FTMCTRL: Memory corruption from accesses to SDRAM and 8-bit SRAM with EDAC

Technical Note

2025-03-04

Doc. No GRLIB-TN-0023

Issue 1.0

CHANGE RECORD

Issue	Date	Section / Page	Description
1.0	2025-03-04	All	First issue

TABLE OF CONTENTS

1 OVERVIEW

1.1 Scope

This document describes behaviour of the FTMCTRL memory controller where a sub-word write to SRAM can lead to EDAC error or silent data corruption. The case can be triggered when SDRAM and 8-bit SRAM with EDAC are simultaneously enabled.

1.2 Affected versions

The issue affects all versions of the FTMCTRL IP core.

The FTMCTRL IP core documentation has been updated to notify users about this issue starting with release GRLIB 2025.1.

1.3 Affected components

The following Frontgrade components are affected:

- GR712RC
- LEON3FT-RTAX
- UT699
- UT699E
- UT700

The GR740 is not affected since FTMCTRL is only utilized to provide PROM and IO access in this component. The GR716 components are not affected since they do not support SDRAM.

1.4 Distribution and responsibilities

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.

Users of the affected products are responsible to thoroughly review this document before undertaking the described mitigation actions and to adhere to the provided instructions.

The latest version of this document is available from https://www.gaisler.com/notes

1.5 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 TECHNICAL DESCRIPTION

The FTMCTRL combined 8/16/32-bit memory controller provides a bridge between the on-chip AHB bus and external memory. The memory controller can handle four types of devices: PROM, asynchronous static ram (SRAM), synchronous dynamic ram (SDRAM) and memory mapped I/O devices (IO). The PROM, SRAM and SDRAM areas can be EDAC-protected.

The SRAM area in the memory controller can be connected to other types of memory than SRAM. The description below talks about SRAM but the exact memory technology of the external device connected to the memory controller's SRAM signals is not relevant for the behaviour described in this technical note.

When the SRAM is configured in 8-bit mode and EDAC is enabled, a read access will automatically read the four data bytes individually from the nominal addresses and the EDAC checkbit byte from the top part of the memory bank. A write cycle is performed the same way. Byte or half-word write accesses will result in an automatic read-modify-write access where 4 data bytes and the checkbit byte are firstly read, and then 4 data bytes and the newly calculated checkbit byte are written back to the memory.

When SDRAM and 8-bit SRAM with EDAC are simultaneously enabled, there is a condition where a sub-word (byte or half-word, 8- or 16-bit) write to SRAM that is in between two SDRAM accesses will be performed incorrectly. The logic that calculates the least significant address bits for the SRAM access uses the current state of the AHB bus instead of a previously latched state. This results in the read-modify-write sequence to SRAM starting from the wrong address, reading too few bytes, and then EDAC decoding results in an uncorrectable error with high probability (or silent memory corruption with a smaller probability).

To trigger the case, the sub-word write to SRAM must occur in between two SDRAM accesses where the second SDRAM access must be a read operation. The required sequence is:

- Access to SDRAM area (read or write)
- Sub-word write access to SRAM area
- Read access to SDRAM area

The above sequence has been simplified. The case where the read-modify-write sequence starts from the wrong address also depends on the size of the second access to the SDRAM area. There is also a minimum number of cycles required between accesses (the case always triggers for back-to-back accesses) that is not further detailed in this document.

3 FUNCTIONAL IMPACT

A sub-word write operation under the described condition will either lead to an EDAC decoding error or a silent memory corruption. An EDAC decoding error will lead to the write operation receiving an AMBA ERROR response. This event can be detected by peripherals such as the AHBSTAT AHB status register. A processor core will, unless it has been configured to ignore write errors, raise a data store exception. Communication controllers will stop upon receiving the ERROR response.

In the case where the too few bytes read does not trigger an EDAC error then the result can be incorrect data stored to the external SRAM, leading to silent data corruption.

4 WORKAROUND / MITIGATION

There are three possible workarounds that allow operation of 8-bit wide SRAM at the same time as SDRAM is used:

1. Disable RAM EDAC (controlled by register field MCFG3.RE that controls EDAC for SRAM and SDRAM area), or
2. Avoid simultaneous accesses to SRAM and SDRAM, or
3. Do not perform 8- and 16-bit write operations to SRAM

Additional details are provided for each workaround in the subsections below.

4.1 Disable RAM EDAC

Disabling RAM EDAC will disable EDAC for both the SRAM and SDRAM memory area and is unlikely to be a practical workaround.

4.2 Avoid simultaneous accesses to SRAM and SDRAM

Avoiding simultaneous accesses to SRAM and SDRAM is difficult to achieve if there is more than one processor core active, or if communication controllers capable of DMA are active, and there is a need for these entities to work with information both in SRAM and SDRAM.

In the case with only one processor core, and no active DMA from other entities in the system, it may be possible to generate the SDRAM access, sub-word SRAM write, and SDRAM read access depending on the processor core behaviour. At the time of writing, it has not been confirmed whether a single LEON3FT processor is able to generate this sequence.

4.3 Do not perform 8- and 16-bit write operations to SRAM

This workaround means that the only write accesses that shall be issued to the SRAM area are 32-bit (word) accesses.

Use of the SRAM to hold data buffers for communication controllers need to be avoided as the communication controllers may issue 8- and -16-bit write accesses for unaligned buffers and for buffers where the data length is not a multiple of four bytes.

Processor accesses to the SRAM should be made through an API that is verified to only generate 32-bit accesses for writes. Note that this is beneficial from a performance perspective since it avoids the overhead of the read-modify-write operation. Instruction fetches can be performed from SRAM.