ARM CoreLink™ Level 2 Cache Controller (L2C-310 or PL310)

r2 releases

Software Developers Errata Notice

ARM®

ARM CoreLink Level 2 Cache Controller (L2C-310 or PL310)
Software Developers Errata Notice

Release Information

The following changes have been made to this book.

			Change History
Date	Issue	Confidentiality	Change
14 June 2012	A	Non-confidential	First release for r2 releases, limited distribution
20 September 2012	B	Non-confidential	Second release for r2 releases

Product Status

The information in this document is final, that is for a developed product.

Web Address

http://www.arm.com

Contents
ARM CoreLink Level 2 Cache Controller (L2C-310 or PL310) Software Developers Errata Notice

Chapter 1
Introduction

This chapter introduces the errata notices for ARM CoreLink Level 2 Cache Controller L2C-310 (PL310).

1.1 Scope of this document

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

- the current status of the defect

- where the implementation deviates from the specification and the conditions under which erroneous behavior occurs

- the implications of the erratum with respect to typical applications

- the application and limitations of a 'work-around' where possible

This document describes errata that may impact anyone who is developing software that will run on implementations of this ARM product.

1.2   Categorization of errata

Errata recorded in this document are split into the following levels of severity:

Table 1-1 Categorization of errata

Errata type	Definition
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.

1.3 Errata summary

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 text of the erratum Description, Conditions, Implications or Workaround. Fixed errata are not shown as updated, unless the erratum text has changed.

Table 1-2 Changes in Document v1

Status	ID	Area	Cat	Rare	Summary of erratum
	727915	Prog	B		Background Clean and Invalidate by Way operation can cause data corruption
	769419	Prog	B		No automatic Store Buffer drain, visibility of written data requires an explicit Cache Sync operation
	738415	Prog	B	Rare	A cacheable read with address bits [20:5] equal to 0x0000 can be stalled by non-cacheable read traffic targeting other address region
	719601	Prog	C		A cacheable read at address 0x0 can be stalled by non-cacheable read traffic targeting other address region
	719603	Prog	C		An AXI locked transaction can cause deadlock when received at the same time as a non-cacheable prefetch hint and the first data of a write
	729815	Prog	C		The "High Priority for SO and Dev reads" feature can cause Quality of Service issues to cacheable read transactions
	754670	Prog	C		A continuous write flow can stall a read targeting the same memory area
	765569	Prog	C		Prefetcher can cross 4KB boundary if offset is programmed with value 23

Chapter 2
Errata Descriptions

This chapter includes the errata descriptions for ARM CoreLink Level 2 Cache Controller L2C-310 (PL310).

2.1   Category A

There are no errata in this category.

2.2   Category A (Rare)

There are no errata in this category.

2.3 Category B

This section describes Category B errata.

2.3.1 (727915) Background Clean and Invalidate by Way operation can cause data corruption

Status

Affects: product PL310 Level-2 Cache Controller.

Fault Type: Cat B

Fault Status: Present in: r2p0, r3p0. Fixed in r3p1. Unchanged in this document.

Description

PL310 implements the Clean and Invalidate by Way L2 cache maintenance operation (offset 0x7FC). This operation runs in the background so that PL310 can handle normal accesses while it is in progress. Under very rare circumstances, because of this erratum, write data can be lost when PL310 treats a cacheable write transaction during a Clean and Invalidate by Way operation.

Conditions

This problem occurs when the following conditions are met:

1. A Clean and Invalidate by Way operation (offset 0x7FC) is in progress.

2. A cacheable write-back write in the store buffer hits in a clean line of the L2 cache.

3. The write does a request to update data of this line in the L2 cache.

4. The maintenance operation also targets this specific line.

Implications

When the conditions above are met in conjunction with some precise timing conditions, the targeted cache line can be invalidated by the maintenance operation at the same time as its data is updated by the write. As a result, the write data is lost and the corresponding address location appears as corrupted.

Workaround

Software workarounds exist for this erratum but they are different between revisions.

- For r2p0, there are two possible workarounds, which have the disadvantage of removing the background aspect of the Clean and Invalidate by Way operation:

    1. Software loop on all sets and ways to cover the whole L2 cache one cache line after the other. The following pseudo-code routine has to replace the Clean and Invalidate by Way operation:
    ```
    loop_ways
    loop_sets
    Clean & Invalidate Line by Set/Way (0x7F8)
    Increment set to cover all sets
    B loop_sets if not all sets are covered (actual number depend on way size)
    Increment way to cover all targeted ways (actual number depends on associativity)
    B loop_ways if not all ways are covered
    ```

    2. Before the CPU responsible for cleaning and invalidating the L2 cache actually issues the background operation, it must acquire complete control of all other L1 masters driving PL310 and prevent them from issuing cacheable accesses for the whole duration of the background operation. Note that if PL310 is connected either to Cortex-A9 or to A5 MPCore processors, "all processors" also include the

masters driving the ACP port. Note also that, in order to guarantee that there is no cacheable access during the operation, it is necessary to disable interrupts. This workaround might have a noticeable impact on the interrupt latency of the whole system.

- For r3p0, the two workarounds listed above also work. However, it is possible to implement another workaround and still benefit from the background aspect of the maintenance operation. Find below the pseudo-code routine that has to replace the Clean and Invalidate by Way operation:

`Disable Write-Back and Cache Linefill (set bits [1:0] of the Debug Control Register)`

`Clean & Invalidate by Way (0x7FC)`

`Re-enable Write-Back and Cache Linefill (reset bits [1:0] of the Debug Control Register)`

If PL310 is integrated in a system where TrustZone is implemented, the workaround described above and involving the Debug Control Register involves calling the Secure world to change the values in the Debug Control Register, because this register is only accessible in Secure world.

2.3.2 (769419) No automatic Store Buffer drain, visibility of written data requires an explicit Cache Sync operation

Status

Affects: product PL310 Level-2 Cache Controller.

Fault Type: Cat B

Fault Status: Present in: r0p0, r1p0, r2p0, r3p0, r3p1. Fixed in r3p2. Unchanged in this document.

Description

Before revision r3p2, the PL310 Store Buffer did not have any automatic draining mechanism. Any data written to one of these devices would consequently remain in the buffer, invisible to the rest of the system.

If an L3 external agent keeps on polling this memory location, waiting to see the update of the written data to make any further progress, then a system livelock might happen.

Conditions

The erratum can only happen on Normal Memory regions.

The following scenario is an example that can exhibit the erratum, where an L3 agent might loop infinitely waiting for the notification from the CPU for an unbounded amount of time:

• An L3 agent is waiting for notification from CPU before making progress

• CPU attached to PL310 issues the necessary notification using a write access, which stays in PL310 store buffer

• No additional activity forcing the store buffer to drain is received by PL310.

Implications

Because of the erratum, a livelock situation might occur in the system.

Workaround

If a write access needs to be made visible to an L3 external agent, the workaround for this erratum consists of using a Cache Sync operation in order to force the PL310 Store Buffer to drain. This is illustrated in the following pseudo-code sequence:

```
STR // to be made visible to L3
DSB
CACHE_SYNC
```

Revision r3p2 implements a counter so that slots are automatically drained after 256 cycles of presence in the store buffer.

2.4 Category B (Rare)

This section describes Category B rare errata.

2.4.1 (738415) A cacheable read with address bits [20:5] equal to `0x0000` can be stalled by non-cacheable read traffic targeting other address region

Status

Affects: product PL310 Level-2 Cache Controller.

Fault Type: Cat B (rare)

Fault Status: Present in: r0p0, r1p0, r2p0. Fixed in r3p0. Unchanged in this document.

Description

When a cacheable read transaction issues its lookup to the L2 cache, address hazard checking is performed between the cacheable read and the active address slots in the PL310 master port(s).

- On r0p0 and r1p0, this address hazard checking is done by only comparing bits[20:5] of the address.

- On r2p0, there is an RTL configuration (`define pl310_FULL_ADDR_HAZARD in pl310_defs.v) that controls the number of bits involved in address hazard checking:

  — If the RTL configuration is not implemented, the behavior is the same as on r0p0 and r1p0. The erratum applies to this case.

  — If the RTL configuration is implemented (this is the recommended configuration), bits[31:5] are taken into account for address hazard checking. In this case, this erratum does not apply; refer to defect #719601.

When non-cacheable reads are active in the master port(s), the corresponding slots are marked as valid but their address is always considered as 0x0 even if they target another address range.

As a result, if the cacheable read with address bits [20:5] is equal to `0x0000`, the address hazard can be detected with non-cacheable reads active in the master port(s).

Conditions

This problem occurs when the following conditions are met:

1. The revision of PL310 is r0p0, r1p0, or r2p0 with pl310_FULL_ADDR_HAZARD configuration not implemented.

2. A cacheable read with address bits [20:5] equal to 0x0000 is received by one PL310 slave port.

3. A continuous flow of non-cacheable reads targeting another address region is serviced by PL310 such that there is always at least one outstanding non-cacheable request that has been issued by PL310.

   In this context, non-cacheable can be Strongly Ordered, Device or Normal Memory Non-Cacheable.

Implications

Under the conditions described, a cacheable read can be unnecessarily stalled by non-cacheable read requests. The cacheable read can only be serviced when there are no outstanding non-cacheable read transactions active in PL310. This is most likely to be an issue where the L2 cache is servicing requests from multiple cores, and so the number of non-cacheable accesses is unbounded.

If there is no dependency between the cacheable read and the non-cacheable traffic, the stall can potentially take a long time. This will not result in functionally incorrect operation.

If there is a dependency between the cacheable read and the non-cacheable traffic, this could in principle result in deadlock. For example, this could occur when the continuous non-cacheable traffic is actually coming from multiple polling loops each of which are waiting for the end of a process and the cacheable read itself is part of the process. The requirements for such multiple polling loops to non-cacheable space dependent on the forward progress of cacheable transactions are thought to be extremely unlikely to occur in general code, but might occur in very specific cases, for example during the booting of a multiprocessor operating system.

Workaround

If necessary, the WFE/SEV workaround can be implemented as follows:

- the primary processor that is responsible for updating a status location at the end of the process does this by executing the following pseudo-code sequence:

```
STR [SO location]
DSB
SEV
```

- the secondary processors that poll the status location do this by executing the following pseudo-code sequence:

```
loop
LDR [SO location]
CMP
WFENE
BNE loop
```

2.5 Category C

This section describes Category C errata.

2.5.1 (719601) A cacheable read at address `0x0` can be stalled by non-cacheable read traffic targeting other address region

Status

Affects: product PL310 Level-2 Cache Controller.

Fault Type: Cat C

Fault Status: Present in: r2p0; fixed in r3p0. Unchanged in this document.

Description

When a cacheable read transaction issues its lookup to the L2 cache, address hazard checking is performed between the cacheable read and the active address slots in the PL310 master port(s).

In r2p0, there is an RTL configuration (`define pl310_FULL_ADDR_HAZARD in pl310_defs.v) that controls the number of bits involved in address hazard checking:

* If the RTL configuration is not implemented, only address bits[20:5] are compared. In this case, this erratum does not apply; refer to defect 738415.

* If the RTL configuration is implemented, bits[31:5] are taken into account for address hazard checking. The erratum applies to this case.

When non-cacheable reads are active in the master port(s), the corresponding slots are marked as valid but their address is always considered as `0x0` even if they target another address range.

As a result, if the cacheable read targets the address `0x0`, some address hazard can be detected with non-cacheable reads active in the master port(s).

Conditions

This problem occurs when the following conditions are met:

1. The RTL configuration pl310_FULL_ADDR_HAZARD is implemented.

2. A cacheable read targeting the `0x00-0x1F` cache line is received by one PL310 slave port.

3. A continuous flow of non-cacheable reads targeting another address region is treated by PL310.

    In this context, non-cacheable can be Strongly Ordered, Device or Normal Memory Non-Cacheable.

Implications

Under the conditions listed above, a cacheable read targeting the 0x00-0x1F cache line can be unnecessarily stalled by non-cacheable read requests. The cacheable read can only be serviced when there are no outstanding non-cacheable read transactions active in PL310. This is most likely to be an issue where the L2 cache is servicing requests from multiple cores, and so the number of non-cacheable accesses is unbounded.

In the ARM architecture, the 0x00-0x1F physical address can be used for storing the exception vectors on a system that does not implement address translation or map the exception vectors to other locations.

If there is no dependency between the cacheable read and the non-cacheable traffic, the stall can potentially take a long time. This does not result in functionally incorrect operation but the unexpected latency of waiting for Non-cacheable locations to drain can cause an increase in interrupt latency.

If there is a dependency between the cacheable read and the non-cacheable traffic, this could in principle result in deadlock. For example, this could occur when the continuous non-cacheable traffic is coming from multiple polling loops, each of which is waiting for the end of a process, and the cacheable read is part of the process. The requirements for such multiple polling loops to non-cacheable space dependent on the forward progress of cacheable transactions are thought to be extremely unlikely to occur in general code, but might occur in very specific cases, for example during the booting of a multiprocessor operating system.

In those specific cases, apply the workaround that uses WFE and SEV.

Workaround

If there is a dependency between the cacheable read and the non-cacheable traffic or if the increased latency is not acceptable, there are several possible workarounds for this erratum.

In a system where the L2 cache is accessed by more than one memory requestor, a workaround is to avoid the physical address range `0x00-0x1F` or to map that physical address range as non-cacheable.

In a system where the L2 cache is accessed by more than one memory requestor including R-profile cores, the R-profile cores should use the HIVECS configuration to move the exception vectors to a different address range.

In a system with the following characteristics:

* the L2 cache is accessed by more than one memory requestor supporting the WFE/SEV scheme;

* the non-cacheable traffic is due to multiple polling loops;

the workaround can be implemented as follows:

* the primary processor that is responsible for updating a status location at the end of the process shall do this by executing the following pseudo-code sequence:

  ```
  STR [SO location]
  DSB
  SEV
  ```

* the secondary processors that poll the status location shall do this by executing the following pseudo-code sequence:

  ```
  loop
  LDR [SO location]
  CMP
  WFENE
  BNE loop
  ```

2.5.2 (719603) An AXI locked transaction can cause deadlock when received at the same time as a non-cacheable prefetch hint and the first data of a write

Status

Affects: product PL310 Level-2 Cache Controller.

Fault Type: Cat C

Fault Status: Present in: r1p0, r2p0. Fixed in r3p0. Unchanged in this document.

Description

When received, an AXI locked transaction (for example caused by a SWP instruction) has the following effects:

- The slave ports drive their AxREADY (ARREADY and AWREADY) outputs low, preventing any new transaction from being accepted

- The internal buffers are drained

- The AXI locked transaction only proceeds after all other transactions already accepted complete.

When connected to the Cortex-A9 MPCore processor, PL310 can receive prefetch hints indicated by ARUSER[8]=1'b1. See the PL310 Cache Controller Technical Reference Manual for more information. If these prefetch hints are non-cacheable, PL310 ignores them.

Conditions

This problem occurs when the following conditions are met:

1. PL310 is connected to the Cortex-A9 MPCore processor and is implemented with two slave ports.

2. Prefetch hints are enabled in Cortex-A9 MPCore and the processor is configured in SMP mode.

3. A memory location is marked as Inner Cacheable Outer Non-Cacheable in the MMU page tables.

4. One A9 CPU executes a SWP instruction leading to an AXI locked read received by one PL310 slave port.

5. At the same time, the other slave port receives a non-cacheable prefetch hint and a write transaction.

6. The prefetch hint is ignored and the first data of the write transaction is accepted by the slave port.

7. The AXI locked transaction implies the two slave ports drive their AxREADY signals low, preventing the address of the write transaction from being accepted.

Implications

Under the conditions listed above, a deadlock can occur. Note that the SWP instruction is deprecated in the ARM architecture and the Cortex-A9 processor has a specific mechanism for enabling its use.

Workaround

A workaround for this erratum is only necessary if PL310 is connected to the Cortex-A9 MPCore processor. In this case, the possible workarounds are:

- Do not enable SWP in the Cortex-A9 processors, or

- Do not enable the prefetch hint capability in the Cortex-A9 processor, or

- Do not have memory locations that are Inner Cacheable Outer Non-Cacheable.

2.5.3 (729815) The "High Priority for SO and Dev reads" feature can cause Quality of Service issues to cacheable read transactions

Status

Affects: product PL310 Level-2 Cache Controller.

Fault Type: Cat C

Fault Status: Present in: r2p0, r3p0, r3p1, r3p2. Unchanged in this document.

Description

The "High Priority for SO and Dev reads" feature can be enabled by setting to 1 the bit[10] of the PL310 Auxiliary Control Register. When enabled, this feature gives priority to Strongly Ordered and Device reads over cacheable reads in the PL310 AXI master interfaces. When PL310 receives a continuous flow of Strongly Ordered or Device reads, this can prevent cacheable reads that miss in the L2 cache from being issued to the L3 memory system.

Conditions

This problem occurs when the following conditions are met:

1. Bit[10] "High Priority for SO and Dev reads enable" of the PL310 Auxiliary Control Register is set to 1.

2. PL310 receives a cacheable read that misses in the L2 cache.

3. PL310 receives a continuous flow of Strongly Ordered or Device reads that take all address slots in the master interface.

Implications

When the conditions above are met, the linefill resulting from the L2 cache miss is not issued until the flow of SO/Device reads stops. Note that each PL310 master interface has four address slots, so that the Quality of Service issue only appears on the cacheable read if the L1 is able to issue at least four outstanding SO/Device reads.

Workaround

A workaround is only necessary in systems that are able to issue a continuous flow of Strongly Ordered or Device reads. In such a case, the workaround is to disable the "High Priority for SO and Dev reads" feature. This is the default behavior.

2.5.4 (754670) A continuous write flow can stall a read targeting the same memory area

Status

Affects: product PL310 Level-2 Cache Controller.

Fault Type: Cat C

Fault Status: Present in: all revisions. Unchanged in this document.

Description

In PL310 r3px (x=0, 1, 2), and previous revisions (r0 to r2) with the pl310_FULL_ADDR_HAZARD RTL option implemented, hazard checking is done on bits [31:5] of the address.

In PL310 r0 to r2 with the pl310_FULL_ADDR_HAZARD RTL option NOT implemented, hazard checking is done on bits [20:5] of the address.

When PL310 receives a read with Normal Memory (cacheable or not) attributes, hazard checking is performed with the active writes of the store buffer. If an address match is detected, the read is stalled until the write completes.

Because of this erratum, a continuous flow of writes can stall a read that targets the same memory area.

Conditions

This problem occurs when the following conditions are met:

1.   PL310 revision is r3, or r0 to r2 with the pl310_FULL_ADDR_HAZARD RTL option implemented.

2.   PL310 receives a continuous write traffic targeting the same address marked with Normal Memory attributes.

3.   While treating this flow, PL310 receives a read that targets the same 32-byte memory area.

OR

1.   PL310 revision is r0 to r2 with the pl310_FULL_ADDR_HAZARD RTL option NOT implemented.

2.   PL310 receives a continuous write traffic with Normal Memory attributes and targeting addresses with identical [20:5] bits.

3.   While treating this flow, PL310 receives a read with the same address bits [20:5].

Implications

When the conditions above are met, the read might be stalled until the write flow stops.

Note that this erratum does not lead to any data corruption.

Note also that normal software code is not expected to contain long write sequence like the one causing this erratum to occur.

Workaround

There is no workaround and there is unlikely to be a need for a workaround for this erratum.

2.5.5   (765569) Prefetcher can cross 4KB boundary if offset is programmed with value 23

Status

Affects: product PL310 Level-2 Cache Controller.

Fault Type: Cat C

Fault Status: Present in: all revisions. Unchanged in this document.

Description

When the prefetch feature is enabled (bits[29:28] of the Auxiliary or Prefetch Control Register set HIGH), the prefetch offset bits of the Prefetch Control Register (bits[4:0]) configure the advance taken by the prefetcher compared to the current cache line. Refer to the PL310 Cache Controller Technical Reference Manual for more information. One requirement for the prefetcher is to not go beyond a 4KB boundary. If the prefetch offset is set to 23 (5'b10111), this requirement is not fulfilled and the prefetcher can cross a 4KB boundary.

Conditions

This problem occurs when the following conditions are met:

1.   One of the Prefetch Enable bits (bits [29:28] of the Auxiliary or Prefetch Control Register) is set HIGH.

2.   The prefetch offset bits are programmed with value 23 (5'b10111).

Implications

When the conditions above are met, the prefetcher can issue linefills beyond a 4KB boundary compared to the original transaction. This can cause system issues because those linefills can target a new 4KB page of memory space, regardless of page attribute settings in L1 MMU.

Workaround

A workaround for this erratum is to program the prefetch offset with any value except 23.