FTMCTRL: BCH EDAC with multiple 8-bit wide PROM and SRAM banks

---

Application note 2017-04-21

Doc. No GRLIB-AN-0003

Issue 1.0

CHANGE RECORD

Issue	Date	Section / Page	Description
1.0	2017-04-21		First issue

TABLE OF CONTENTS

1 INTRODUCTION

1.1 Scope of the Document

This application note describes the BCH checkbits code storage scheme in 8-bit memory bus mode of the FTMCTRL Fault-Tolerant Memory Controller IP core. The memory controller generates additional bus cycles to access checkbits data in a dedicated part of the memory device. The way the address of checkbits data is generated imposes some constraints on the device size when used in multiple banks in certain configurations.

1.2 Reference Documents

[RD1] GRLIB IP core User's manual: http://gaisler.com/products/grlib/grip.pdf

2 ABBREVIATIONS

EDAC	Error Detection And Correction
BCH	Bose–Chaudhuri–Hocquenghem
CB	Check-Bits
AMBA	Advanced Microcontroller Bus Architecture

3 BCH EDAC AND BUS WIDTH

BCH EDAC requires 7 bits of checkbit data per 32-bit data word. On memory buses with 32-bit data width, a separate CB[] data bus is used to transfer checkbits simultaneously with data to and from the memory device. The memory must be of total width of 39 bit at least. On memory buses with support for 16-bit data width, EDAC is not supported. On buses configured to 8-bit data width, only the most significant bits D[31..24] out of the 32 physical data lines present in most implementations are connected to the memory device. 32-bit AMBA bus accesses are broken down into four sequential 8-bit memory accesses. If EDAC is enabled, a fifth access is added by the memory controller to access checkbits. The checkbits are located in the upper fifth of each memory bank, in descending order. For the first data word at address 0x0, the corresponding checkbits byte is located at the last byte address of the same bank. For the second data word at address 0x4, the corresponding checkbits byte is located at the last but one byte address of the same bank, and so on.

Software must take care to never allocate user data or code in the memory space allocated for the checkbits. Reading and writing checkbits is done automatically by the FTMCTRL hardware in the RAM area. Reading, but not writing, checkbits is done automatically by the FTMCTRL hardware in the PROM area.

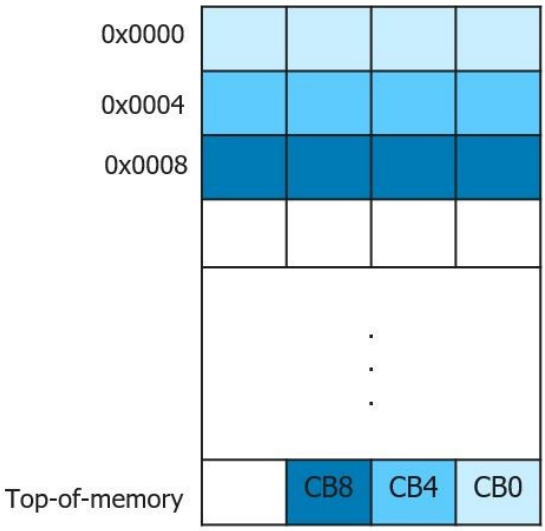

Figure 1 Corresponding data and checkbits in one memory bank

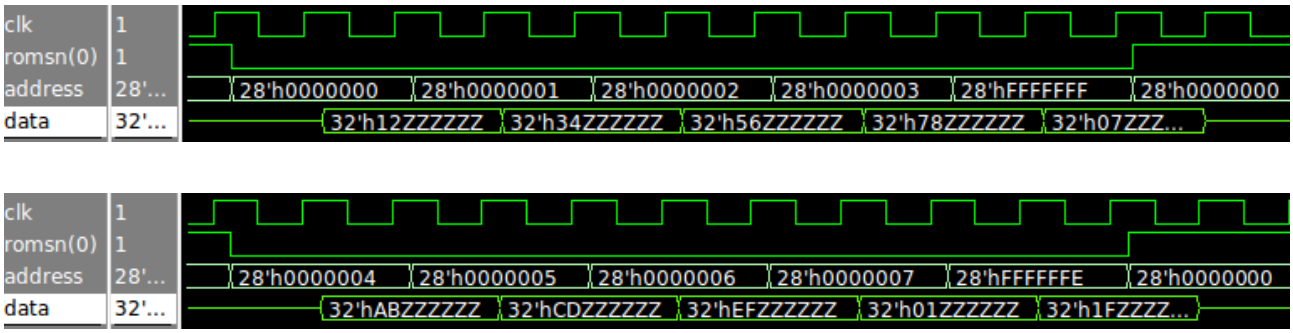

Figure 2 Transfer of data 0x12345678 and 0xABCDEF01 from memory addresses 0x0 and 0x4, including checkbits

The generation of the address of the checkbits location is described in [RD1] as follows:

> The corresponding checkbits are located at the address acquired by inverting the word address (bits 2 to 27) and using it as a byte address.

Figure 3 Shows the principle of CB address generation for an arbitrary bank size of 16 byte. The Bank Size fields in the FTMCTRL configuration registers select the two high address bits used for chip select decoding. It is indicated as a pointer in Figure 3 . The formula given in [RD1] for bank size calculation, for example "8KiB * $2^{ROMBANKSZ}$", reflects a certain fixed offset (8KiB, i.e. address bit 13), plus ROMBANKSZ as variable for the selection of address lines for chip select decoding.

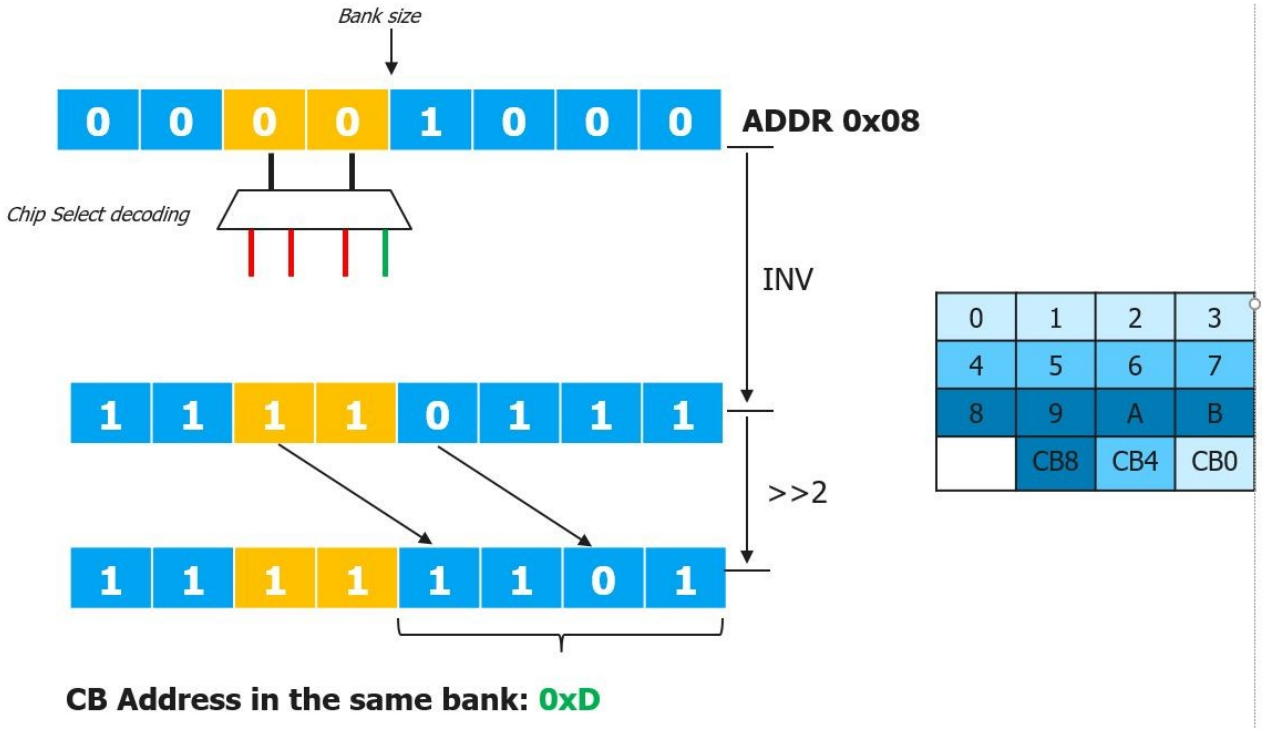

Figure 3 Generation of CB address

"`Using [the word address] as a byte address`" means in other words shifting the address by two to the right. Checkbits are always located in the same bank of the corresponding data word. In other words, the chip select decoding is done before the generation of CB addresses and kept constant for the same access.

4 MULTIPLE MEMORY BANKS

When multiple SRAM or PROM memory banks are used in 8-bit mode with EDAC, the described address generation scheme for CB can impose a memory configuration that seems counter-intuitive: **For all PROM or SRAM bank sizes except the two largest possible (256 MiB and 128 MiB), the configured bank size must be set to at least four times the actual device size.** For both device sizes of 256 MiB and 128 MiB, the 256 MiB bank size setting must be used.

To illustrate the impact of a device size equal to the bank size, Figure 4 shows an example of two banks of an arbitrary size of 16 bytes.

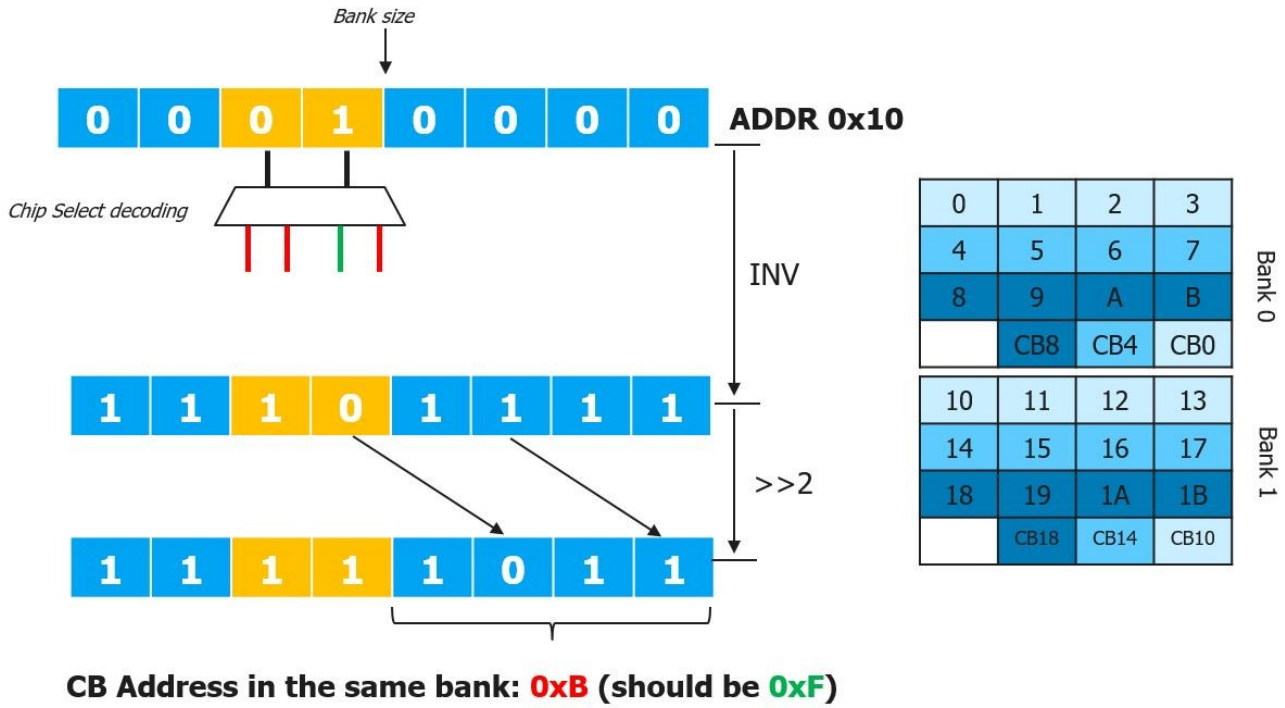

Figure 4 Generation of CB address with two banks

The two bits used for chip select decoding are part of the generated CB address and can fold the CB address into the data space. If the bank size is however four times the actual device size, the bits used for chip select decoding are located two positions to the left, and cannot be shifted into the used part of the bank address, as in Figure 5:

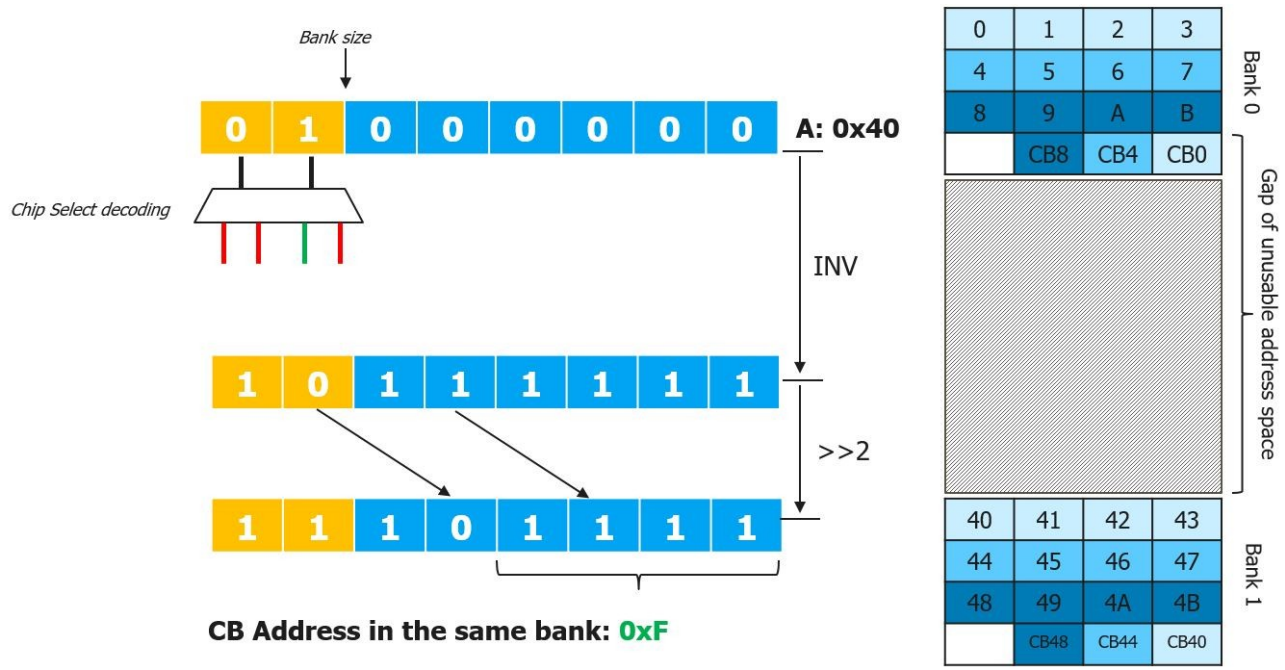

Figure 5 Generation of CB address with increased bank size

The largest bank size setting (256 MiB) represents a special case, since the actual CB address generation from the data address `ADDRESS` is as follows:

```
ADDRESS(31 downto 28) & "11" & not ADDRESS(27 downto 2)
```

For this bank size (or in other words, when the "bank size pointer" points to `ADDRESS(28)`), effectively a constant "`11`" is shifted into the CB bank address, instead of the (inverted) address bits used for CS decoding. The device size can therefore be equal to the bank size indicated by or set in the bank size configuration register field.

5 FTMCTRL IMPLEMENTATIONS AND THE PROM AREA

The FTMCTRL IP core, part of the GRLIB IP core library, can be implemented with a number of different features through hardware configuration options (VHDL generics) at implementation time. Some of those options make a choice to have features disabled, enabled, enabled with a fixed setting, or enabled with a setting programmable through configuration registers at run-time. Configuration options relevant to this application note are:

- `romasel`: Sets the PROM bank size. Depending on the `romasel` value, the bank size can be fixed or programmable through configuration registers. The PROM bank size value in the configuration register always reflects the actual bank size.
- `ram8`: Enables support for 8-bit RAM and PROM. The actual bus width is programmable through the configuration registers. This application note applies only when this option is enabled
- `edac`: Enables support for EDAC with different codes and correction capabilities. Actual EDAC protection is enabled through the configuration registers separately for the PROM, RAM and SDRAM areas. The PROM and RAM areas only supports BCH EDAC. This application note applies only when this option is enabled.

The different bank sizes resulting from different `romasel` settings are documented in [RD1], chapter FTMCTRL, Configuration Options. For all configurations, the bank size requirement as described in chapter 4 applies, unless the Bank Size field in the configuration register reads "`0000`" or "`1111`" (the maximum size of 256 MiB).

NOTE: It has been documented in the past that for `romasel` settings that result in two chip-select signals, or two PROM banks, the bank size requirement as described in chapter 4 does not apply. This could not be confirmed at the time of writing of this application note. The requirement in fact applies to all use-cases where more than one PROM bank is configured and used, except if used with the maximum bank size as above.

For implementations with software programmable PROM bank sizes, the reset value is always "`0000`", the maximum size, where the device size can be up to the indicated bank size.

Beyond the immediate scope of this application note, it should be noted that on the PROM area, the consecutive access of several 8-bit transfers per 32-bit AMBA request includes the fifth CB byte only for reads. 32-bit AMBA write requests to the PROM area will translate only into four 8-bit data transfers, even with EDAC enabled. Many non-volatile technologies are not compatible with those consecutive transfers in write mode, and it is hence recommended to perform all write

accesses to the PROM area with 8-bit AMBA transfers (`stb` instructions). The BCH checkbits must be calculated by software and stored to the appropriate address.

6 FREQUENTLY ASKED QUESTIONS

6.1 Affected devices

Q: Is the device I am using affected? Do I need to set a larger PROM or RAM bank size, or use a smaller memory device, in 8-bit mode with EDAC and multiple memory banks?

A:

- **GR712RC**: PROM and RAM bank size are programmable and must be set to four times the device size under above conditions. Maximum bank size (256 MiB) does not apply since the available external address lines on the device are limited to ADDRESS[23:0], i.e. 16 MiB are addressable per bank.
- **UT699/UT699E/UT700**: PROM bank size is fixed to maximum bank size (256 MiB). Devices up to the same size can be used. RAM bank size is programmable and must be set to four times the device size under above conditions, except for the maximum bank size setting (256 MiB).

  Current versions of the UT699 user manual state "*When the EDAC is enabled in 8-bit bus mode, only the first bank select RAMS[0], ROMS[0] can be used*". This addresses the same limitation by a more restrictive requirement.
- **LEON3FT-RTAX**: Standard configurations IC1, IC2 and SC1 of this component do not contain a FTMCTRL memory controller with the described EDAC checkbits storage scheme.

  For standard configurations SC2, SC3, SC4, PC1, PC2, PROM and RAM bank size are programmable and must be set to four times the device size under above conditions, except for the maximum bank size setting (256 MiB).

  However, variations of the standard configurations exist and it should always be verified whether FTMCTRL is used in the component.
- **GR740**: PROM bank size is programmable and must be set to four times the device size under above conditions. For the maximum bank size setting (256 MiB), only one chip select is available since the area of the second bank is not within the PROM area. The device has no controller for RAM area (only SDRAM).

  Current versions of the GR740 user manual state "*When the EDAC is enabled in 8-bit bus mode, only the first bank select PROM_CEN[0] can be used*". This addresses the same limitation by a more restrictive requirement.

6.2 Future versions

Q: Will future versions of the FTMCTRL IP core remove the bank size requirement?

A: No change of the FTMCTRL IP core addressing the bank size requirement is planned.

Even though the requirement of a bank size larger than the actual device has been published as “erratum” in the past, it represents rather an inherent limitation of the chosen addressing scheme. The limitation could be avoided by shifting in “11” at the position of the (variable) address line defining chip select decoding, instead of choosing the (fixed) position of the highest possible address line. However, compared to adding complexity to an IP core which is consolidated in design and implementation characteristics (synthesis) for a long time, the impact of the limitation is considered low:

The impact on software design is that a larger gap of unusable memory must be implemented in linker scripts. However, a gap must be implemented either way in order to protect the area occupied by the checkbits. The location of the borders of that gap is a mere detail of that implementation.

There is no impact on hardware design. All memory sizes can be implemented. For device sizes of 256 MiB, 128 MiB and 64 MiB, the 256 MiB bank size setting is suitable.