ARML210 Errata List

IP Products Division

Document number:	AS006-PRDC-003350 14.0
Date of Issue:	3 February 2006
Author:	Christophe Evrard
Authorised by:	Peter Middleton

Abstract

This document describes the known errata of all releases up to and including r0p5 of the ARML210 design.

Keywords

ARML210, L2CC, errata

This is a working document throughout the product lifecycle and, as such, the content may be modified as new information is uncovered.

Contents

.

.

.

1 ABOUT THIS DOCUMENT

1.1 Current History

Issue	Date	By	Change
1.0	17 June 2003	Christophe Evrard	New document for r0p0 and r0p1 erratum merge
2.0	21 August 2003	Christophe Evrard	Updated Document for r0p2 release
5.0	22 January 2004	Christophe Evrard	Added new category 2 errata (BWABT).
6.0	22 April 2004	Christophe Evrard	Added erratum [6]. [7] and [8]. For r0p3 release.
6.1	13 May 2005	Christophe Evrard	Added erratum [9]. Draft version
7.0	19 May 2005	Christophe Evrard	Reworked erratum [9]
8.0	25 May 2005	Christophe Evrard	Updated document for r0p2_02 and r0p4 releases
9.0	3 June 2005	Christophe Evrard	Fixed Date in Document header
9.1	29 July 2005	Christophe Evrard	Added erratum [10]. Draft Version
10.0	12 September 2005	Christophe Evrard	Updated document for r0p5 release
11.0	14 September 2005	Christophe Evrard	Fixed Date in Document header
12.0	23 September 2005	Christophe Evrard	Changed revision number in abstract
13.0	30 January 2006	Christophe Evrard	Added erratum [11] and documentation errata
14.0	3 February 2006	Christophe Evrard	Updated erratum [11].

1.2 References

This document refers to the following documents.

Ref.	Document No	Author(s)	Title
1	ARM DDI 0284 F	ARM	ARML210 (r0p5) Technical Reference Manual

.

1.3 Scope

This document describes all the errata discovered in the different implementations of the ARML210, categorised by level of severity. Each description includes:

- where the implementation deviates from the specification
- the conditions under which erroneous behaviour occurs
- the implications of the erratum with respect to typical applications
- the application and limitations of a ‘work-around’ where possible
- The status of corrective action.

1.4 Terms and Abbreviations

This document uses the following terms and abbreviations.

Term	Meaning

.

2 CATEGORISATION OF ERRATA

Errata recorded in this document are split into three groups:

Category 1 Features which are impossible to work around and severely restrict the use of the device in all or the majority of applications rendering the device unusable.

Category 2 Features, which contravene the specified behaviour and may limit or severely impair the intended use of specified features but does not render the device unusable in all or the majority of applications.

Category 3 Features that were not the originally intended behaviour but should not cause any problems in applications.

2.1 Errata Summary

The errata associated with this product are categorised in the following way. Numbers in brackets after the errata description indicate the order in which the errata were found chronologically.

		r0p0	r0p1	r0p2_01	r0p2_02	r0p3	r0p4	r0p5
Doc	TRM must state that any write to registers of ARM L210 must be preceded by an explicit cache-sync operation	X	X	X	X	X	X	X
Category 1	No known errata.							
Category 2	[4] Incorrect IDLE assertion on BUSY to IDLE transaction transition on master ports.	X	X					
	[5] BWABT event only asserted if abort response is received on the last access of buffered write or eviction transaction	X	X	X	X			
	[6] IDLE can be active while a transaction starts on a master port (HTRANSMx = NSEQ)			X	X			
	[7] Abort response can be lost on a non-bufferable write access with master port in 32-bit configuration.	X	X	X	X			
	[9] Evictions can cause write data from write buffer to be overwritten	X	X	X		X		
	[10] Read-after-write hazards can be incorrectly handled between write allocation linefills and dirty data evictions in three master ports configuration.	X	X	X	X	X	X	
	[11]Starting a background maintenance operation while a line is sitting in the write-allocate buffer can prevent that line from being allocated	X	X	X	X	X	X	X

.

Category 3	[1] Incorrect ERRWD event assertion.	X	X					
	[2] Incorrect HTRANS during WRAP16, 64 bits access.	X	X					
	[3] Incorrect ERRRT event assertion.	X	X					
	[8] Incorrect HPROTM1 information for read transaction initiated to a write miss in write-allocate region	X	X	X	X			

.

3 CATEGORY 1 ERRATA

No known errata

.

4 CATEGORY 2 ERRATA

4.1 Incorrect IDLE assertion on BUSY to IDLE transaction transition on master ports.

4.1.1 Summary

IDLE signal is asserted even though there is an outstanding transaction on one master port.

4.1.2 Description

The problem occurs when:

- CLK/HCLK ratio of at least 2:1 (clock division or HCLKEN usage)
- One slave port is doing an INCR burst (unspecified length burst).
- Corresponding master port goes BUSY after receiving the last data beat of this burst transfer.

The result is that IDLE signal is asserted before master HTRANS can transition from BUSY to IDLE.

4.1.3 Implication

If the IDLE signal is used to stop the L210 clocks, the master HTRANS will stay in BUSY cycle and may cause harm to the rest of the memory system.

4.1.4 Software work-around

There is no Software work-around for this errata.

4.1.5 Impacted revisions

Impacted revisions are r0p0 and r0p1.

.

4.2 BWABT event only asserted if abort response is received on the last access of buffered write or eviction transaction.

4.2.1 Summary

Aborts on first accesses of buffered write transactions (write buffer or eviction buffer draining) are not reported through BWABT even if the event bus is enabled. Only aborts received on the last word of the transaction are signalled.

4.2.2 Description

The problem occurs when:

- A line in the write buffer or in the eviction buffer is drained. The write buffer line draining should cause more than a single beat burst transaction (INCR or INCR4).
- One or more error response is received on HRESP bus for any access for the first accesses but not the last one.

The result is that buffered write abort event (BWABT) is not asserted at the end of the transaction even if event bus is enabled.

4.2.3 Implication

If the memory region accessed by buffered write transaction can generate an abort response and that the BWABT is used in the system to report the error by any means, some transactions errors can be missed.

4.2.4 Work-around

There is no Software work-around for this errata.

A hardware work-around is to ensure that if, a slave returns an error response during a burst transaction, the error response remains asserted until the end of the burst.

4.2.5 Impacted revisions

Impacted revisions are r0p0, r0p1, r0p2_01 and r0p2_02.

.

4.3 IDLE can be active while a transaction starts on a master port (HTRANSMx = NSEQ).

4.3.1 Summary

IDLE signal is asserted even though a transaction starts on one master port.

4.3.2 Description

The problem occurs when:

- One line of the write buffer contains valid data meant to be written to a non-write allocate region.
- A single buffered write access on a slave port cause the previous write-buffer line to be drained.
- If no other transaction or background operation is on going, the IDLE signal is asserted at the same time write transaction (resulting from line draining) appears on master port (HTRANSMx = NSEQ).

4.3.3 Implication

If the IDLE signal is used to stop the L210 clocks, the master HTRANS will stay in NSEQ cycle and may cause harm to the rest of the memory system.

4.3.4 Software work-around

There is no Software work-around for this errata.

4.3.5 Impacted revisions

Impacted revisions are r0p2_01 and r0p2_02.

.

4.4 Abort response can be lost on a non-bufferable write access with master port in 32-bit configuration.

4.4.1 Summary

If master ports are configured as being 32-bit wide, non-bufferable 64-bit write accesses on slave ports are split in two sequential 32-bit accesses. An abort response on the master side is signalled to the slave interface only if it appears on the second word access.

4.4.2 Description

The problem occurs when:

- Slave ports are configured as 64-bit wide.
- Master ports are configured as 32-bit wide.
- A non-bufferable 64-bit write transaction is requested on a slave port (S1 or S2).
- The transaction appears as two 32-bit write accesses on master port (M1 or M2).
- If the first write access received an ABORT response, the abort is not reported on the requesting slave port.

4.4.3 Implication

It is not ensured that 64 bit transactions on a 32-bit memory system (down-sizing being done at L210 level) can receive an abort response.

4.4.4 Work-around

There is no Software work-around for this errata.

A hardware work-around is to ensure that if, a slave returns an error response during a burst transaction, the error response remains asserted until the end of the burst.

4.4.5 Impacted revisions

Impacted revisions are r0p0, r0p1, r0p2_01 and r0p1_02.

.

4.5 Evictions can cause write data from write buffer to be overwritten.

4.5.1 Summary

A write buffer drain and an eviction to the same half cache line occurring at the same time can cause the L3 data to be overwritten with older value.

4.5.2 Description

There is a scenario whereby the draining of the eviction buffer and the write buffer does not maintain the correct precedence dictated by the cache controller.

If two half lines are in the eviction buffer, there is a one cycle window between the end of the draining of the 1st half line and starting of the 2nd half line where the write buffer can wrongly take priority over the eviction buffer.

Under most circumstances, this is benign. However, if the 2nd half line in the eviction buffer and the data in the write buffer are from the same set of addresses, the newer data in the write buffer will be drained before the older data from the eviction buffer.

The figure below shows the normal order of write access to L3 memory system. The intended behaviour is that the eviction buffer is always empty when a write buffer slot is draining, so that any hazards between these two buffers are always naturally eliminated.

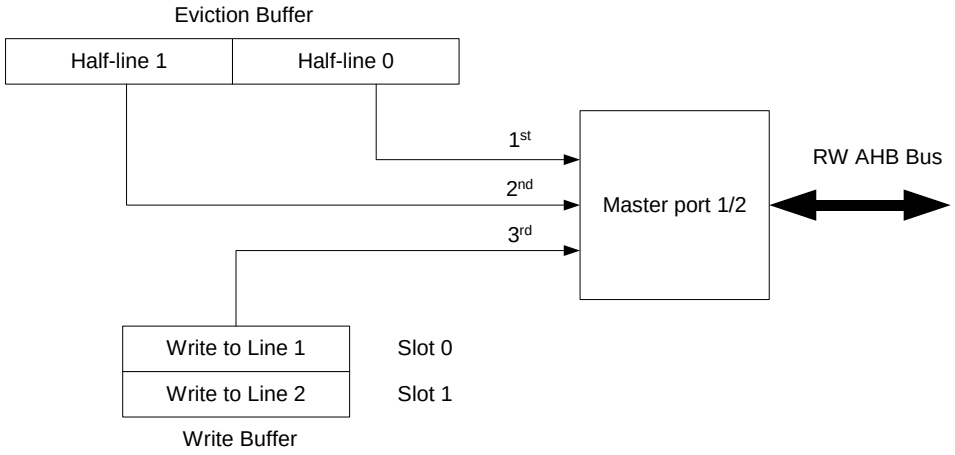

Figure 4-1 Expected order of eviction and write buffer draining

In the current case, the write buffer is actually taking ownership of the master port while a slot in the eviction buffer is still valid. Both slots of the eviction buffer are sharing the same request line to the master port and that request line wrongly goes low for one cycle once the first half eviction has been acknowledge by the master. That cycle gap is enough for the master to see that the next source in its priority list, the write buffer, is requesting access to the bus.

.

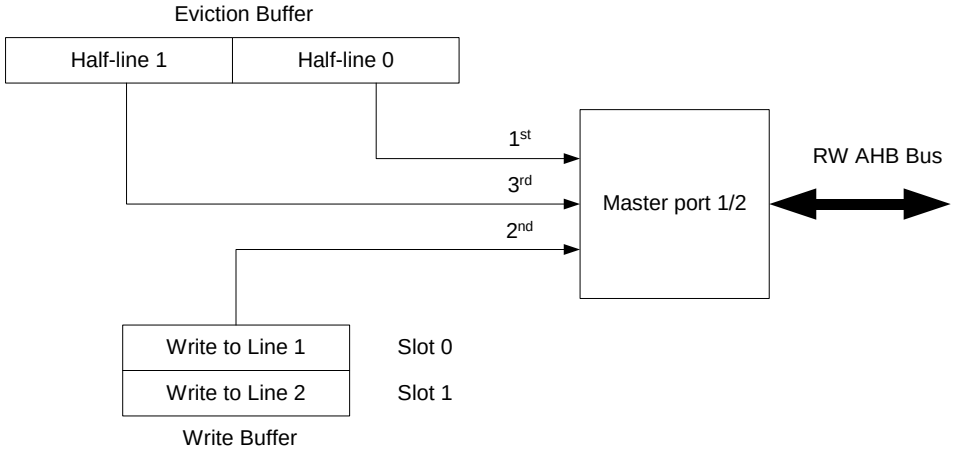

Figure 4-2 Faulty order of eviction and write buffer draining

The result is that the L3 memory now contains the older data from the eviction buffer.

4.5.3 Implication

Data in L3 will be corrupted.

4.5.4 Software work-around

The work-around for this erratum is to use the L210 as a write-through cache only. When in write-through, all cache lines are always clean, so no evictions are ever performed when a cache line is replaced by a new allocation. From the software perspective, there are two ways to use L210 as a write-through cache:

1. Change memory regions attributes so that they are mapped as L2 write-though regions.
2. Set “Disable write-back” bit in ARM L210 Debug Control register (bit [1]). When that bit is set, all cacheable access are treated as write-through, allocate on read miss only in L2.

4.5.5 Impacted revisions

All revisions up to r0p3, except r0p2_02, are impacted.

.

4.6 Read-after-write hazards can be incorrectly handled between write allocation linefills and dirty data evictions in three master ports configuration.

4.6.1 Summary

In L210 three masters configuration only, a write allocation linefill and dirty data eviction to the same half or full line occurring at the same time can cause the L2 to be allocated with out-of-date data.

4.6.2 Description

The problem occurs when:

- L210 is configured so that all three master ports are used.
- And, a cache allocation linefill causes a half or full line to be transferred to the eviction buffer. The eviction does not start at once because master M2 is busy with another transaction.
- And, the evicted line address corresponds to a memory region defined as Outer Write-Back Write-Allocate (OWBWA) or Outer Write-Back Non Write allocate (OWBNWA) with the “Write allocate override” bit of Auxiliary Control register (bit 23) set.
- Then, a write buffer slot containing a write access to the same address is processed. The write buffer sends a lookup request to the cache controller and receives a cache miss answer as the line is now in the eviction buffer.
- The write buffer slot is then transferred to the write-allocate buffer that, if some data is missing in the line, will request a linefill transaction to master M1.
- In the case where master port M2 has not yet started the line eviction transaction, and master port M1 is free, the linefill will occur first in the L3 memory system, fetching out of date data.

4.6.3 Implication

As the write-allocation linefill transaction can start on master port M1 before the dirty data eviction transaction starts on master port M2, the data allocated in L2 cache can be out of date.

4.6.4 Software work-around

The software work-around for this erratum is to change memory regions attributes so that “allocate on read miss only” policy is used for all outer write-back regions. It must also be checked that the “Write allocate override” bit of Auxiliary Control register (bit 23) is not set accordingly.

4.6.5 Hardware work-around

As this erratum only applies to L210 three masters configurations, a hardware work-around is to force MASTNUM[1:0] configuration pins to 2’b00 or 2’b01 in order to only use one or two master ports.

4.6.6 Impacted revisions

All revisions up to r0p4 are impacted.

.

4.7 Starting a background clean or clean-and-invalidate maintenance operation while a line is sitting in the write-allocate buffer can prevent that line from being allocated.

4.7.1 Summary

When a background clean or clean-and-invalidate cache maintenance operation is requested to the L210, all the targeted ways are considered as being locked by the cache controller. If a cache maintenance operation is requested which targets at least all of the non-locked ways between the time a data line has been transferred to the write allocate buffer and the time that the buffer request line allocation to the cache controller has occurred, the cache controller won't allocate the line as all ways are seen as locked.

4.7.2 Description

This problem occurs when the following conditions are met:

1. A write transaction to a write-back write allocate memory region, or to a cacheable memory with Force Write-allocate bit set in Auxiliary Control Register, is pending in one of the write buffer slot.
2. This write buffer slot is drained because of a new write transaction to the write buffer, a hazard or a non-cacheable transaction.
3. The write transaction misses in the cache, causing the line to be transferred from the write buffer to the write allocate buffer. That can only happen if some ways are not locked. If all ways are locked when the write buffer receives the cache miss information, the write buffer forces the write to L3 memory system.
4. A background clean or clean-and-invalidate cache maintenance operation is requested which targets at least all of the non-locked ways before the write-allocate buffer is requesting data line allocation to the cache controller. An automatic cache sync is triggered by the cache maintenance request, causing the write-allocate buffer to be drained.
5. When the write allocate buffer is requesting allocation to the cache controller, the cache controller already considers all cache maintenance targeted ways to be locked, so that if all ways are seen as locked, the allocation is not performed.
6. The write access that has been transferred from the write buffer to the write-allocate buffer is subsequently lost.

4.7.3 Implication

Performing a background clean or clean-and-invalidate cache maintenance request on all non-locked available ways can cause data corruption because the write transaction sitting in the write allocate buffer at that time can be lost.

Even if the Invalidate By Way operation has similar behaviour in terms of way lockdown, no work-around needs to be applied for that operation as any updated data in dirty cache line will be lost anyway when the line will be invalidated.

4.7.4 Software work-around

The software work-around for this erratum is to perform an explicit Cache Sync operation before initiating a background clean or clean-and-invalidate cache maintenance operation. By doing so, the write-allocate buffer is empty when the background cache operation is initiated, therefore removing the erratum conditions.

That work-around relies on the atomicity of the Cache Sync and cache maintenance operations from cache controller perspective so that it is ensured that no write transactions are performed between them. If it appears to be impossible to prove atomicity of the operations, cache maintenance operations should be performed one way

.

at a time so that at least one way is always seen as non-locked. This has no or little performance impact for Clean By Way and Clean-And-Invalidate By Way operations as they are performed one way at a time by the cache controller.

4.7.5 Impacted revisions

All revisions up to r0p5 are impacted.

.

CATEGORY 3 ERRATA

4.8 Incorrect ERRWD event assertion.

4.8.1 Summary

ERRWD event is wrongly asserted if the data read access of a natural eviction (due to an allocation) receives an error response from the data RAM through DATAERR pin.

4.8.2 Description

That problem occurs when:

- The event bus is enabled by setting bit 20 of auxiliary control register.
- Data RAM is capable of generating RAM errors on DATAERR input pin.
- A lookup-miss causes a linefill request.
- Linefill allocation causes an eviction because lines to the corresponding index of unlocked ways are all valid.
- Data RAM raise DATAERR signal on the eviction data read access.

The ARML210 should only report a data read error event through ERRRD but because the main cache controller operation is seen as a data write operation (allocation), a data write error is also reported through ERRWD pin.

4.8.3 Implication

If an event monitor is plugged on ARML210 event bus and is set-up to track data write errors, write errors counting can be erroneous.

4.8.4 Software work-around

There is no Software work-around for this errata.

4.8.5 Impacted revisions

Impacted revisions are r0p0 and r0p1.

.

4.9 Incorrect HTRANS during WRAP16, 64 bits access.

4.9.1 Summary

WRAP16, 64 bits non-cacheable read or non-bufferable write transactions request on a slave port causes incorrect HTRANS (signalling SEQ access instead of NSEQ) on 32-bit configured master for the access with wrapping address.

4.9.2 Description

WRAP16, 64 bits non-cacheable read or non-bufferable write transaction requests on a slave port should be translated into two undefined length burst transactions on a 32-bit configured master. First access starts with first address presented on slave side and the second one should start with the wrapping address, signalled by as a NSEQ transfer by HTRANSMx. In current revisions, that NSEQ access is seen as a SEQ access.

4.9.3 Implication

If a memory system use burst type and size combined with first transaction address, it will not detect address wrapping and will consider the transaction as a single 64 bit burst access with incrementing address.

4.9.4 Software work-around

If wrap access are disabled by setting bit 12 of auxiliary control register, the transaction correctly appears as two undefined length bursts on the master side with correct HTRANSMx information.

4.9.5 Impacted revisions

Impacted revisions are r0p0 and r0p1.

Note: No ARM core currently produces transactions of this type.

.

4.10 Incorrect ERRRT event assertion.

4.10.1 Summary

A cache maintenance operation by index/way can cause an additional incorrect tag read error event assertion (through ERRRT pin) if a read tag error occurred on another way during the previous tag lookup.

4.10.2 Description

That problem occurs when:

- The event bus is enabled by setting bit 20 of auxiliary control register.
- Tag RAM is capable of generating RAM errors on TAGERR input pin.
- A Tag lookup causes a tag read error (signalled through TAGERR pin and ERRRT event).
- That tag lookup is followed by a clean or clean and invalidate by index/way cache maintenance operation to a way different than the one which tag RAM report an error.

The result is that an additional false error is signalled on ERRRT event pin.

However the cache maintenance operation is correctly.

4.10.3 Implication

If an event monitor is plugged on ARML210 event bus and is set-up to track tag read errors, read errors counting can be erroneous.

4.10.4 Software work-around

There is no Software work-around for this errata.

4.10.5 Impacted revisions

Impacted revisions are r0p0 and r0p1.

.

4.11 Incorrect HPROTM1 information for read transaction initiated to a write miss in write-allocate region.

4.11.1 Summary

For read transactions initiated by a write cache miss in a memory region marked as write-allocate, attribute value put on HPROTM1 is defined as write-back read-allocate only.

4.11.2 Description

That problem occurs when:

- A write request is made in a memory region marked as write-back write-allocate on S1 or S2 slave ports.
- The write-buffer line containing that write access is drained (write buffer line is not full).
- The line draining cause a cache lookup that signals a cache miss, so the write transaction is put in the write allocate buffer.
- Master port M1 initiates a read transaction in order to get missing data in the write allocate buffer.

The write allocation is normally completed but the read transaction has been marked as being performed in a write-back read-allocate only memory region.

4.11.3 Implication

L3 memory system (or L3 cache) can't rely on the HPROT information for write-back write-allocate regions if it wishes to distinguish write-back write-allocate from write-back read-allocate only.

4.11.4 Software work-around

There is no Software work-around for this errata.

4.11.5 Impacted revisions

Impacted revisions are r0p0, r0p1, r0p2_01 and r0p1_02.

.

5 ERRATA - DOCUMENTATION

5.1 TRM must state that any write to registers of ARM L210 must be preceded by an explicit cache-sync operation

5.1.1 Summary

A statement of software integration kit README must be made clear in the TRM.

5.1.2 Description

The README provided with the software integration kit contains the following statement:

```
However, writes to any of the other registers should be preceded by an explicit
cache-sync request. This is especially important when the cache is enabled and
changes to how the cache allocates new cache-lines are to be made, such as, cache
lockdown and change of debug control attributes.
```

This statement is very important and must be made visible in the ARM L210 TRM. For example, changing the data cache lockdown register while a line is in the write allocate buffer can cause the allocation of this line in the cache to fail.

5.1.3 Implications

none

5.1.4 Workaround

none

.