



ARM PrimeCell®  
SRAM NOR Flash Memory Controller (PL352)  
**Errata Notice**

This document contains all errata known at the date of issue in releases up to and including revision r2p1 of SRAM NOR Flash Memory Controller -3YT

## Proprietary notice

Words and logos marked with ® or ™ are registered trademarks or trademarks of ARM Limited in the EU and other countries, except as otherwise stated below in this proprietary notice. Other brands and names mentioned herein may be the trademarks of their respective owners.

Neither the whole nor any part of the information contained in, or the product described in, this document may be adapted or reproduced in any material form except with the prior written permission of the copyright holder.

The product described in this document is subject to continuous developments and improvements. All particulars of the product and its use contained in this document are given by ARM Limited in good faith. However, all warranties implied or expressed, including but not limited to implied warranties of merchantability, or fitness for purpose, are excluded.

This document is intended only to assist the reader in the use of the product. ARM Limited shall not be liable for any loss or damage arising from the use of any information in this document, or any error or omission in such information, or any incorrect use of the product.

## Document 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.

## 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

<b>INTRODUCTION</b>	<b>6</b>
<b>ERRATA SUMMARY TABLE</b>	<b>9</b>
<b>ERRATA - CATEGORY 1</b>	<b>12</b>
<b>410562:</b> Memory interface locks up when EBIBACKOFF asserted during turnaround time of burst finishing at boundary	<b>12</b>
<b>ERRATA - CATEGORY 2</b>	<b>14</b>
<b>372884:</b> Mode updates not holding off subsequent transactions	<b>14</b>
<b>410561:</b> tAVH violation on reads in mux-mode operation	<b>15</b>
<b>415054:</b> Error when asynchronous write follows a synchronous write and is to a different chip-select	<b>16</b>
<b>415513:</b> CS remaining asserted between transactions is not supported by CRAM 1.0	<b>17</b>
<b>456580:</b> Async mux_mode behaviour is not supported by some flash devices.	<b>18</b>
<b>456826:</b> PREADY not returned early for APB writes to direct_cmd register when PCLKEN is not tied high	<b>19</b>
<b>534963:</b> Potential data error when writing to SRAM with ACLK async to MCLK	<b>20</b>
<b>ERRATA - CATEGORY 3</b>	<b>22</b>
<b>408515:</b> Memory Configuration Register does not indicate mux_mode correctly	<b>22</b>
<b>409123:</b> Setting tWP to 1 in async mux_mode gives a tWP value of 0.	<b>23</b>
<b>412923:</b> Can enter low_power mode while transactions are still outstanding	<b>24</b>
<b>418214:</b> Register update mechanism mismatching manager commands	<b>25</b>
<b>445213:</b> Remap functionality can be wrong	<b>26</b>
<b>447567:</b> READY not returned when zero's written to memc_cfg_set & memc_cfg_clr	<b>27</b>
<b>533517:</b> FIFO depths chosen in AMBA Designer may be inconsistent	<b>28</b>
<b>583618:</b> Explicit Perl paths causing RTL rendering issues on certain machines	<b>29</b>
<b>ERRATA - DOCUMENTATION</b>	<b>31</b>
<b>381891:</b> Row boundary behaviour not documented	<b>31</b>
<b>404182:</b> DDI0380C SMC (PL350 series) r1p1 TRM, incorrect reset value for sram_cycles register	<b>32</b>
<b>404184:</b> DDI0380C SMC (PL350 series) r1p1 TRM, incorrect waveform in Fig 2-16	<b>33</b>
<b>404865:</b> DDI0380C SMC (PL350 series) r1p1 TRM, Figure 3-5 has incorrect address mapping	<b>34</b>

<b>405861:</b>	<b>Incorrect version of documentation in release</b>	<b>35</b>
<b>408513:</b>	<b>DDI0380D SMC (PL350 series) r1p2 TRM, memory configuration register description is incorrect</b>	<b>36</b>
<b>408962:</b>	<b>DII0137C SMC (PL350 series) r1p1 IM, how to connect EBI to SMC when the interfaces run at different frequencies</b>	<b>37</b>
<b>411811:</b>	<b>DDI0380C SMC (PL350 series) r1p1 TRM, Fig 5-1 shows incorrect values for set_cycles_val</b>	<b>38</b>
<b>442865:</b>	<b>DDI0380E SMC (PL350) TRM, incorrect ecc_value field width in ecc_value register</b>	<b>39</b>
<b>ERRATA – DRIVER SOFTWARE</b>		<b>41</b>
<b>There are no Errata in this Category</b>		<b>41</b>

## 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

### Related Products

This document only captures defects specific to the ZA815 product. The current release is r0p0-00dev0 and has been based on PL301-r1p1-00bet2. All defects on that apply to this PL301 revision will apply to ZA815 too and details can be found in the latest PL301-DC-11001 Errata document. If instantiating the DMP then all declared erratum relating to PL368 will also apply and document PL368-DC-11001 should be consulted.

### 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

### 04 Nov 2008: Changes in Document v8

Page	Status	ID	Cat	Summary
20	New	534963	Cat 2	Potential data error when writing to SRAM with ACLK async to MCLK
19	Updated	456826	Cat 2	PREADY not returned early for APB writes to direct_cmd register when PCLKEN is not tied high
29	Updated	583618	Cat 3	Explicit Perl paths causing RTL rendering issues on certain machines
28	New	533517	Cat 3	FIFO depths chosen in AMBA Designer may be inconsistent

### 23 Oct 2007: Changes in Document v7

Page	Status	ID	Cat	Summary
14	Updated	372884	Cat 2	Mode updates not holding off subsequent transactions
18	Updated	456580	Cat 2	Async mux_mode behaviour is not supported by some flash devices.
19	Updated	456826	Cat 2	PREADY not returned early for APB writes to direct_cmd register when PCLKEN is not tied high
27	New	447567	Cat 3	PREADY not returned when zero's written to memc_cfg_set & memc_cfg_clr
22	Updated	408515	Cat 3	Memory Configuration Register does not indicate mux_mode correctly
25	Updated	418214	Cat 3	Register update mechanism mismatching manager commands
31	Updated	381891	Doc	Row boundary behaviour not documented
32	Updated	404182	Doc	DDI0380C SMC (PL350 series) r1p1 TRM, incorrect reset value for sram_cycles register
33	Updated	404184	Doc	DDI0380C SMC (PL350 series) r1p1 TRM, incorrect waveform in Fig 2-16
34	Updated	404865	Doc	DDI0380C SMC (PL350 series) r1p1 TRM, Figure 3-5 has incorrect address mapping
39	Updated	442865	Doc	DDI0380E SMC (PL350) TRM, incorrect ecc_value field width in ecc_value register

### 21 Sep 2007: Changes in Document v6

Page	Status	ID	Cat	Summary
16	New	415054	Cat 2	Error when asynchronous write follows a synchronous write and is to a different chip-select
17	New	415513	Cat 2	CS remaining asserted between transactions is not supported by CRAM 1.0
39	New	442865	Doc	DDI0380E SMC (PL350) TRM, incorrect ecc_value field width in ecc_value register
26	New	445213	Cat 3	Remap functionality can be wrong
18	New	456580	Cat 2	Async mux_mode behaviour is not supported by some flash devices.
19	New	456826	Cat 2	PREADY not returned early for APB writes to direct_cmd register when PCLKEN is not tied high
23	Updated	409123	Cat 3	Setting tWP to 1 in async mux_mode gives a tWP value of 0.

**22 Dec 2006: Changes in Document v5**

Page	Status	ID	Cat	Summary
25	New	418214	Cat 3	Register update mechanism mismatching manager commands

**21 Nov 2006: Changes in Document v4**

Page	Status	ID	Cat	Summary
12	New	410562	Cat 1	Memory interface locks up when EBIBACKOFF asserted during turnaround time of burst finishing at boundary
14	New	372884	Cat 2	Mode updates not holding off subsequent transactions
15	New	410561	Cat 2	tAVH violation on reads in mux-mode operation
22	New	408515	Cat 3	Memory Configuration Register does not indicate mux_mode correctly
23	New	409123	Cat 3	Setting tWP to 1 in async mux_mode gives a tWP value of 0.
24	New	412923	Cat 3	Can enter low_power mode while transactions are still outstanding
33	New	404184	Doc	Incorrect waveform in Fig 2-16 of TRM
34	New	404865	Doc	TRM Figure 3-5 has incorrect address mapping
36	New	408513	Doc	Memory configuration register description incorrect in TRM
37	New	408962	Doc	Lack of documentation on connecting EBI to memory controllers that run at different frequencies
38	New	411811	Doc	TRM error on set_cycles_val
31	Updated	381891	Doc	Row boundary behaviour not documented
32	Updated	404182	Doc	Incorrect reset value for sram_cycles register
35	Updated	405861	Doc	Incorrect version of documentation in release

**27 Sep 2006: Changes in Document v3**

Page	Status	ID	Cat	Summary
35	New	405861	Doc	Incorrect version of documentation in release
32	New	404182	Doc	Incorrect reset value for sram_cycles register in TRM
31	New	381891	Doc	Row boundary behaviour not documented

**10 Jul 2006: Changes in Document v2**

No Errata in this document revision

**18 Apr 2006: Changes in Document v1**

No Errata in this document revision

## 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.

**NOTE:** Some documentation errata are not directly applicable to the PL352 but are included in this table as the documentation is common across the PL351, PL352, PL353, PL354 products. This may result in certain product revisions appearing in the table for which no corresponding PL352 release actually exists.

ID	Cat	Summary of Erratum	r0p0-00rel0	r1p0-00rel0	r1p1-00rel0	r1p1-00rel1	r1p1-01rel0	r1p1-01rel1	r1p2-00rel0	r2p0-00rel0	r2p0-02rel0	r2p1-00rel0
381891	Doc	Row boundary behaviour not documented		X								
404182	Doc	DDI0380C SMC (PL350 series) r1p1 TRM, incorrect reset value for sram_cycles register	X	X	X	X	X	X				
404184	Doc	DDI0380C SMC (PL350 series) r1p1 TRM, incorrect waveform in Fig 2-16			X	X	X	X	X			
404865	Doc	DDI0380C SMC (PL350 series) r1p1 TRM, Figure 3-5 has incorrect address mapping			X	X	X	X	X			
405861	Doc	Incorrect version of documentation in release				X						
408513	Doc	DDI0380D SMC (PL350 series) r1p2 TRM, memory configuration register description is incorrect			X	X	X	X	X			
408962	Doc	DII0137C SMC (PL350 series) r1p1 IM, how to connect EBI to SMC when the interfaces run at different frequencies			X	X	X	X	X			

ID	Cat	Summary of Erratum														
			r0p0-00rel0	r1p0-00rel0	r1p1-00rel0	r1p1-00rel1	r1p1-01rel0	r1p1-01rel1	r1p2-00rel0	r2p0-00rel0	r2p0-02rel0	r2p1-00rel0				
411811	Doc	DDI0380C SMC (PL350 series) r1p1 TRM, Fig 5-1 shows incorrect values for set_cycles_val			X	X	X	X	X							
442865	Doc	DDI0380E SMC (PL350) TRM, incorrect ecc_value field width in ecc_value register											X			
410562	Cat 1	Memory interface locks up when EBIBACKOFF asserted during turnaround time of burst finishing at boundary	X	X	X	X	X	X								
372884	Cat 2	Mode updates not holding off subsequent transactions	X	X	X	X	X	X								
410561	Cat 2	tAVH violation on reads in mux-mode operation	X	X	X	X	X	X								
415054	Cat 2	Error when asynchronous write follows a synchronous write and is to a different chip-select	X	X	X	X	X	X								
415513	Cat 2	CS remaining asserted between transactions is not supported by CRAM 1.0	X	X	X	X	X	X								
456580	Cat 2	Async mux_mode behaviour is not supported by some flash devices.	X	X	X	X	X	X	X	X	X					
456826	Cat 2	PREADY not returned early for APB writes to direct_cmd register when PCLKEN is not tied high	X	X	X	X	X	X	X	X	X					
534963	Cat 2	Potential data error when writing to SRAM with ACLK async to MCLK	X	X	X	X	X	X	X	X	X	X	X	X	X	X

ID	Cat	Summary of Erratum																
			r0p0-00rel0	r1p0-00rel0	r1p1-00rel0	r1p1-00rel1	r1p1-01rel0	r1p1-01rel1	r1p2-00rel0	r2p0-00rel0	r2p0-02rel0	r2p1-00rel0						
408515	Cat 3	Memory Configuration Register does not indicate mux_mode correctly		X	X	X	X	X										
409123	Cat 3	Setting tWP to 1 in async mux_mode gives a tWP value of 0.		X	X	X	X	X										
412923	Cat 3	Can enter low_power mode while transactions are still outstanding	X	X	X	X	X	X										
418214	Cat 3	Register update mechanism mismatching manager commands		X	X	X	X	X										
445213	Cat 3	Remap functionality can be wrong	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
447567	Cat 3	READY not returned when zero's written to memc_cfg_set & memc_cfg_clr	X	X	X	X	X	X	X									
533517	Cat 3	FIFO depths chosen in AMBA Designer may be inconsistent							X	X	X	X	X	X				
583618	Cat 3	Explicit Perl paths causing RTL rendering issues on certain machines	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

## Errata - Category 1

### **410562: Memory interface locks up when EBIBACKOFF asserted during turnaround time of burst finishing at boundary**

#### **Status**

Affects: product AXI Static Memory Controller, SRAM NOR Flash Memory Controller -3YT.

Fault status: Cat 1, Present in: r0p0-00rel0,r1p0-00rel0,r1p1-00rel0,r1p1-00rel1,r1p1-01rel0,r1p1-01rel1, Fixed in r1p2-00rel0.

#### **Description**

PSRAM devices deassert the wait signal at the end of a burst that finishes at or about to cross a page boundary.

The wait signal is sampled on the feedback clock which is only driven during memory transactions. If the last rising edge of the feedback clock does not sample wait high after the transaction then the wait signal will remain low internally until the next memory access starts. The last value sampled on the wait signal depends on a combination of clock speed, feedback clock delay and the time between chip-select deassertion and the wait signal returning to its high impedance state.

If the sampled wait signal remains low and a turnaround time is required i.e. between a read and write then if EBIBACKOFF is asserted during that turnaround time the memory interface will lock-up.

This problem can occur with all PSRAM devices.

#### **Implications**

Under the above described conditions, the memory interface will lock up and will never release the EBI.

#### **Workaround**

none



## Errata - Category 2

### **372884: Mode updates not holding off subsequent transactions**

#### **Status**

Affects: product AXI Static Memory Controller, SRAM NOR Flash Memory Controller -3YT.

Fault status: Cat 2, Present in: r0p0-00rel0,r1p0-00rel0,r1p1-00rel0,r1p1-00rel1,r1p1-01rel0,r1p1-01rel1,  
Fixed in r1p2-00rel0.

#### **Description**

The PL350 provides a mechanism for synchronising the update of a memory configuration register with its own internal configuration registers.

Following an update, subsequent commands should be formatted using the new settings. However, because the command fifo in the memory interface can be populated immediately prior to the update occurring, the commands already in the fifo when the update occurs will be incorrectly formatted for the new settings.

Commands during an update should be throttled in the format block such that the final command that performs the update is the only command active in the memory interface at that time.

#### **Implications**

This defect will affect a system which is accessing a chip whilst performing an update which, for example, changes burst length to a smaller value.

Typically it is expected that the change would be from default (smaller) burst lengths to longer, in which case the resulting behaviour should not cause any issues.

#### **Workaround**

There is no workaround for this issue.

**410561: tAVH violation on reads in mux-mode operation****Status**

Affects: product AXI Static Memory Controller, SRAM NOR Flash Memory Controller -3YT.

Fault status: Cat 2, Present in: r0p0-00rel0,r1p0-00rel0,r1p1-00rel0,r1p1-00rel1,r1p1-01rel0,r1p1-01rel1,  
Fixed in r1p2-00rel0.

**Description**

When reads are made to mux-mode memory devices, the timing between the deassertion of ADV and the address changing (tAVH) is violated.

This is because the address is changed on the same clock edge as ADV is deasserted.

tAVH is a mux-mode parameter and hence this violation does not occur with non-mux-mode memory devices.

**Implications**

This tAVH violation would cause protocol errors on the memory interface and could result in data corruption.

**Workaround**

This problem can be avoided by not performing async mux-mode reads. Synchronous mux-mode reads are not affected by this defect.

**415054: Error when asynchronous write follows a synchronous write and is to a different chip-select****Status**

Affects: product AXI Static Memory Controller, SRAM NOR Flash Memory Controller -3YT.

Fault status: Cat 2, Present in: r0p0-00rel0,r1p0-00rel0,r1p1-00rel0,r1p1-00rel1,r1p1-01rel0,r1p1-01rel1,  
Fixed in r1p2-00rel0.

**Description**

When an asynchronous write immediately follows a synchronous write and is therefore to a different chip-select, then the asynchronous write is treated as a synchronous read.

**Implications**

This erratum results in the asynchronous write being treated as a synchronous read by the memory device.

**Workaround**

Program all chip selects to run in the same mode of operation.

**415513: CS remaining asserted between transactions is not supported by CRAM 1.0****Status**

Affects: product AXI Static Memory Controller, SRAM NOR Flash Memory Controller -3YT.

Fault status: Cat 2, Present in: r0p0-00rel0,r1p0-00rel0,r1p1-00rel0,r1p1-00rel1,r1p1-01rel0,r1p1-01rel1,  
Fixed in r1p2-00rel0.

**Description**

The SMC does not deassert chip-select between immediately consecutive transfers to memory. This behaviour is not supported by CRAM 1.0

**Implications**

Not deasserting CS between transactions could cause undefined behaviour from the memory. One of the results could be the memory returning incorrect data for back to back read transfers.

**Workaround**

None.

**456580: Async mux\_mode behaviour is not supported by some flash devices.****Status**

Affects: product AXI Static Memory Controller, SRAM NOR Flash Memory Controller -3YT.

Fault status: Cat 2, Present in: r0p0-00rel0,r1p0-00rel0,r1p1-00rel0,r1p1-00rel1,r1p1-01rel0,r1p1-01rel1,r1p2-00rel0,r2p0-00rel0,r2p0-02rel0, Fixed in r2p1-00rel0.

**Description**

In async mux mode, WE is asserted at the same time as CS. However, for some flash devices, WE must be asserted after the address phase of the transfer.

**Implications**

Some mux\_mode flash devices may not be supported.

**Workaround**

None.

**456826: PREADY not returned early for APB writes to direct\_cmd register when PCLKEN is not tied high****Status**

Affects: product AXI Static Memory Controller, SRAM NOR Flash Memory Controller -3YT.

Fault status: Cat 2, Present in: r0p0-00rel0,r1p0-00rel0,r1p1-00rel0,r1p1-00rel1,r1p1-01rel0,r1p1-01rel1,r1p2-00rel0,r2p0-00rel0,r2p0-02rel0, Fixed in r2p1-00rel0.

**Description**

If the APB is clocked at a slower frequency than ACLK (i.e. PCLKEN is not tied high) then PREADY may not be returned until the completion of the access. For APB direct\_command accesses this can involve a sequence of the current APB access and one or more AXI accesses. In this instance PREADY will not be returned until the data match occurs.

**Implications**

In some systems it may not be possible to write to the AXI channel until the APB channel has completed. This could cause a system deadlock. Please note that the NAND flash memory controller is not affected by this problem.

**Workaround**

When executing AXI direct\_command sequences ensure PCLKEN is tied high.

**534963: Potential data error when writing to SRAM with ACLK async to MCLK****Status**

Affects: product AXI Static Memory Controller, SRAM NOR Flash Memory Controller -3YT.

Fault status: Cat 2, Present in: r0p0-00rel0,r1p0-00rel0,r1p1-00rel0,r1p1-00rel1,r1p1-01rel0,r1p1-01rel1,r1p2-00rel0,r2p0-00rel0,r2p0-02rel0,r2p1-00rel0, Open.

**Description**

When writes are performed by the memory controller, write data and command are pushed into the respective FIFOs in the same cycle. When running with ACLK asynchronous to MCLK, there is no guarantee that the two sets of data will arrive in the MCLK domain in the same cycle. The write data may be one cycle later.

The memory FSM only checks that the write data has arrived when it is in the IDLE state.

A problem can occur if the FSM is not idle when a new write command is popped from the command FIFO. This new command can be started back to back with a previous command but there is no check to see that the write data has arrived.

This problem can only occur when ACLK/MCLK are asynchronous doing non mux mode writes to SRAM where all the write data for the memory burst is contained in a single AXI beat.

The NAND interface is unaffected.

**Implications**

If this occurs then data written to memory will be corrupted.

**Workaround**

The following work arounds exist

1) For async non-muxmode writes use a memory burst length that is long enough to contain more than one AXI beat (normally memory burst length of 4 will do this).

This workaround has least impact on performance because AXI bursts of data (greater than 1 beat) are treated optimally on the memory interface.

This workaround can be used provided the memory device supports back to back transactions without chip select being deasserted.

2) If the memory device requires chip select to be deasserted between bursts, then the best workaround is to set the refresh\_period register to 1. This ensures the FSM returns to idle between every transaction. This adds tTR idle cycles between each transaction of a burst that would otherwise have completed back to back.



## Errata - Category 3

### **408515: Memory Configuration Register does not indicate mux\_mode correctly**

#### **Status**

Affects: product AXI Static Memory Controller, SRAM NOR Flash Memory Controller -3YT.

Fault status: Cat 3, Present in: r1p0-00rel0,r1p1-00rel0,r1p1-00rel1,r1p1-01rel0,r1p1-01rel1, Fixed in r1p2-00rel0.

#### **Description**

The Memory Configuration Register does not correctly report if an interface is in mux\_mode as stated in the TRM.

#### **Implications**

none

#### **Workaround**

none

**409123: Setting tWP to 1 in async mux\_mode gives a tWP value of 0.****Status**

Affects: product AXI Static Memory Controller, SRAM NOR Flash Memory Controller -3YT.

Fault status: Cat 3, Present in: r1p0-00rel0,r1p1-00rel0,r1p1-00rel1,r1p1-01rel0,r1p1-01rel1, Fixed in r1p2-00rel0.

**Description**

Setting tWP to 1 in async mux\_mode gives a write pulse width of 0.

**Implications**

none

**Workaround**

To work around the problem set tWP = 2 which will give a longer but valid write pulse width.

**412923: Can enter low\_power mode while transactions are still outstanding****Status**

Affects: product AXI Static Memory Controller, SRAM NOR Flash Memory Controller -3YT.

Fault status: Cat 3, Present in: r0p0-00rel0,r1p0-00rel0,r1p1-00rel0,r1p1-00rel1,r1p1-01rel0,r1p1-01rel1,  
Fixed in r1p2-00rel0.

**Description**

If the SMC is idle and receives a power down request followed by an AXI transaction in the same or subsequent cycle the transaction will get registered into the AXI interface but the SMC will still enter low-power mode.

If all the necessary data has been received (or the transaction is a read) it will be completed by the SMC. However, if there is still data outstanding the transaction will not complete until a power-up request is received.

If this occurs, resetting the SMC while in low-power mode, before all the transactions have been completed, would result in AXI protocol violations.

**Implications**

AXI protocol violations could result from this activity.

**Workaround**

There are no registers in the SMC that cannot be accessed while not in low-power mode. Therefore, low-power mode should not be entered unless all masters have completed accessing the SMC.

## **418214: Register update mechanism mismatching manager commands**

### **Status**

Affects: product AXI Static Memory Controller, SRAM NOR Flash Memory Controller -3YT.

Fault status: Cat 3, Present in: r1p0-00rel0,r1p1-00rel0,r1p1-00rel1,r1p1-01rel0,r1p1-01rel1, Fixed in r1p2-00rel0.

### **Description**

The PL350 controller series implement a mechanism to synchronise updates to format and timing configuration registers with updates to memory devices' internal registers.

If a "ModeReg And UpdateRegs" direct command is applied whilst a "ModeReg" direct command to the same chip is being processed in the controller's pipeline then the update logic incorrectly updates the controller's internal registers when the first ModeReg command completes on the memory interface.

For example, a PSRAM device has a Bus Configuration Register (BCR) that dictates parameters such as burst length. A use case for programming the controller and memory to a new burst length is to issue a "ModeReg and UpdateRegs" direct command.

PSRAM devices also contain a Refresh Configuration Register (RCR) to control its self-refresh function. A use case for programming the memory to a new setting is to issue a "ModeReg" direct command.

If the RCR update is performed immediately prior to the BCR, the described error condition could occur.

### **Implications**

If an AXI access is accepted into the format stage of the pipeline after the internal registers have updated, but before the second ModeReg command, then it will be formatted with the new settings. This will result in unpredictable behaviour.

### **Workaround**

The recommended workaround is to always perform "ModeReg and UpdateRegs" commands before "ModeReg" commands.

For example, in the case described above involving PSRAM, perform the BCR then the RCR.

**445213: Remap functionality can be wrong****Status**

Affects: product AXI Static Memory Controller, SRAM NOR Flash Memory Controller -3YT.

Fault status: Cat 3, Present in: r0p0-00rel0,r1p0-00rel0,r1p1-00rel0,r1p1-00rel1,r1p1-01rel0,r1p1-01rel1,r1p2-00rel0,r2p0-00rel0,r2p0-02rel0,r2p1-00rel0, Open.

**Description**

The remap pins on PL35x can be used to map chip 0 of a memory interface to address 0x00000000.

However, it aliases chip 0 but does not hide other chips out of the way.

For example if PL35x is set up so that chip 1 is at 0x00000000 and chip 0 is at 0x01000000 then when remap is asserted PL35x would try to map an incoming AXI command to both chips.

The other chip can be on either interface on dual interface configurations

There are three possible (pre-silicon) workarounds

- 1) Do not place any memory chip selects at base address 0x00000000
- 2) Configure chip select 0 at address 0x00000000 therefore removing the need for a remap.
- 3) Tie the address match and mask pins to programmable registers to allow the chip select addresses to be programmable.

**Implications**

PL350 does not support accesses to two chips at the same time. The behaviour is undefined.

**Workaround**

None

**447567: READY not returned when zero's written to memc\_cfg\_set & memc\_cfg\_clr****Status**

Affects: product AXI Static Memory Controller, SRAM NOR Flash Memory Controller -3YT.

Fault status: Cat 3, Present in: r0p0-00rel0,r1p0-00rel0,r1p1-00rel0,r1p1-00rel1,r1p1-01rel0,r1p1-01rel1,r1p2-00rel0, Fixed in r2p0-00rel0.

**Description**

The memc\_cfg\_set and memc\_cfg\_clear registers are write-only registers in which each bit enables or disables a particular function.

Writing 32'h00000000 to these registers should have no functional effect. However,

**pready** is never returned if such a write does occur.

(This will prevent **hready** being returned in an AHB based system)

**Implications**

The APB (or connected AHB) transaction will never complete.

**Workaround**

Do not write zero to either the memc\_cfg\_clear or memc\_cfg\_set registers.

## **533517: FIFO depths chosen in AMBA Designer may be inconsistent**

### **Status**

Affects: product AXI Static Memory Controller, SRAM NOR Flash Memory Controller -3YT.

Fault status: Cat 3, Present in: r1p1-01rel1,r1p2-00rel0,r2p0-00rel0,r2p0-02rel0,r2p1-00rel0, Open.

### **Description**

When configuring the memory controller in AMBA Designer there is a potential case that choosing Read FIFO depths may cause confusion.

In the configuration window, the desired Read FIFO depth can be selected for each memory interface (for example "12").

If the desired memory width is then chosen, the Read FIFO depth is reset to the default for that memory width (eg "2"). This is because the valid options change for different memory widths.

If the "OK" button is pressed, the memory controller is generated with the default Read FIFO depth (eg "2").

If the right click AMBA Designer -> "reconfigure" option is now selected, the configuration window is opened. The value in the Read FIFO depth is the "12" input above, not that of the rendered memory controller (which was the default "2").

If the "OK" button is pressed the memory controller will be generated with the Read FIFO depth from the configuration window (here "12"). This would lead to changing the Read FIFO depth, without specifically choosing to.

The values in the configuration window, when "OK" is pressed are always those used for RTL generation.

### **Implications**

Confusion may arise when selecting FIFO depths.

The options in the configuration window are those used in the RTL generation process.

### **Workaround**

Check all the values in the configuration window, before pressing the "OK" button.

The values in the configuration window are those used in the RTL generation process.

**583618: Explicit Perl paths causing RTL rendering issues on certain machines****Status**

Affects: product AXI Static Memory Controller, SRAM NOR Flash Memory Controller -3YT.

Fault status: Cat 3, Present in: r0p0-00rel0,r1p0-00rel0,r1p1-00rel0,r1p1-00rel1,r1p1-01rel0,r1p1-01rel1,r1p2-00rel0,r2p0-00rel0,r2p0-02rel0,r2p1-00rel0, Open.

**Description**

The Perl script used to render the desired configuration of the RTL has a hard coded path to where it expects Perl to be installed.

shared/bin/render.pl

contains:

```
#!/usr/local/bin/perl -w
```

If a suitable Perl version is not installed here, an error can occur.

The cause of the error can be hard to trace.

**Implications**

During the render process an error may occur.

The RTL will not be rendered successfully.

**Workaround**

The simplest workaround is to create a unix soft link so that the installed version of Perl is found in /usr/local/bin/perl.

Alternatively the render.pl script can be modified to replace

```
#!/usr/local/bin/perl -w
```

with

```
eval "exec perl -w -S $0 $@" # -*- Perl -*-
```

```
if ($running_under_some_sh);
```

```
undef ($running_under_some_sh);
```

This ensures the script uses the installed version of Perl.



## Errata - Documentation

### **381891: Row boundary behaviour not documented**

#### **Status**

Affects: product AXI Static Memory Controller, SRAM NOR Flash Memory Controller -3YT.

Fault status: Doc, Present in: r1p0-00rel0, Fixed in r1p1-00rel0.

#### **Description**

The documentation does not specify the behaviour of the controller with respect to memory row boundaries.

The controller has no knowledge of row boundary information so cannot prevent memory bursts crossing such boundaries. However, the burst alignment functionality of the controller allows bursts to always be aligned, and not cross, memory burst address boundaries. Since memory burst boundaries are smaller than row boundaries, in this mode row boundaries will not be crossed.

#### **Implications**

Some memories, for example Cellular RAM, do not support row boundary crossing. The `burst_align` bit of the `set_opmode` register must be left at 1'b0 so that bursts are aligned to memory burst address boundaries. This will ensure that memory row boundaries are never crossed.

#### **Workaround**

none

**404182: DDI0380C SMC (PL350 series) r1p1 TRM, incorrect reset value for sram\_cycles register****Status**

Affects: product AXI Static Memory Controller, SRAM NOR Flash Memory Controller -3YT.

Fault status: Doc, Present in: r0p0-00rel0,r1p0-00rel0,r1p1-00rel0,r1p1-00rel1,r1p1-01rel0,r1p1-01rel1,  
Fixed in r1p2-00rel0.

**Description**

In Table 3-1 of the TRM DDI0380C, the reset value of register sram\_cycles<x>\_<n> is given as 0x0000AAFF. This is incorrect. The reset value should be set to 0x0002B3CC.

**Implications**

none

**Workaround**

none

**404184: DDI0380C SMC (PL350 series) r1p1 TRM, incorrect waveform in Fig 2-16****Status**

Affects: product AXI Static Memory Controller, SRAM NOR Flash Memory Controller -3YT.

Fault status: Doc, Present in: r1p1-00rel0,r1p1-00rel1,r1p1-01rel0,r1p1-01rel1,r1p2-00rel0, Fixed in r2p0-00rel0.

**Description**

In Fig 2-16 of the TRM DDI0380C, the WE waveform is incorrect. It is shown as being asserted many cycles after CS gets asserted.

WE should be shown to go low, that is get asserted on the same cycle as CS.

WE in this figure should also be shown to be asserted for  $twp+2$  cycles instead of  $twp$ , since in asynchronous mux-mode WE is asserted for  $twp+2$  cycles.

The following restriction on timing parameters should also be mentioned:

For writes in mux mode,  $tWC$  should be set to a value  $\geq tWP + 2$ .

**Implications**

none

**Workaround**

none

**404865: DDI0380C SMC (PL350 series) r1p1 TRM, Figure 3-5 has incorrect address mapping****Status**

Affects: product AXI Static Memory Controller, SRAM NOR Flash Memory Controller -3YT.

Fault status: Doc, Present in: r1p1-00rel0,r1p1-00rel1,r1p1-01rel0,r1p1-01rel1,r1p2-00rel0, Fixed in r2p0-00rel0.

**Description**

The top four registers

Comp\_id\_3, Comp\_id\_2, Comp\_id\_1, Comp\_id\_0

should be replaced by

pcell\_id\_3, pcell\_id\_2, pcell\_id\_1 and pcell\_id\_0 respectively.

The addresses shown on the right hand side of the figure are not correct.

From bottom to top, they should be

0xFE0 (corresponding to Periph\_id\_0), 0xFE4, 0xFE8, 0xFEC, 0xFF0, 0xFF4, 0xFF8 and 0xFFC.

**Implications**

none

**Workaround**

none

**405861: Incorrect version of documentation in release****Status**

Affects: product AXI Static Memory Controller, SRAM NOR Flash Memory Controller -3YT.

Fault status: Doc, Present in: r1p1-00rel1, Fixed in r1p1-01rel0.

**Description**

Incorrect documentation versions relating to product revision r1p0 were delivered in error with the release of product r1p1-00rel1.

**Implications**

None

**Workaround**

Documentation from the previous (r1p1-00rel0) or subsequent (r1p1-01rel0) releases can be used.

### **408513: DDI0380D SMC (PL350 series) r1p2 TRM, memory configuration register description is incorrect**

#### **Status**

Affects: product AXI Static Memory Controller, SRAM NOR Flash Memory Controller -3YT.

Fault status: Doc, Present in: r1p1-00rel0,r1p1-00rel1,r1p1-01rel0,r1p1-01rel1,r1p2-00rel0, Fixed in r2p0-00rel0.

#### **Description**

In the Memory Interface Configuration Register description on page 3-8 of TRM DDI0380C

bits[9:8] memory\_type1 should read

Returns the memory interface 1 type:

b00 = Configuration does not include this memory interface

b01 = SRAM non-muxed

b10 = NAND

b11 = SRAM muxed

If b00, the remaining bit slices for memory interface 1 are always read as 0.

(not SRAM mixed as it currently states)

similarly bits[1:0] memory\_type0 should read

Returns the memory interface 0 type:

b00 = Reserved

b01 = SRAM non-muxed

b10 = NAND

b11 = SRAM muxed.

However due to an rtl defect the register will never read b11. If the rtl defect is not fixed both the above descriptions should be:

b01 = SRAM

b10 = NAND

b11 = Reserved.

#### **Implications**

none

#### **Workaround**

none

**408962: DII0137C SMC (PL350 series) r1p1 IM, how to connect EBI to SMC when the interfaces run at different frequencies****Status**

Affects: product AXI Static Memory Controller, SRAM NOR Flash Memory Controller -3YT.

Fault status: Doc, Present in: r1p1-00rel0,r1p1-00rel1,r1p1-01rel0,r1p1-01rel1,r1p2-00rel0, Fixed in r2p0-00rel0.

**Description**

On page 2-9 of the IM, the following note should be added for better clarification of system connectivity:

If the EBI were used to interface between two memory controllers (for instance SMC0 and SMC1), and SMC0 and the EBI were running faster than SMC1, then the EBIGRANT signal from EBI to SMC1 must be synchronised to the slow clock domain of SMC1.

Since EBICLOCK is synchronous to the memory controller clock domain, one flip-flop is enough to match the EBIGRANT signal from the EBICLK domain to the memory controller clock domain.

This logic is applicable even if the two different memory interfaces of PL350 are running at different frequencies, or if one of the memory controllers were a DMC.

**Implications**

none

**Workaround**

none

## **411811: DDI0380C SMC (PL350 series) r1p1 TRM, Fig 5-1 shows incorrect values for set\_cycles\_val**

### **Status**

Affects: product AXI Static Memory Controller, SRAM NOR Flash Memory Controller -3YT.

Fault status: Doc, Present in: r1p1-00rel0,r1p1-00rel1,r1p1-01rel0,r1p1-01rel1,r1p2-00rel0, Fixed in r2p0-00rel0.

### **Description**

In the TRM DDI0380C, Figure 5-1 says:

```
set_cycles_val = (t6<<18) | (t5<<15) | (t4<<12) | (t3<<9) | (t2<<6) | (t1<<3) | (t0);
```

This should be changed to

```
set_cycles_val = (t6<<20) | (t5<<17) | (t4<<14) | (t3<<11) | (t2<<8) | (t1<<4) | (t0);
```

### **Implications**

none

### **Workaround**

none

**442865: DDI0380E SMC (PL350) TRM, incorrect ecc\_value field width in ecc\_value register****Status**

Affects: product AXI Static Memory Controller, SRAM NOR Flash Memory Controller -3YT.

Fault status: Doc, Present in: r2p0-00rel0, Fixed in r2p0-02rel0.

**Description**

In PL350 r2p0 TRM, in table 3-22

Field ecc\_value is given as bits [24:0]. This should be [23:0].

The reserved bits are given as bits [26:25]. This should be [26:24].

**Implications**

None

**Workaround**

None



## Errata – Driver Software

**There are no Errata in this Category**