating cache misses) are issued. These accesses will be queued after the flush operation.

3.2 Functional impact

When the flush operation is executed from the internal queue, there is a window for the first write (in the queue) propagate through the L2C pipeline and be readded to the queue. This would remove this write access from the first position in the queue and place it in the last position in the queue and this write will be reordered related to the other accesses present in the internal queue. This only result in stale data when the reordered writes are overlapping in address space.

3.3 Workaround

All write accesses to the L2C flush register need to be done using atomic (read-write) operations.

4 ISSUE 2: A R-W-R OR R-A ACCESS SEQUENCE TO THE L2C REGISTER INTERFACE CAN CAUSE L2C LOCKUP

4.1 Technical description

There are two scenarios where accesses to the L2C register interface can cause a L2C lockup, causing the on-chip bus to lock up.

- Scenario A: An access sequence with two (or more CPUs) where one CPU reads (L2C) register followed by a second CPU writing and then reading same register (or a register within the same 32-byte address block), the L2C can end up in a state locking the bus interface by not completing any access.
- Scenario B: Similar to scenario A, but the second CPU performed an atomic access.

The text below describes the conditions required to be met to trigger the issue.

- Both scenarios:
  - The L2C has been configured to make use of SPLIT responses.
  - The L2C is busy handling a cache miss.
- Scenario A:
  - The feature "wait-states for discarded bypass data" need to be enabled (bit L2CACCC.DBPWS = 0).
  - CPU1: read a L2C register. Access will be queued due to L2C being busy handling a cache miss.
  - CPU2: write to a L2C register (within the same 32-bytes block)
  - CPU2: reads a L2C register.
  - When the read access for CPU1 and CPU2 is reissued (SPLIT complete), the L2C needs to be busy, and the accesses are queued up. This happens for example when CPU2 writes to the flush register and triggers a write-back.
- Scenario B:
  - CPU1: read a L2C register. Access will be queued due to L2C being busy handling a cache miss
  - CPU2: access a L2C register (within the same 32-bytes block) with an atomic (locked) access.

4.2 Functional impact

The read access from CPU1 is un-split and wait-states are inserted on the bus until access completes. This access never complete due to the access sequence performed by CPU2 and the bus will be locked up.

Note that the labels CPU1 and CPU2 should not be read to mean that a specific processor or master in the system needs to perform these accesses. CPU1 can be any bus master and CPU2 can be any other bus master.

4.3 Workaround

For both scenarios:

- Disable SPLIT response OR,
- Limit the number of concurrent accesses to the L2C register interface to one. This can be done by limiting access to the register interface to one CPU with a spinlock.

For scenario A, the following is an additional workaround:

- Disable the feature "wait-states for discarded bypass data" (bit L2CACCC.DBPWS = 1).

5 ISSUE 3: CORRUPTED CACHE LINE BY R-W SEQUENCE FOR UNCACHED MEMORY OR IO AREA ACCESS

5.1 Technical description

A read-write sequence, within the same 32-byte address block, for uncached memory or the IO area could cause data in a cache line to be overwritten.

The following conditions are needed to trigger the issue.

- The L2C has been configured to make use of SPLIT responses.
- CPU1: read of an uncached memory address or the IO area.
- CPU2: write to the same 32-byte address block that is currently being fetched for memory (or IO area) due to a previous uncached read access.

Note that the labels CPU1 and CPU2 should not be read to mean that a specific processor or master in the system needs to perform these accesses. CPU1 can be any bus master and CPU2 can be any other bus master.

5.2 Functional impact

Corrupt data in the cache line matching the read access (same line which would have been replaced if the read access was cachable).

5.3 Workaround

- Disable SPLIT response OR,
- Prevent a write access to the same 32-byte block of uncached memory (or IO area) which is currently being fetched due to a previous read access.

6 ISSUE 4: L2C LOCKUP AFTER MATCH IN ERROR STATUS REGISTER FOR UNCACHED MEMORY ACCESS

6.1 Technical description

A match in the L2C error status register for an uncached memory access is not handled correctly and can cause a L2C lock up.

The following conditions are needed to trigger the issue, there are two different scenarios:

- Scenario A: An error was detected and stored in the L2C error status register.
  - The access matching against the error status register is enabled (L2CERR.COMP = 1).
  - An uncached memory read access is performed that match the stored access in the L2C error status register.
- Scenario B: The L2C is used as on-chip RAM (all ways are locked)
  - A read access detects an uncorrectable TAG error (in a dirty cache line).

6.2 Functional impact

The access will not complete and the L2C will lock up.

6.3 Workaround

- Scenario A:
  - Disable matching accesses to the L2C error status register (L2CERR.COMP = 0) OR,
  - Disable SPLIT response. This will prevent errors for uncached memory accesses to be stored in the error status register (see Issue 7) OR,
  - Do not define uncached memory areas (using MTRR and HPROT) and keep the L2C enabled.
- Scenario B:
  - Disable matching accesses to the error status register (L2CERR.COMP = 0) OR,
  - Do not lock the entire cache. Have at least one way un-locked.

7 ISSUE 5: INCORRECT DATA WHEN ACCESSING UNCACHED MEMORY OR IO AREAS WITH SPLIT RESPONSES ENABLED

7.1 Technical description

Incorrect data can be returned for non-prefetchable uncached memory read and read from the IO area when SPLIT is enabled. The result depends on previous access to the L2C, if the access is a single access or a burst access and also which area (memory or IO) that is accessed.

The following conditions are needed to trigger the issue (two scenarios).

- For both scenarios:
  - SPLIT response is enabled.
- Scenario A, Access to uncached memory:
  - Uncached memory accesses need to be configured non-prefetchable (L2CACCC.DBPF = 1)
  - Last cache read access handled as a miss did not trigger a full cache fetch.
- Scenario B, Access to the IO area:
  - Last cache read access handled as a miss did not trigger a full cache fetch.

7.2 Functional impact

- Scenario A, Access to uncached memory:
  - Read will return stale data.
  - For a burst access the first data word will be stale data and replicated on offset 0x0 - 0x10.
- Scenario B, Access to the IO area:
  - Single access read will return stale data.
  - For a burst access the first data word will be replicated on offset 0x0 - 0x10.

7.3 Workaround

- Scenario A, Access to uncached memory:
  - Disable SPLIT response OR
  - Do NOT disable prefetching for uncached memory accesses (L2CACCC.DBPF = 0, this is the default setting)
- Scenario B, Access to the IO area:
  - Disable SPLIT responses, OR
  - Enable the "128-bit write line fetch" (L2CACCC.128WF = 1) option. This configures the L2C to always fetch a full cache line from memory.

After reset, the L2C is in the same state as when "Last cache read access handled as a miss did not trigger a full cache fetch". To correct this, a memory read access need to be performed to trigger a cache line fetch. To keep the L2C in this state, the workaround for Scenario B (L2CACCC.128WF = 1) need to be applied.

8 ISSUE 6: AHB ERROR NOT PROPAGATED FOR IO AREA ACCESSES WHEN SPLIT IS ENABLED

8.1 Technical description

AMBA error response encountered when reading the IO area behind the L2C is not propagated when SPLIT is enabled.

The following condition are needed to trigger the issue.

- The L2C has been configured to make use of SPLIT responses.
- Read access to the IO area (0xFFE00000 - 0xFFEFFFFF in GR740) that results in an AMBA error response.

8.2 Functional impact

Read access will complete without AMBA ERROR.

8.3 Workaround

Disable SPLIT response.

The workaround to disable SPLIT responses can be a drastic measure. Section 10 provides additional alternatives where accesses to the IO area are instead avoided.

9 ISSUE 7: MISSING ERROR STATUS REGISTER UPDATES FOR UNCACHED MEMORY AND IO AREA ACCESS

9.1 Technical description

When SPLIT is disabled for uncached memory accesses (or when the IO area is accessed), the error status register is not updated when a AMBA error response is detected on the backend (memory) bus.

The following conditions are needed to trigger the issue.

- SPLIT response need to be disabled OR Access to IO area
- An uncached memory access or an access to the IO area which results in an AMBA error response on the backend bus.

9.2 Functional impact

Error status register not updated to reflect the detected error response on the backend bus.

9.3 Workaround

Do not use the internal error status register to detect errors on the memory bus for uncacheable accesses (including accesses to the IO area).

In the GR740, the memory scrubber (MEMSCRUB) can be used to detect AMBA ERROR responses that occur on the Memory AHB bus.

10 PROPOSED COMBINED WORKAROUND

10.1 Combined workaround overview

It is possible to avoid all issues described in this document by software workarounds or by making use of other resources in a system. The two itemized lists below summarize the workarounds:

- Issue 1 To avoid this issue atomic (read-write) operation is needed when accessing the flush register.
- Issue 2-3 Can be avoided in single CPU systems and by avoiding concurrent DMA access to uncached memory.
- Issue 4 Is Not triggered in the L2-cache default configuration (L2CERR.COMP = 0)
- Issue 3,5-7 Are only triggered for access to uncached memory or the IO area.
- Issue 2-6 Can be avoided when SPLIT response is disabled and the entire cache is not locked (used as on-chip memory)

Expressing the workarounds differently we can summarize the workarounds to:

To avoid issue:

- Issue 1: Do not use L2C flush, OR make use of atomic L2C register accesses.
- Issue 2: Disable SPLIT, OR avoid concurrent access to L2C register accesses (single CPU)
- Issue 3: Disable SPLIT, OR avoid concurrent access to uncached memory and IO area (single CPU)
- Issue 4: (Disable SPLIT AND do not lock all ways in L2C), OR Do not enable L2CERR.COMP (default disabled), OR (no uncached memory access AND do not lock all ways in L2C)
- Issue 5: Disable SPLIT, OR (no IO area access AND no non-prefetchable uncached access) (default prefetchable uncached access) OR (No non-prefetchable uncached access AND Enable the "128-bit write line fetch" (L2CACCC.128WF = 1) option).
    - If SPLIT is enabled then a memory read access needs to be performed to trigger a cache line fetch after reset before any accesses to the L2C IO area.
- Issue 6: Disable SPLIT, OR use MEMSCRUB to report backend bus error, OR avoid IO area access
- Issue 7: (Use other functionality, MEMSCRUB in GR740, to report backend bus error) OR (no uncached memory access AND no IO area access)

The feature "Disable wait-states for discarded bypass data" is recommended to be set (L2CACCC.DBPWS = 1) since it is a workaround for one scenario in Issue 2 and can allow multiple concurrent accesses to the register interface. Issue 2 can still be triggered by the workaround for Issue 1 that recommends to make use of atomic accesses for L2C flush.

10.2 Combined workaround with SPLIT disabled

Applications that make use of the L2C flush functionality will need to be updated to make use of atomic accesses for the L2C flush register accesses.

The remaining issues can be avoided by disabling use of AMBA SPLIT responses from the L2C and relying on other functionality to report AMBA errors (MEMSCRUB in GR740).

10.3 Combined workaround with SPLIT enabled

Disabling use of AMBA SPLIT responses can have a drastic impact on a system since this functionality improves general performance and can reduce effects of interference from accesses on the on-chip bus. For applications that need to keep SPLIT responses enabled, the following approach is proposed:

- Applications that make use of the L2C flush functionality will need to be updated to make use of atomic accesses for the L2C flush register accesses.
- Avoid concurrent accesses to L2C registers, user a spinlock to serialize access in a system with multiple processors that need to access L2C registers.
- Do not lock all ways in the L2C
- The L2C configuration to match incoming access to previous detected errors (stored in the error status register) need to be disabled (L2CERR.COMP = 0 which is the default)
- Uncached memory accesses need to be configured to be prefetchable (L2CACCC.DBPF = 0 which is the default)
- Enable the "128-bit write line fetch" (L2CACCC.128WF = 1) option. This configures the L2C to always fetch a full cache line from memory and avoids issue 5 for IO area accesses.
- After reset, the L2C is in the same state as when "Last cache read access handled as a miss did not trigger a full cache fetch" (issue 5). To correct this, a memory read access need to be performed to trigger a cache line fetch.

10.4 Workaround per access type

Finally, the text below groups the workarounds per access type that were listed in section 2. This section is provided for information, the workarounds described in the two preceding sections are intended to be complete.

To avoid all issue described in this document for uncached accesses, the following configuration and restriction need to be applied:

- Avoid concurrent access to uncached memory areas (CPU or DMA) [Issue 3]
- The L2C configuration to match incoming access to previous detected errors (stored in the error status register) need to be disabled (L2CERR.COMP = 0 which is the default) [Issue 4]
- Uncached memory accesses need to be configured to be prefetchable (L2CACCC.DBPF = 0 which is the default) [Issue 5]
- L2C needs to have SPLIT response enabled or the internal L2C error status register should not be used to detect memory bus errors [Issue 7]

To avoid all the issues described in this document for IO area accesses, the following configuration and restriction need to be applied:

- Limit concurrent access to the IO areas to 1 (CPU or DMA) [Issue 3]
- The IO area should only be accessed when SPLIT is disabled [Issue 5 and 6] or Enable the "128-bit write line fetch" (L2CACCC.128WF = 1) option to avoid Issue 5.
- After reset, a memory read access need to be performed to trigger a cache line fetch to avoid Issue 5 when SPLIT is enabled.
- The internal L2C error status register should not be used to detect bus errors when accessing the IO area [Issue 7]

To avoid all issues described in this document for L2C register access, the following configuration and restriction need to be applied:

- Limit concurrent access to the L2C registers to 1 (CPU or DMA) [Issue 2]
- Only access the L2C flush register with atomic (read-write) operations [Issue 1]

11 SOFTWARE PACKAGES WITH WORKAROUNDS

Software packages provided by Frontgrade Gaisler do not, at the time of writing, include workarounds for the L2C issues. This document will be updated once updates software packages are available.

The following software packages are foreseen to be updated:

- GRBOOT-1.5
- MKPROM2
- GRMON3
- Linux MKLINUXIMG-2.0
- RCC-1.3 RTEMS-5
- VxWorks 6.9 and 7
- BCC2

Please contact support@gaisler.com for an updated schedule for individual packages.

