ECC Error Handling Guide

Version 1.1

ECC Error Handling Guide

Proprietary Notice

Release Information

Version	Date	Changes
Version 1.0	May 1, 2019	• Initial release
Version 1.1	Jan 29, 2020	• Updated BEU table with default values

Contents

Chapter 1

SiFive ECC Error Detection & Correction Operation

1.1 What is ECC

ECC (Error Correction Codes) is a hardware mechanism that detects and in some cases corrects defects in memory. On SiFive designs, SRAM blocks that optionally support ECC are instruction cache, Instruction Tightly Integrated Memory (ITIM), Data Tightly Integrated Memory (DTIM), L1 cache, and L2 data cache. The simplest case is a single bit error that is detected, reported via interrupt handler, and corrected automatically by hardware without any software intervention. More difficult scenarios involve double or multi-bit errors that are still reported and tracked in hardware but are not correctable. The ECC hardware includes logic for detection and correction, in addition to 7 redundant bits per 32-bit codeword or 8 redundant bits per 64-bit codeword.

1.2 ECC Configuration

The Bus Error Unit (BEU) configuration registers are used to enable error reporting, and optionally interrupt reporting. Interrupts can be handled on a system level using the Platform Level Interrupt Controller (PLIC), or locally, directly to an individual CPU.

The BEU register memory map is shown below:

Offset	Size	Attr	Register	Default Reset Value	Notes
0x000	1B	RW	cause	0x0	Cause of error event
0x008	XLEN	RW	value	Not assigned	Physical address of error event
0x010	1B	RW	enable	0x20	Event enable mask
0x018	1B	RW	plic_interrupt	0x0	Platform-level interrupt-enable mask
0x020	1B	RW	accrued	0x0	Accrued event mask

Table 1: BEU Registers

The BEU `enable` bitfields are mask values used to enable reporting of each event by the BEU. To enable reporting, write a `0x1` in the correct bitfield location to allow events to be reported via the `cause` register.

> **Note**
>
> The `enable` register default reset value is `0x20` which enables TileLink bus errors.

```
[0] No Error
[1] Reserved
[2] Instruction cache or ITIM correctable ECC error
[3] ITIM uncorrectable ECC error
[4] Reserved
[5] Load or store TileLink bus error
[6] Data cache correctable ECC error
[7] Data cache uncorrectable ECC error
```

The `cause` register will contain the most recent event recorded by the BEU. For example, a value of `3` represents an ITIM uncorrectable ECC error. A value of `0` means no error. The `cause` register will only be updated for enabled bits in the `enable` register. If multiple events occur then `cause` will report the first one latched since the last time it was cleared by software.

The BEU register `accrued` also has the same bitfield description as `enable` and is used to track errors as they are received. The BEU sets bits in the `accrued` register whether or not they are enabled in the `enable` register. The `accrued` register can be used to monitor ECC events which are not setup to be reported via the BEU.

The `value` register supplies the physical address that caused the event, or 0 if the address is unknown. The BEU will only update the `value` register when the `cause` register is 0.

1.2.1 ECC Initialization

Any SRAM block or cache memory containing ECC functionality needs to be initialized prior to use. ECC will correct defective bits based on memory contents, so if memory is not first initial-

ized to a known state, then ECC will not operate as expected. It is recommended to use a DMA, if available, to write the entire SRAM or cache to zeros prior to enabling ECC reporting. If no DMA is present, use store instructions issued from the processor. Initializing memory with ECC from an external bus is not recommended. After initialization, ECC related registers, such as the Bus Error Unit `cause` register, can be written to zero, and then ECC reporting can be enabled. For 32-bit architectures, 32-bit aligned writes are recommended. Simililarly, for 64-bit architectures, 64-bit aligned writes are recommended.

An example procedure using the DMA is described below.

```
li t0 DMA_CTRL_ADDR + 0x80000 // channel 0
li t1, 1
sw t1, 0(t0)  // claim channel 0
li t1, SIZE_OF_TRANSFER
sd t1, 8(t0)  // bytes
li t1, DEST_BASE_ADDR
sd t1, 16(t0) // dest
li t1, CACHEABLE_ZERO_MEM_ADDR
sd t1, 24(t0) // src
li t1, 0xff000000 //rsize and wsize
sw t1, 4(t0)  // full speed copy
li t1, 3
sw t1, 0(t0)  // start transfer
1: // wait for completion
lw t1, 0(t0)
andi t1, t1, 2
bnez t1, 1b

// release DMA
sw zero, 0(t0)
```

1.3 ECC Interrupt Handling and Error Injection

The code running within the interrupt handler is not responsible for repairing single bit errors, as this is done automatically by hardware.

The `plic_interrupt` register indicates which `accrued` events should generate an interrupt to the PLIC. An interrupt is generated when any bit is set in both `accrued` and `plic_interrupt`, i.e., when (`accrued` & `plic_interrupt`) != 0.

The `local_interrupt` register indicates which `accrued` events should generate an interrupt directly to the hart associated with this bus-error unit. An interrupt is generated when any bit is set in both `accrued` and `local_interrupt`, i.e., when (`accrued` & `local_interrupt`) != 0.

BEU errors are always enabled and thus do not have a control bit in `mie` (machine interrupt enable) CSR. Likewise, there is no dedicated control bit for BEU errors in the `mideleg` (machine interrupt delegation register) CSR, so it cannot be delegated to a mode less privileged than M-mode.

Monitoring overall ECC events can be accomplished in software via the interrupt handler. If the design utilizes L2 cache, the L2 cache controller contains hardware counters to track ECC events, and optionally inject ECC errors to test the software handling of ECC events. Consult the L2 manual for more information. If the design does not have an L2 cache controller, error injection and thus software handling of errors can be accomplished manually by writing the `cause` register.

The exception code value, located in `mcause` (machine trap cause) CSR, will be 11 (0xB) when BEU interrupts are routed through the PLIC. The exception code value is independent of the PLIC interrupt number used to connect the BEU to the PLIC. When ECC interrupts are routed locally, the default exception code value will be 128 (0x80). Prior to software development, consult the User Manual for your design to confirm the exception code.

1.4 Hardware Operation Upon ECC Error

Harware will operate differently depending on which memory type encounters an ECC errror:

- **Instruction Cache:** The error is corrected and the cache line is flushed
- **Data Cache:** The error is corrected, the cache line is invalidated and written back to the next level of memory
- **ITIM, DTIM:** Single bit errors are corrected and written back to the RAM. Errors will toggle the `err_from_tile_X signal` for system level tracking
- **L2:** Single bit correction for L2 data and meta-data (meta-data includes index, tag, and directory information). Double bit detection only on L2 data array.

Double bit errors are reported at the Core Complex boundary via the signal: `halt_from_tile_X`. This signal, if asserted, remains high until reset.