FTMCTRL/MCTRL/SDCTRL: SDRAM Initialization Errata

Technical note 2022-10-12

Doc. No GRLIB-TN-0020

Issue 1.1

CHANGE RECORD

Issue	Date	Section / Page	Description
1.0	2022-10-11	All	First issue of document.
1.1	2022-10-12	S5	Expanded footnote to also cover FTMCTRL

TABLE OF CONTENTS

1 INTRODUCTION

1.1 Scope of the Document

This document describes an errata for several SDRAM memory controllers where initialization of external SDRAM may fail, this can in turn lead to SDRAM memory accesses malfunctioning. The document presents a software workaround for the issue.

1.2 Distribution

Users of affected products are free to use the material in this document in their own documents and to redistribute this document. Please contact us for inquires on other distribution.

1.3 Contact

For questions on this document, please contact our support at support@gaisler.com. When requesting support, include the part name if the question is a specific device or the full GRLIB IP library package name if the question relates to a GRLIB IP library license.

2 AFFECTED PRODUCTS

2.1 General

This document applies to the memory controllers of the versions listed below.

- FTMCTRL, before GRLIB build 4102
- MCTRL, before GRLIB build 4080
- SDCTRL, before GRLIB build 4080

2.2 Components

The following CAES components are affected by the errata:

- UT699
- GR712RC
- LEON3FT-RTAX (depends on GRLIB build version and design configuration)

The following CAES components are **not** affected by the errata

- GR716A/B – Not affected, does not implement any SDRAM memory controller
- GR740 – Not affected, does not use an affected SDRAM memory controller
- UT699E – Not affected, makes use of an unaffected version of FTMCTRL
- UT700 – Not affected, makes use of an unaffected version of FTMCTRL

2.3 How to check if a custom design is affected

If you have licensed GRLIB for use in your own FPGA or ASIC design, you can check the following conditions in the design's VHDL source to see if the erratum applies to your system:

1. Check the GRLIB revision. This can be seen in the file name of the downloaded release package, in the directory name after unpacking the release, and in the file “lib/grlib/stdlib/version.vhd” in the release file tree (constant grlib_build).
2. Determine if your design is using affected memory controller IP and has SDRAM support enabled.

In case the VHDL sources are not available, the status of the design can be checked from software as follows:

1. Check the GRLIB revision. This can be read out from the Plug’n’play information on the devices, by default at the lower 16 bits of address 0xFFFFFFF0. The revision is also reported automatically when connecting with GRMON
2. Determine whether you are using an affected memory controller by examining the plug’n’play information.

3 IMPACT

If the recommended workaround is not implemented, affected memory controllers may perform incorrect SDRAM initialization rendering the SDRAM interface non-operational.

If the presented workaround is implemented, the risk is eliminated. The workarounds must be implemented in software that runs before the system starts making use of SDRAM. Typically the workaround needs to be implemented in boot loader software.

Software workarounds have been implemented in the following software packages:

- MKPROM 2.0.68
- GRBOOT 1.5.1
- GRMON 3.2.18

4 DETAILED DESCRIPTION

After power-up, an SDRAM memory device needs to be initialized before it can be used. The detailed requirements of this initialization sequence is typically specified in the memory device datasheet. The specific sequence used in the SDRAM controllers affected by this errata is:

1. One PRECHARGE command
2. Eight AUTO-REFRESH commands
3. One LOAD-MODE-REG command

In between each command above, NOPs are inserted.

The command in the sequence above (LOAD-MODE-REG) is used to set programmable parameters in the mode register of the SDRAM memory device. The contents of this register must be consistent with the configuration of the SDRAM memory controller. Notably, these programmable parameters include CAS latency and burst length. Figure 1 contains a block diagram showing the location of registers whose values need to be compatible for correct SDRAM operation.

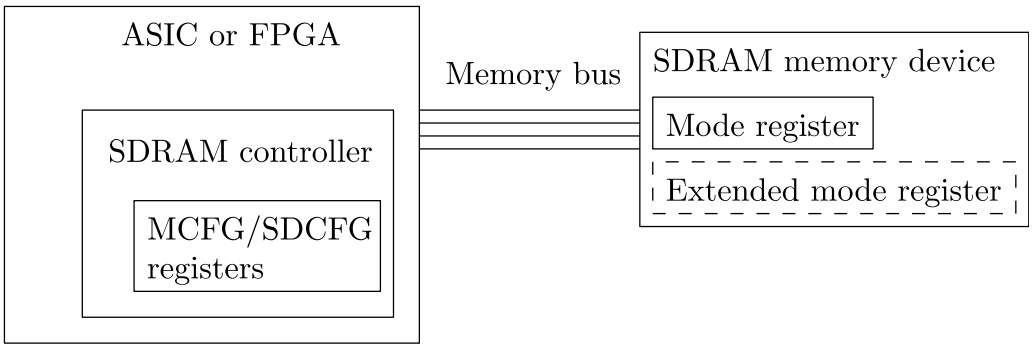

Figure 1: Simplified overview of a system that could be affected by the errata. Note that not all SDRAM memory devices implement an extended mode register.

4.1 Triggering sequence

Incorrect operation of the SDRAM can result if the following conditions are true:

1. The value of the internal memory controller register that is connected to bit 16 of the SDRAM address is ‘1’ at power-up. Address bit 16 corresponds to SDRAM bank address 1 (BA1).
2. The SDRAM mode register of the external SDRAM memory device, contains values incompatible with the intended memory controller configuration register settings at reset/power-up.
3. The SDRAM memory device uses the state of BA1 (address bit 16) to determine if it is

receiving a LOAD-MODE-REG command or not. For example, this is the case for any SDRAM device that implements an extended mode register.

4.2 Errata behaviour

When the errata occurs, the SDRAM memory component will not be initialized with the configuration parameters set in the memory controller. This may lead to a mismatch of settings for parameters such as CAS latency and burst length. Figure 2 below shows an example of an incorrect SDRAM initialization sequence being issued from a GR712RC component.

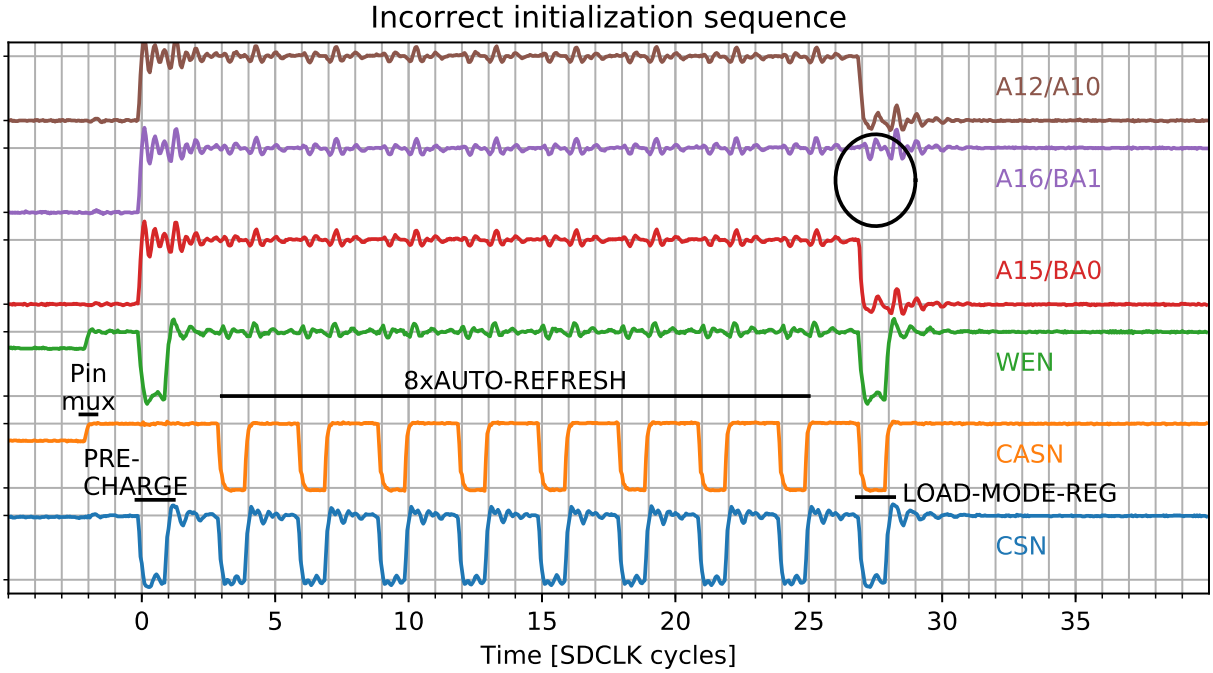

Figure 2: Example oscilloscope measurement of an SDRAM initialization sequence where the issue resulted in the final LOAD-MODE-REG command being incorrect. The traces were recorded on a GR712RC-BOARD.

Specifically, Figure 2 shows oscilloscope traces from a GR712RC where the issue resulted in an incorrect SDRAM initialization sequence. The SDRAM controller is enabled at t = -2. This can be seen as a small edge on the WEN and CASN pins (these pins are multiplexed in the GR712RC and enabling the SDRAM controller changes the mux setting). Then the initialization sequence begins at t = 0 with a PRECHARGE command, followed by 8 AUTO-REFRESH commands and concluding with what should have been a LOAD-MODE-REG command at t = 27. However the issue described in this document caused BA1 to remain at 1 so this is not a valid LOAD-MODE-REG command. External SDRAM memory devices that implement an "extended mode register" in addition to the regular "mode register" would interpret this as a LOAD-EXT-MODE-REG command.

Note that the mode register in the SDRAM memory device may have power-up values matching the intended values, leading to correct operation even if the SDRAM controller memory initialization fails. Tests performed in connection with the development of this technical note indicate that this behaviour of the external memory component may vary between different SDRAM components of the same model from the same manufacturer. Additionally, power-up values may depend sensitively on system startup conditions such as temperature. Thus a system affected by this issue may function correctly at some temperatures, but not others.

4.3 Likelihood of occurrence

The value of the internal register connected to SDRAM address bit 16 is not affected by system reset and will therefore be unpredictable during the first SDRAM initialization sequence after power-up. Furthermore the power-up value of this bit may depend sensitively on the systems environment, such as temperature and power-supply ramp rate. Therefore systems may sporadically pass testing without the errata manifesting. Hence it is recommended that all systems implement the recommended workaround.

The preceding paragraph applies when the system is powered up. But the issue can also occur in a system that is restarted using a soft or hard reset. In that case the value of address bit 16 depends deterministically on the last memory operation made by the SDRAM memory controller prior to the SDRAM controller being disabled. This is the basis for the software workaround described in section 5.

The workaround does not need to be applied in cases where it can be confirmed by the SDRAM memory vendor that the SDRAM configuration will match the intended configuration, even without mode register initialization.

5 WORKAROUND

A correct SDRAM initialization sequence can be attained by initializing the internal memory controller register that drives the address pin corresponding to SDRAM bank address 1 (BA1, address bit 16). The following sequence will initialize this register and initialize the SDRAM with the values configured in the memory controller register interface:

1. Enable the SDRAM controller.[1]
2. Wait for the SDRAM initialization sequence to complete.
3. Write any value to the first SDRAM address.
4. Disable the SDRAM controller.
5. Re-enable the SDRAM controller.

This sequence (specifically step 3) guarantees that address bit 16 is 0 during the LOAD-MODE-REG command executed in step 5.

Note that the state of address bit 16/BA1 in step 1 is unpredictable, so it is unkown whether the initialization in this step succeeded or not. Hence the value written in step 3 may not have been stored in the SDRAM.

Alternatively, the write in step 3 may be replaced with a read from the first SDRAM address. However, for memory controllers implementing EDAC, a write to uninitialized memory is preferred over reads since a read could result in an EDAC error.

1 The SDCTRL SDRAM controller is enabled/disabled through the SDRAM refresh (RF) bit in the SDRAM configuration register 1. The FTMCTRL SDRAM controller (used in UT699 and GR712RC) is enabled/disabled through the SDRAM enable (SE) bit in the MCFG2 register.

6 ROOT CAUSE DETAILS

The SDRAM controllers affected by this issue contain an internal register that drives the address bus during SDRAM operations. However, due to a design error, bit 16 of this register is not updated during execution of a LOAD-MODE-REG command, but instead retains its previous value. Additionally this internal address register is not affected by reset and therefore contains undefined values after power-up. This is illustrated in Figure 3 that shows the state of the SDRAM memory bus in an RTL simulation of an affected memory controller. Undefined values (U) are shown in red.

The issue was not detected during memory controller IP core verification due to the SDRAM simulation models used not caring about the uninitialized register value.

The issue was not detected during FPGA verification due to the register being initialized to zero in FPGA prototypes.

The issue has gone undetected in applications with UT699 and GR712RC. Our assumption is that this is due to the majority of applications not complying with all of conditions 1-3 from section 4.1. Namely, in any given application, at least one of the following has been true:

1. The value of bit 16 the internal SDRAM controller address register been 0 at power-up.
2. Or, the power-up state of the mode register in the external SDRAM memory device has matched memory the controller configuration.
3. Or, the particular SDRAM memory devices used have not taken the state of BA1 into account when determining if a LOAD-MODE-REG command is being issued.

The issue was identified as part of debugging failing SDRAM operation on a subset of GR712RC development boards where a range of different SDRAM modules are used.

The issue was corrected in the MCTRL and SDCTRL memory controllers as a side-effect of adding support for Mobile SDRAM. The issue was corrected in FTMCTRL as part of updates in 2010.

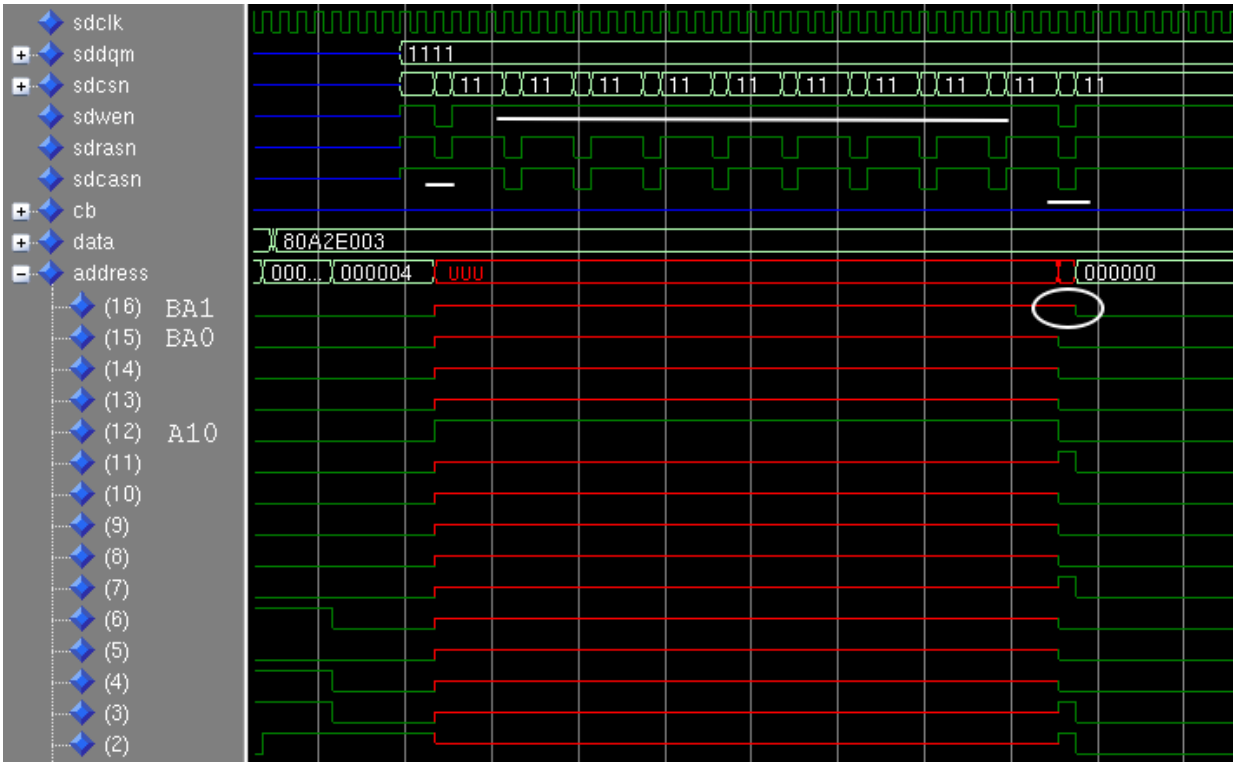

Figure 3: RTL simulated view of the external memory bus of an affected SDRAM controller during the first initialization sequence after simulation start (equivalent to power-up). Compare Figure 2.