ARM PrimeCell
ARM MultiPort Memory Controller (GX175)
Errata Notice

This document contains all errata known at the date of issue in releases up to and including revision.r0p2 of GX175 Memory Controller Primecell It also references known errata on PL175 memory controller which will also affect GX175.

Proprietary notice

Document confidentiality status

This document is Non Confidential.

Web address

http://www.arm.com/

Feedback on the product

If you have any comments or suggestions about this product, contact your supplier giving:

- The product name
- A concise explanation of your comments.

Feedback on this document

If you have any comments on about this document, please send email to mailto:errata@arm.com giving:

- The document title
- The documents number
- The page number(s) to which your comments refer
- A concise explanation of your comments

General suggestion for additions and improvements are also welcome.

Contents

Introduction

Scope

This document describes errata categorised 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

Categorisation of Errata

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

Category 1 Behavior that is impossible to work around and that severely restricts the use of the product in all, or the majority of applications, rendering the device unusable.

Category 2 Behavior that contravenes the specified behavior and that might limit or severely impair the intended use of specified features, but does not render the product unusable in all or the majority of applications.

Category 3 Behavior that was not the originally intended behavior but should not cause any problems in applications.

Change Control

07 Jun 2006: Changes in Document v5

Page	Status	ID	Cat	Summary
13	Updated	368212	Cat 2	Lockup when performing SRAM and SDRAM accesses on multiple AHB ports
15	New	373235	Cat 3	False failure in verilog verification environment

15 Dec 2005: Changes in Document v4

Page	Status	ID	Cat	Summary
13	New	368212	Cat 2	Lockup when performing SRAM and SDRAM accesses on multiple AHB ports
12	Updated	363261	Cat 2	IDLE cycle required after a LOCK transfer
9	Updated	359332	Cat 2	GX port misses some read transactions
14	Updated	358802	Cat 3	Memory access failures if tRAS is set smaller than RAS delay

14 Oct 2005: Changes in Document v3

Page	Status	ID	Cat	Summary
12	New	363261	Cat 2	IDLE cycle required after a LOCK transfer

22 Sep 2005: Changes in Document v2

Page	Status	ID	Cat	Summary
9	Updated	359332	Cat 2	GX port misses some read transactions

14 Sep 2005: Changes in Document v1

Page	Status	ID	Cat	Summary
9	New	359332	Cat 2	GX port misses some read transactions
14	New	358802	Cat 3	Memory access failures if tRAS is set smaller than RAS delay

Errata Summary Table

The errata associated with this product affect product versions as below.

A cell shown thus **X** indicates that the defect affects the revision shown at the top of that column.

ID	Cat	Summary of Erratum	r0p0	r0p1	r0p2
359332	Cat 2	GX port misses some read transactions	X		
363261	Cat 2	IDLE cycle required after a LOCK transfer	X	X	
368212	Cat 2	Lockup when performing SRAM and SDRAM accesses on multiple AHB ports	X	X	
358802	Cat 3	Memory access failures if tRAS is set smaller than RAS delay	X	X	X
373235	Cat 3	False failure in verilog verification environment			X

Errata - Category 1

There are no Errata in this Category

Errata - Category 2

359332: GX port misses some read transactions

Status

Affects: product GX175 Memory Controller Primecell.

Fault status: Cat 2, Present in: r0p0, Fixed in r0p1-00REL0. Unchanged in this document.

Description

A read transfer can be missed on the GX port when the DMC (Dynamic Memory Controller) is busy and all 10 of the tags in the GX port are in use.

The hardware fix requires the following lines of code to be modified in the MpmcGxIfReqGen.v file.

1) The following line needs to be added after line 531.

```
wire   OldRdReqServed;
```

2) Line 635 (p_AddrGenComb):

```
always @(NxtTmpGAddr or DriveAgain or iGAddrQ2 or NxtTmpGBstrb or iGBstrbQ2 or
         iGWDataQ2 or NxtTmpGWdata or iTagQ2 or NxtTmpTag or TagFree or
         WtdReq or TagFreeQ)
begin : p_AddrGenComb
   iGAddrQ          = iGAddrQ2;
   iGBstrbQ         = iGBstrbQ2;
   iGWdataQ         = iGWDataQ2;
   iTagQ            = iTagQ2;
  if (DriveAgain == 1'b0 | (TagFree == 1'b1 & WtdReq == 1'b0 &
      DriveAgain == 1'b1 & TagFreeQ == 1'b0))
    begin
      iGAddrQ          = NxtTmpGAddr;
      iGBstrbQ         = NxtTmpGBstrb;
      iGWdataQ         = NxtTmpGWdata;
      iTagQ            = NxtTmpTag;
    end
end // p_AddrGenComb
```

Becomes (the addition of OldRdReqServed):

```
always @(NxtTmpGAddr or DriveAgain or iGAddrQ2 or NxtTmpGBstrb or iGBstrbQ2 or
         iGWDataQ2 or NxtTmpGWdata or iTagQ2 or NxtTmpTag or TagFree or
         WtdReq or TagFreeQ or OldRdReqServed)
begin : p_AddrGenComb
   iGAddrQ          = iGAddrQ2;
   iGBstrbQ         = iGBstrbQ2;
   iGWdataQ         = iGWDataQ2;
   iTagQ            = iTagQ2;
```

```
  if (DriveAgain == 1'b0 | OldRdReqServed |
      (TagFree == 1'b1 & WtdReq == 1'b0 & DriveAgain == 1'b1 & TagFreeQ == 1'b0))
    begin
      iGAddrQ          = NxtTmpGAddr;
      iGBstrbQ         = NxtTmpGBstrb;
      iGWdataQ         = NxtTmpGWdata;
      iTagQ            = NxtTmpTag;
    end
end // p_AddrGenComb
```

3) Line 939 (p_IntWrReqComb):

```
always @(DyBufFull or Delay4Boot or TagFree or OldReqServed or IntRdReqQ or
         WtdRdReq)
begin : p_IntRdReqComb
  if (DyBufFull == 1'b0 & Delay4Boot == 1'b0 & TagFree == 1'b1)
    begin
      if (OldReqServed == 1'b0)
        begin
          iMemRdReqQ       = IntRdReqQ;
        end
      else
        begin
          iMemRdReqQ       = WtdRdReq;
        end
    end
  else
    begin
      iMemRdReqQ       = 1'b0;
    end
end // p_IntRdReqComb
```

Becomes (OldReqServed becomes OldRdReqServed):

```
always @(DyBufFull or Delay4Boot or TagFree or OldRdReqServed or IntRdReqQ or
         WtdRdReq)
begin : p_IntRdReqComb
  if (DyBufFull == 1'b0 & Delay4Boot == 1'b0 & TagFree == 1'b1)
    begin
      if (OldRdReqServed == 1'b0)
        begin
          iMemRdReqQ       = IntRdReqQ;
        end
      else
        begin
          iMemRdReqQ       = WtdRdReq;
        end
    end
  else
    begin
```

```
      iMemRdReqQ       = 1'b0;
    end
end // p_IntRdReqComb
```

4) Line 1021:

```
assign NxtWtdRdReq      = (iGAREADY == 1'b1 & (DyBufFull == 1'b1 |
                          GxiGntCo == 1'b0 | TagFree == 1'b0) &
                          (IntRdReqQ == 1'b1 |
                          IntWrReqQ == 1'b1) & GTRANS == 1'b1 & GSELG == 1'b1
                          & GWRITE == 1'b0 & DelGAREADY == 1'b1
                          & ErrorCond == 1'b0) ? 1'b1 : ((OldReqServed == 1'b1
                          & DyBufFull == 1'b0 &
                          GxiGntCo == 1'b1) ? 1'b0 : WtdRdReq);
```

Becomes (addition of TagFree and OldReqServed becomes OldRdReqServed):

```
assign NxtWtdRdReq      = (iGAREADY == 1'b1 & (DyBufFull == 1'b1 |
                          GxiGntCo == 1'b0 | TagFree == 1'b0) &
                          (IntRdReqQ == 1'b1 |
                          IntWrReqQ == 1'b1) & GTRANS == 1'b1 & GSELG == 1'b1
                          & GWRITE == 1'b0 & DelGAREADY == 1'b1
                          & ErrorCond == 1'b0) ? 1'b1 : ((OldRdReqServed == 1'b1
                          & DyBufFull == 1'b0 & TagFree &
                          GxiGntCo == 1'b1) ? 1'b0 : WtdRdReq);
```

5) The following line needs to be added:

```
assign OldRdReqServed = (OldReqServed && !MemWrReqQ2);
```

Implications

If the GX175 misses a read transfer on the GX port the GX master connected to the memory controller will receive the following read data rather than the data that it requested.

This could cause the GX master to behave erratically.

When the GX175 has been running for a long time a number of read transfers may be missed. This could cause the GX master to lock up, as it will be waiting for the results of the read transactions (that were missed) before it can submit any more transactions.

Workaround

The software workaround requires that the MBX only access data in a single SDRAM bank.

This workaround requires the memory controller to be configured in Bank, Row, Column (BRC)

address mapping mode.

363261: IDLE cycle required after a LOCK transfer

Status

Affects: product GX175 Memory Controller Primecell.

Fault status: Cat 2, Present in: r0p0,r0p1, Fixed in r0p2. Unchanged in this document.

Description

If any UNLOCKED transfer immediately follows a LOCKED transfer without an IDLE cycle then PL175 hangs on this AHB transfer.This AHB transfer does not complete and PL175 continues to hold HREADYOUT low.

Implications

The PL175 is unable to support masters which do not insert an IDLE cycle after a LOCK transfer.

Workaround

If an IDLE cycle is inserted after a LOCK transfer, then PL175 does not exhibit unexpected behaviour on any transfer following this IDLE cycle.

368212: Lockup when performing SRAM and SDRAM accesses on multiple AHB ports

Status

Affects: product GX175 Memory Controller Primecell.

Fault status: Cat 2, Present in: r0p0,r0p1, Fixed in r0p2. Updated in this document.

Description

This defect occurs in the following situation when multiple AHB ports are accessing SRAM and SDRAM at the same time:

1 - An SDRAM row is open.

2 - A bufferable SRAM write burst is performed on a low priority port (Port A) where the AHB transfer width is less than the memory width, and the burst contains more than 32 bits of data in total. The HPROT bits have to be set to indicate the write is bufferable, and the MPMC AHB port's buffer is enabled.

3 - During the AHB write burst, a higher priority port (Port B) requests an SDRAM read from the open row.

4 - Also during the AHB write burst, an even higher priority port (Port C) requests an SRAM read.

5 - Port A's write burst is followed immediately by a cacheable 8 or 16-bit SDRAM read with a burst type other than SINGLE from the currently open row.

When all of these conditions are met, the PL175 MPMC may stop responding and hold

HREADYOUT low on all AHB ports.

Implications

The symptoms of this defect are that the SRAM write burst will be terminated early on the external memory bus, the SDRAM reads will occur, then the static memory controller will attempt the SRAM read but will lock up due to it not completing the SRAM write burst.

This causes all AHB port accesses to lock up with **HREADYOUT** held low.

Workaround

This defect can be avoided by not meeting the particular conditions described, which includes but is not limited to:

- do not perform bufferable writes to SRAM (by controlling the **HPROT** value or disabling the AHB port's buffer) with the AHB transfer width less than the SRAM width at the same time as SDRAM accesses

- do not perform bufferable writes to SRAM with the AHB transfer width less than the SRAM width and the total data greater than 32 bits at the same time as SDRAM accesses

- do not perform cacheable SRAM reads with a width of 8 or 16-bit at the same time as SDRAM accesses

Errata - Category 3

358802: Memory access failures if tRAS is set smaller than RAS delay

Status

Affects: product GX175 Memory Controller Primecell.

Fault status: Cat 3, Present in: r0p0,r0p1,r0p2, Open. Unchanged in this document.

Description

If you program the value of MPMCDynamictRAS to be smaller than the RAS delay value in the MPMCDynamicRasCasX register then memory access errors occur.

Most, if not all memory devices, require MPMCDynamictRAS to be larger than RAS in the MPMCDynamicRasCasX register. Therefore this problem will not be encountered for most memory devices.

Implications

The value programmed into the MPMCDynamictRAS register must always be larger than the value programmed in the RAS field of the MPMCDynamicRasCasX register.

Workaround

When programming the memory controller ensure that MPMCDynamictRAS is larger than RAS in the MPMCDynamicRasCasX register.

373235: False failure in verilog verification environment

Status

Affects: product GX175 Memory Controller Primecell.

Fault status: Cat 3, Present in: r0p2, Open. New in this document.

Description

When BufArb6DyWrRdTest.c test is run in the verilog verification environment, it reports read failures. However, the same test run in the vhdl verification environment passes.

The failure reported for this test in the verilog verification environment is false. The test results between the two verification worlds differ because of delta timing differences between the test components in the two worlds. This timing difference leads to AHB transfers happening in a slightly different order between the two worlds.

This test fires various write and read requests through all the AHB ports. It checks data integrity by comparing the data read back from a memory location with the data written to it. This check is made with respect to write and read transfers made from one port. However, it doesnot have the check that no other port has written to the same memory location before a read is made from this memory location. This is because the addresses chosen for the various transfers through the different ports are generated randomly and are not expected to coincide between requests on different ports.

In the verilog environment, the timing differences are such that the same memory location is written to through two different AHB ports. When the read is made from the memory location, the test expects to read back the value that was written through the same port that the read is being made from. However the value read back is what was written to the memory location last through another AHB port. Hence a mismatch error is reported. If the test had included an extra check to make sure addresses between different ports don't conicide, this failure would not have been reported.

Hence the failure reported here is a false negative and can be ignored.

Implications

A failure is reported in the verilog verification world when BufArb6DyWrRdTest.s test is run. This test reports a read error. This error is false and hence can be ignored.

Workaround

None.

Errata - Documentation

There are no Errata in this Category

Errata – Driver Software

There are no Errata in this Category