ARM PrimeCell
Single Master DMA Controller (PL081)
Errata Notice

This document contains all errata known at the date of issue in releases up to and including revision r1p2 of PL081 DMA Controller -Perpetual

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

03 Mar 2011: Changes in Document v6

Page	Status	ID	Cat	Summary
13	New	756020	Cat 2	A write to a channel config register coincident with a slave error response can cause the channel to lock up
14	New	756021	Cat 3	Corrupted read data when a non-channel register is read immediately following a write to a channel register

09 May 2006: Changes in Document v5

Page	Status	ID	Cat	Summary
21	Updated	364107	Doc	Integration Test Output Register 3 bit order in TRM
20	Updated	334140	Doc	Setting TransferSize in TRM
19	Updated	330499	Doc	Transfer size count direction in TRM
17	Updated	320216	Doc	Various errors in TRM

24 Oct 2005: Changes in Document v4

Page	Status	ID	Cat	Summary
9	Updated	313733	Cat 2	Only DMACSREQ serviced if both DMACBREQ and DMACSREQ asserted
21	New	364107	Doc	Integration Test Output Register 3 bit order incorrect in TRM

07 Dec 2004: Changes in Document v3

Page	Status	ID	Cat	Summary
9	Updated	313733	Cat 2	Only DMACSREQ serviced if both DMACBREQ and DMACSREQ asserted
11	New	328551	Cat 2	Incorrect Update of LLI Loading
17	New	320216	Doc	Various errors in TRM
19	New	330499	Doc	TRM transfer size count direction
20	New	334140	Doc	Setting TransferSize

18 Dec 2003: Changes in Document v2

Page	Status	ID	Cat	Summary
9	New	313733	Cat 2	Only DMACSREQ serviced if both DMACBREQ and DMACSREQ asserted

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	1	r1p1-00ltd0	r1p2-00rel0	r1p2-01rel0
320216	Doc	Various errors in TRM		X	X	
330499	Doc	Transfer size count direction in TRM		X	X	
334140	Doc	Setting TransferSize in TRM	X	X	X	
364107	Doc	Integration Test Output Register 3 bit order in TRM			X	
313733	Cat 2	Only DMACSREQ serviced if both DMACBREQ and DMACSREQ asserted		X		
328551	Cat 2	Incorrect Update of LLI Loading	X	X	X	X
756020	Cat 2	A write to a channel config register coincident with a slave error response can cause the channel to lock up		X	X	X
756021	Cat 3	Corrupted read data when a non-channel register is read immediately following a write to a channel register		X	X	X

Errata - Category 1

There are no Errata in this Category

Errata - Category 2

313733: Only DMACSREQ serviced if both DMACBREQ and DMACSREQ asserted

Status

Affects: product PL081 DMA Controller -Perpetual.

Fault status: Cat 2, Present in: r1p1-00ltd0, Fixed in r1p2-00rel0.

Description

If the DMAC has been programmed with the DMAC as the flow controller performing burst transfers with both the DMACBREQ and DMACSREQ signals connected, then the DMAC does not operate in accordance with the TRM. If the DMACSREQ signal is asserted first before the DMACBREQ when there is more than (or equal to) a burst transfer to be performed, it performs only single transfers and no burst transfers (unless the DMACSREQ is inactive).

The correct functionality should be such that:

The peripheral should to be able to assert both the DMACSREQ and DMACBREQ signals.

The DMA controller should look at the DMA transfer length. If there are less than a burst of transfers to perform the DMAC should look at the DMACSREQ signal.

If there is more than (or equal to) a burst transfer to perform the DMAC should ignore the DMACSREQ signal and wait to see if the DMACBREQ is asserted and so use that.

Thus the DMAC should perform burst transfers until the last amount of data to be transferred is less than the burst size, under control of DMACBREQ (regardless of DMACSREQ being active), at this point it will complete transfers by performing singles, under the control of DMACSREQ. It will ignore single requests until it cannot complete the final transfers with bursts.

However, the DMA controller will perform a single transfer whenever the DMACSREQ signal goes active and not wait for the DMACBREQ. This is very inefficient as a large number of single requests are generated.

Example:

Data to transfer - 66 words in bursts of 16.

Correct behaviour - The DMAC will perform four bursts of 16 word transfers and then two single word transfers. It will not perform any singles transfers until it cannot complete a whole burst (after it has transferred the 64 words).

Incorrect behaviour - Due to this defect, the DMAC will perform 66 single word transfers.

Implications

With the DMAC as flow controller, if both burst request (BREQ), and single request (SREQ) signals are asserted by a peripheral, the DMA controller will perform a single transfer whenever

the SREQ signal goes active. This is very inefficient as a large number of single requests

can be generated.

Workaround

If a DMA-capable device is able to be suitably configured, it can assert only the BREQ signal while bursts are required. It can then assert SREQ only once most of the data is transferred and single transfers are desired.

328551: Incorrect Update of LLI Loading

Status

Affects: product PL081 DMA Controller -Perpetual.

Fault status: Cat 2, Present in: 1,r1p1-00ltd0,r1p2-00rel0,r1p2-01rel0, Open.

Description

There is some incorrect updating of a linked list due to a last request being serviced twice. This occurs when the device connected to the DMAC is operating at a lower frequency than the DMAC.

- The peripheral, operating at a lower clock frequency than the DMA controller, requests the last item of data through a single transaction (DMACLSREQ).

- Following this since the LLI address pointer in the previous descriptor is Non-zero, a new LLI load is initiated. (When the peripheral is the flow controller DMACLSREQ or DMACLBREQ starts off an LLI load sequence.)

- Before starting the LLI load, DMACCLR is asserted indicating the peripheral to Clear the DMACLSREQ.

However since the peripheral runs on a Slower clock, at least 2 cycles(assuming double synchronization) of the peripheral clock would be required for it to detect DMACCLR and hence de-assert DMACLSREQ.

Since the DMAC clock is significantly faster than peripheral clock, by the time the peripheral samples DMACCLR, the DMAC would have completed an LLI load and would be ready for the next transaction.

- DMACLSREQ (the original one that caused the LLI load) still being HIGH is sampled once again, which leads to the internal TxrSizeSrc(DmacChRegBlock.v) counter to be loaded with '1'. This non-zero value in the counter causes the internal request mask to go inactive. Thus DMACLSREQ is once again seen internally and the Source transfer state machine (which will in this case fetch data from memory into the FIFO) kicks off.(Even when the peripheral is the flow controller, a good part of the source control logic is reused to fetch data from memory.)

- Since the TrfSizeSrc counter is loaded with 1 and the request is a Last Single Request, in peripheral flow control mode as soon as one item of data is fetched, the state machine SrcDmacState checks if the next descriptor is valid or not. If the next descriptor is not valid

(as indicated by a zero on next descriptor address pointer) the SrcDmacState state machine issues an internal request mask (SrcDisable), so that no further requests will be serviced.

- By this time the peripheral can finally sample DMACCLR and deassert its DMACLSREQ and assert DMACBREQ to start the actual transfer using the new descriptor loaded. For the transaction to complete, data has to be fetched from memory and passed on to the peripheral. DMACCLR (for the new DMACBREQ), which indicates completion of requested transfer, can be asserted only when data equivalent to the programmed burst length is transferred to the peripheral.

- However, DMACBREQ does not reach the SrcDmacState state machine (which has to fetch data from memory)- due to the incorrect trigger caused by the previous DMACLSREQ that had resulted in the setting of a mask(SrcDisable) and locked the Source State machine until the channel is reset or re-enabled.

- Now since the required data for the DMACBREQ is not fetched from memory (actually 1 against programmed Burst count for a DMACBREQ) and the source state machine is locked out, DMACBREQ is never acknowledged with a DMACCLR and the channel goes into a deadlock. A ChannelEn disable enable sequence would be required to bring the system out of this deadlock situation.

Symptoms

This issue was found in the PL080 whereby during a link list operation, a burst request failed to be serviced and caused the DMAC to hang. This was as a result of the signal DMACLSREQ being serviced twice during its assertion.

This was due to the fact that at that point LdAfterLLI(DmacChReqProc.vhd)asserts LdSrcInDstFlow which loads TrfSizeSrc (DmacChRegBlock.vhd).

Because of the second transfer, BrstCntSrc (DmacChSrcXfer.vhd) becomes zero which in turn makes SrcDisable bit HIGH indicating incorrectly that the LLI transfer has completed.

Consequently the state machine 'SrcDmacState' in 'DmacChSrcXfer.vhd' goes to the IDLE state and since SrcDisable bit is set, DMACBREQ is masked.

Affected Items

DmacChReqProc

Implications

During the processing of a link list, if a new DMA request is issued it fails to be serviced and the DMAC hangs.

This issue occurs when the device connected to the DMAC is operating at a lower frequency than the DMAC but does not occur when the device connected to the DMAC is operating at the same frequency as the DMAC.

Workaround

Edit the line number 859 in the file DmacChReqProc.vhd to:

```
LdAfterLLI <= (FinishedLLI and AllQualDstReq) and (not(DMACClrDst));
```

Edit the line number 979 in the file DmacChReqProc.v to:

```
assign LdAfterLLI       = FinishedLLI & AllQualDstReq & ~(DMACClrDst);
```

756020: A write to a channel config register coincident with a slave error response can cause the channel to lock up

Status

Affects: product PL081 DMA Controller -Perpetual.

Fault status: Cat 2, Present in: r1p1-00ltd0,r1p2-00rel0,r1p2-01rel0, Open.

Description

A channel within PL081 will lock-up when the following conditions are met:

1. The channel is enabled and performing accesses on an AHB master port
2. A write access to the channel config register is made with the Enable bit (bit 0) set (the expected use case here is writing to the Halt bit)
3. In the same cycle as the write access is made a slave error response is received on the AHB master port for the channel

In this case the ErrorMask flag within DmacChReqMask is set but never cleared, preventing any further activity by the channel.

The expected use case that would cause this bug to manifest is a write to the config register to set the Halt bit (bit 18) with all other bits unmodified.

Implications

After a coincident write and error response which meets the conditions for the bug the channel can no longer be used. The only way to re-enable the channel is to reset PL081 using the HRESETn input signal.

Workaround

When a channel is already enabled (bit 0 is set HIGH), any write to that channel's configuration register must disable the channel by setting bit 0 LOW.

Errata - Category 3

756021: Corrupted read data when a non-channel register is read immediately following a write to a channel register

Status

Affects: product PL081 DMA Controller -Perpetual.

Fault status: Cat 3, Present in: r1p1-00ltd0,r1p2-00rel0,r1p2-01rel0, Open.

Description

When you perform a read access to a register via the AHB slave interface the HRDATA value is generated from a logical-OR of multiple internal read data buses: one for each channel and one for the non-channel-specific registers.

When you perform a write to a channel-specific register via the AHB slave interface the previous value in that register is driven onto the channel's internal read data bus for a single cycle. If a read to any non-channel-specific register immediately follows the write (i.e. there are no AHB IDLE cycles between the write and the read) then the returned AHB data value will be corrupted because of the value present on the internal bus.

Note that there is no data corruption when a read to a channel-specific register follows a write to a channel-specific register. There is also no data corruption when a read to a non-channel-specific register follows a write to a non-channel-specific register.

The channel-specific registers are: DMACCxSrcAddr, DMACCxDestAddr, DMACCxLLI, DMACCxControl, DMACCxConfiguration (where "x" indicates the channel number). The non-channel-specific registers are all other registers.

Implications

Any read to a non-channel-specific register which immediately follows a write to a channel-specific register (i.e. with no wait states or AHB IDLE cycles) will return a corrupted data value.

Workaround

You must ensure that there is at least one cycle delay between a write to a channel-specific register and a read from a non-channel-specific register.

In a system where only a single master can access the registers, any read from a non-channel-specific register should be performed twice and the first value discarded.

In a system where multiple masters can access the registers, as well as reading each register twice you must ensure that only one master accesses the registers at any one time. You can achieve this by implementing a semaphore function in your software. When a master wishes to read from a non-channel-specific register it must perform the following sequence:

1. Request semaphore

2. Execute first read from register (discard this data)
3. Execute second read from register (keep this data)
4. Release semaphore

Errata - Documentation

320216: Various errors in TRM

Status

Affects: product PL081 DMA Controller -Perpetual.

Fault status: Doc, Present in: r1p1-00ltd0,r1p2-00rel0, Fixed in r1p2-01rel0.

Description

There are a number of incorrect Technical Reference Manual (TRM) descriptions as follows:

(1) Page 3-4: 3.2.7 Programing a DMA channel states:

"DMA channel 0 has the highest priority and DMA channel 7 the lowest priority."

This is incorrect since the PL081 only has 2 channels. It should be:

"DMA channel 0 has the highest priority and DMA channel 1 the lowest priority."

(2) Page 4-5: 4.3.4 Integration Test Output Register 3,DMACITCOP3 defineds TC as [1] and E as [0].

However this is incorrect. It should be: TC as [0] and E as [1]

(3) Page 4-5: states:

"Therefore if both of the TC and E bits are set, then DMACINTR is active."

This is incorrect and should be:

" Therefore if TC or E bits is set, then DMACINTR is active."

(4) P.A-3 "A-2 DMA request and respose signals": Table A-2

The explanation about DMACLCR[15:0] mentions "DMA request acknowledge clear".

This is incorrect. It should be "DMA request clear".

(5) P. B-10 "B.4.5 Memory-to-Peripheral transaction under peripheral flow control:

The TRM mentions: "Othewise,single request are issued on DMACSREQ until the last data item is ready, when DMACLSREQ is used".

The wording seems to have confused a few people. What it is saying:

Basically that DMACSREQ will be used until the last item is due to be transferred where DMACLSREQ will be used.

Another way to look at it would be:

" Otherwise, Until the last data item is ready, when DMACLSREQ is used, single requests are issued using DMACSREQ ".

6) P. B-13 "B.4.8 Peripheral-to-peripheral transaction under destination peripheral flow control" and "B.4.8 Peripheral-to-peripheral transaction under destination peripheral flow control"

>The explanation mentioned : "the single burst request DMACSREQ,if necessary"

The word 'source' is missing off the 'single burst request ....' and so it should say:

"the source burst request DMACSREQ,if necessary"

(7) P.B-14 Figure B-21:

Connective Signal between DMAC and Destination defined as "DMACBREQ,DMACLSREQ,DMACCLR"

This is incorrect and should be: "DMACSREQ,DMACLSREQ,DMACCLR".

Implications

None

Workaround

None

330499: Transfer size count direction in TRM

Status

Affects: product PL081 DMA Controller -Perpetual.

Fault status: Doc, Present in: r1p1-00ltd0,r1p2-00rel0, Fixed in r1p2-01rel0.

Description

The Technical Reference Manual ARM DDI 0218 C states that the transfer size, which is stored in bits 0..11 of a channel control register, counts the number of transfers.

It should be noted that this counts down from the original value to zero and so holds the number of transfers left to complete, NOT the number completed.

Implications

None

Workaround

None

334140: Setting TransferSize in TRM

Status

Affects: product PL081 DMA Controller -Perpetual.

Fault status: Doc, Present in: 1,r1p1-00ltd0,r1p2-00rel0, Fixed in r1p2-01rel0.

Description

Documentation states that TransferSize is not used when the DMAC is not the flow controller.

In fact the TransferSize should be programmed to zero when the DMAC is not the flow controller.

Implications

If the TransferSize is programmed to a non-zero value, the DMAC may attempt incorrectly to use the value.

Workaround

None

364107: Integration Test Output Register 3 bit order in TRM

Status

Affects: product PL081 DMA Controller -Perpetual.

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

Description

In the Technical reference manual (TRM) Section 4.3.4, it states that Bit 0 of the Integration Test Output Register3 corresponds to Err interrupt (E) and Bit 1 corresponds to TC interrupt(TC).

However this is incorrect. It should be: Bit 0 of the of the Integration Test Output Register3 corresponds to TC interrupt (TC) and Bit 1 corresponds to Err interrupt (E).

Implications

none

Workaround

none

Errata – Driver Software

There are no Errata in this Category