LEON3/FT AHB Deadlock After Sequence of Load and Atomic Instructions

---

Technical note

Doc. No GRLIB-TN-0010

Issue 1.3

2017-12-21

CHANGE RECORD

Issue	Date	Section / Page	Description
1.2	2017-11-17		First public release.
1.3	2017-12-21	Section 3.2	Overview of compiler workaround and list of toolchains with the workaround available.

TABLE OF CONTENTS

1 OVERVIEW

1.1 Scope of the Document

This document describes a corner case that affects specific versions of the LEON3/FT processor core that can cause the LEON3/FT core to hang while executing an atomic instruction directly after a load instruction. The corner case requires a number of additional conditions to be met and the processor's memory management unit must be enabled.

1.2 Affected versions

The issue described in this document is present in all LEON3/FT cores implemented with memory management unit (MMU) since GRLIB build b4108 up to b4204.

1.3 Affected components

Cobham and Aeroflex components that are affected are:

- UT699E
- UT700
- LEON3FT-RTAX: only for CID5/CID6 with GRLIB version between b4108-b4204

Cobham and Aeroflex components that are unaffected are:

- UT699
- GR712RC
- GR740 – LEON4/FT is unaffected
- LEON3FT-RTAX: CID1/CID2/CID3/CID4/CID7/CID8 (all GRLIB versions) or CID5/CID6 with GRLIB version smaller than b4108 or higher than b4204.

1.4 How to check if a LEON3/FT design is affected

The GRLIB build ID is present in the AMBA plug&play area. The build ID is also reported by the GRMON debug monitor when connecting to the device.

If you have licensed GRLIB for use in your own FPGA or ASIC design, the GRLIB version 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).

If the GRLIB build ID is smaller than 4108 or higher than b4204 then the GRLIB version is **not** affected.

1.5 Distribution

Users that have designs affected by the issue described in this document are free to use the material in their own documentation. Contact Cobham Gaisler for inquires on other distribution.

1.6 Contact

For questions on this document, please contact Cobham Gaisler 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 corner case described in this document can cause a LEON3/FT core to hang and hold the AHB bus locked until the processor receives a hardware reset. This condition will only be encountered if **all** of the following conditions are met:

1. MMU is enabled.
2. A load instruction (any type) is followed by an atomic instruction (swap, ldstub, casa) without any intervening instruction. The program counter values of the load instruction and the atomic instruction can be consecutive but it is not a requirement. If a PC-relative control-transfer instruction with a load instruction in the delay slot branches to an atomic instruction then this condition will be satisfied. The following two examples illustrates two different occurrences of this condition being met:

   Example 1:

   ```
   PC      INST
   0x04    ld [%l1],%l2
   0x08    swap [%l4],%l3
   ```

   Example 2:

   ```
            PC      INST
   $L0:     0x0C    swap [%l4],%l3
            …
            …
            0xE0    bne $L0
            0xE4    ld [%l1],%l2
   ```

3. The load instruction before the atomic instruction must cause a data cache miss when the processor's store buffer is full.
4. The atomic instruction after load instruction must cause a data translation lookaside buffer (DTLB) miss.

Note:

- A PC-relative control-transfer instruction includes branch on integer condition codes instructions (see SPARC Architecture Manual Version 8, section B.21), Branch on floating-point condition code instructions (see SPARC Architecture Manual Version 8, section B.22) and CALL instruction (see SPARC Architecture Manual Version 8, section B.24).
- The CALL instruction mentioned in the previous point only includes the real CALL instruction (CALL label). The pseudo CALL instruction (call with register relative addressing) does not cause the behaviour to be triggered.

If all the previously defined conditions are met, the processor will hang and the AHB bus will remain locked until the processor receives a hardware reset.

3 WORKAROUND / MITIGATION

3.1 Workaround

The corner case can be avoided by avoiding execution of a load instruction and an atomic instruction consecutively. This can be done by inserting a NOP operation in-between a load instruction and an atomic instruction. If an atomic instruction is a target of a branch, an additional NOP instruction can be inserted before it without any precondition in order to avoid complex analysis.

3.2 Toolchain versions with workaround

Errata workarounds are available for BCC, RCC, and VxWorks. The workarounds include compiler fixes as well as patched assembly code for the system libraries. The following release versions and later will include the workarounds in toolchain and OS/libraries:

- BCC 1.0.50
- BCC 2.0.2
- RCC 1.2.22
- RCC 1.3-RC3
- VxWorks 6.7 1.0.21 – Toolchain 1.0.15
- VxWorks 6.9 2.0.8 – Toolchain 1.0.4
- VxWorks 7 – Toolchain 7.2.0.0
- MKPROM 2.0.63

The compiler workaround will scan each function for instruction sequences that could potentially trigger the errata. If it finds a load instruction followed by an atomic instruction (SWAP, LDSTUB, or CASA) it will insert a NOP instruction between them. If it detects a branch to an atomic instruction it will insert a NOP instruction at the branch target. This is in accordance with the workaround described in the previous section. The VxWorks 6.7 and BCC 1 compilers does not generate atomic instructions and does not need the compiler workaround.

Note that the compiler workaround will not make any modifications to inline assembly code in the user application. Any sensitive assembly instruction sequence will have to be modified manually. Note also that BCC and RCC by default does not use the MMU, which means that the errata can not trigger unless the MMU is explicitly enabled.

The compiler workaround is activated by compiling with the "-mfix-ut700" flag.

The patches for the compiler workaround are listed below. They are included in the official GCC distribution for version 7.3.0 and later.

[SPARC] Errata workaround for GRLIB-TN-0010
https://gcc.gnu.org/viewcvs/gcc?view=revision&revision=255236
* config/sparc/sparc.c (sparc_do_work_around_errata): Use mem_ref
https://gcc.gnu.org/viewcvs/gcc?view=revision&revision=255393
SPARC: Make sure that jump is to a label in errata workaround
https://gcc.gnu.org/viewcvs/gcc?view=revision&revision=255807

3.3 Scan script

A scan script is provided to allow scans of disassembled programs for code that can trigger the issue described in this document. The script is available from:

http://www.gaisler.com/notes

Please see the header in the script file for usage instructions.