

GSI ECCRAMs™

The Benefits of On-Chip ECC

Introduction

Error Correction Code, or ECC, is commonly utilized with SRAMs in applications where data corruption via SER events is not easily tolerated. Typically, the ECC algorithms detect and correct single-bit data errors. In some instances they can also detect multi-bit errors, depending on the type of algorithm used and the number of parity bits allocated to ECC.

Traditionally, in environments that require high data integrity, the ECC error detection and correction algorithms have been implemented in custom ASIC- and FPGA-based memory controllers. However, it has become increasingly rare for such controllers to be designed in-house, due to cost, resource, and time-to-market constraints. Unfortunately, the availability of 3rd party, off-the-shelf SRAM controllers designed for the commercial market is quite limited, and those that are available often do not support ECC functionality, leaving SER-sensitive SRAM users in a difficult situation.

Bringing the ECC on-chip resolves the issue by removing the burden from the controller, thereby simplifying custom controller design and maximizing 3rd party controller options for the application. It also provides utilization efficiency benefits that are explained further below.

Accordingly, GSI Technology has developed a family of SigmaQuad™ and SigmaDDR™ SRAMs with on-chip ECC, referred to collectively as “ECCRAMs”.

Implementation

GSI ECCRAMs utilize a single-bit error detection and correction *Hamming Code* algorithm. The ECC is implemented independently on each external 9-bit data bus, across the entire 18-bit DDR data word transmitted on the bus. For example, x36 devices have four such 9-bit busses, and x18 devices have two such 9-bit busses.

Five ECC parity bits (invisible to the user) are utilized per 18 data bits (visible to the user). Consequently, 72Mb ECCRAMs (for example) are actually 92Mb devices, with 72Mb visible and available to the user.

The ECC algorithm neither corrects nor detects multi-bit errors. However, the ECCRAMs are architected in such a way that a single SER event *very* rarely causes a multi-bit error across any given “data word”, where a “data word” in this context represents the data transmitted as the result of a single read or write operation to a particular address.

The ECC implementation is entirely transparent to the user. Type II and II+ ECCRAMs are fully compatible with other Type II and II+ SigmaQuad and SigmaDDR SRAMs, except with respect to Byte Write support. See *Byte Write Implications* below for further information.

Applications

The primary applications for ECCRAMs are those that demand a very high level of dependability, such as military and other data-critical applications, as well as those that are more susceptible to SER events, such as aerospace, satellite, and other applications expected to be utilized at high-altitude.

Benefits

- Virtually Zero Soft Error Rate (SER)
Accelerated SER testing has been conducted on 72Mb ECCRAMs at the LANSCE WNR facility in Los Alamos, NM. The following table presents the nominal cosmic ray FIT for each event type (SBU, MCU, SEFI, SEL) for the tested devices at sea level New York City. FIT rate values are per Mbit per 10⁹ hours, and represent 95% confidence level.

Cosmic Ray FIT Values at Sea Level, New York City

Test Condition	SEU ¹ FIT	SBU ² FIT	MCU ³ FIT	SEFI ⁴ FIT	SEL ⁵ FIT
ECC Disabled	567	305	262	0	0
ECC Enabled	0	0	0	0	0

Notes:

1. SEU = Single-Event Upset (SBU + MCU)
2. SBU = Single-Bit Upset
3. MCU = Multi-Cell Upset
4. SEFI = Single-Event Functional Interrupt
5. SEL = Single-Event Latch-up

With ECC disabled, the total SEU FIT rate was 567, comprising an SBU FIT rate of 305 and a MCU FIT rate of 262.

With ECC enabled, the total SEU FIT rate decreased to 0, indicating that *all* SBUs and MCUs that occurred during the testing were corrected by the ECC. Which indicates that none of the MCUs that occurred during the testing

with ECC enabled resulted in “MBUs” (i.e., multi-bit upsets in the same data word); if they had, the SEU FIT rate would have been > 0 because the ECC would have been unable to correct them.

However, we say the SER is “virtually zero” because although the SER testing with ECC enabled resulted in an SEU FIT rate = 0, there is still a non-zero statistical probability that some SER events will result in MBUs. Taking that into account, we still expect the SEU FIT rate to be < 1.

- **Simplified Controller Design/Broader Range of 3rd Party Controller Options**
As discussed previously, on-chip ECC removes the burden of implementation from the controller, thereby simplifying custom controller design and maximizing 3rd party controller options for a particular application.
- **Increased Utilization Efficiency**
On-chip ECC provides utilization efficiency benefits over an external ECC implementation.

With external ECC, some number of data bits associated with each Read and Write operation must be allocated for the ECC parity bits. The specific number of ECC parity bits needed depends on the size of the data quantum to be protected.

For example, if a 72-bit data quantum is protected with external ECC, then at least 7 of those 72 bits of data must be allocated to ECC—approximately 10%—leaving 65 bits for working data. In that case, 90% of each data quantum can be used for working data, but the minimum unit of data that can be written without requiring a Read-Modify-Write series of operations is 72 bits—4 times larger than with on-chip ECC, which protects an 18-bit data quantum. That could have a non-negligible impact on the effective bandwidth of the part, depending on how frequently data units less than 72 bits must be written in the application.

Or, if an 18-bit data quantum is protected with external ECC, then at least 5 of those 18 bits of data must be allocated to ECC—approximately 28%—leaving 13 bits for working data. In that case, the minimum unit of data that can be written without requiring a Read-Modify-Write series of operations is 18 bits—the same as with on-chip ECC, but only 72% of each data quantum can be used for working data.

Whereas, with on-chip ECC, the minimum unit of data that can be written without requiring a Read-Modify-Write series of operations is 18-bits, and 100% of each data quantum can be used for working data.

Utilization Efficiency Comparison Between On-Chip ECC and External ECC

	On Chip ECC		External ECC					
	18b ECC Quantum		72b ECC Quantum		36b ECC Quantum		18b ECC Quantum	
	x36 part	x18 part	x36 part	x18 part	x36 part	x18 part	x36 part	x18 part
Data allocated for ECC (min)	none	none	7b	n/a	12b	6b	20b	10b
Working Data (max)	72b	36b	65b		60b	30b	52b	26b
Utilization Efficiency	100%		90%		83%		72%	
Min Write Unit Without Read-Modify-Write	18b		72b		36b		18b	

Byte Write Implications (Type II and II+)

Per the industry standard, Type II and II+ SigmaQuad and SigmaDDR SRAMs support Byte Write operations via Byte Write Enable (\overline{BWn}) input pins. Type II and II+ SigmaQuad and SigmaDDR ECCRAMs also support Byte Write operations via those input pins, with the following exception:

If Half Write operations (i.e., write operations in which a \overline{BWn} pin is asserted Low for only half of a DDR write data transfer on the associated 9-bit data bus, in order to cause only 9 bits of the 18-bit DDR data word to be written) are initiated, the on-chip ECC will be disabled for as long as the ECCRAM remains powered up thereafter. This must be done because ECC is implemented across entire 18-bit data words, not across individual 9-bit data bytes.

The truth table below applies to write operations to Address “m”, where Address “m” is the 18-bit memory location comprising the 2 beats of DDR write data associated with each \overline{BWn} pin in a given clock cycle.

Byte Write Truth Table

\overline{BWn}		Input Data Byte n		Operation	ECC Disabled?	Result
$\uparrow K$ (Beat 1)	$\uparrow \overline{K}$ (Beat 2)	$\uparrow K$ (Beat 1)	$\uparrow \overline{K}$ (Beat 2)			
0	0	D0	D1	Full Write	No	D0 and D1 written to Address m
0	1	D0	X	Half Write	Yes	Only D0 written to Address m
1	0	X	D1	Half Write	Yes	Only D1 written to Address m
1	1	X	X	Abort Write	No	Address m unchanged

Notes:

1. $\overline{BW0}$ is associated with Input Data Byte D[8:0].
2. $\overline{BW1}$ is associated with Input Data Byte D[17:9].
3. $\overline{BW2}$ is associated with Input Data Byte D[26:18] (in x36 only).
4. $\overline{BW3}$ is associated with Input Data Byte D[35:27] (in x36 only).
5. SigmaQuad B4 devices execute Burst of 4 (i.e., 4-beat) write operations. However, for the purposes of this table they should be viewed as a pair of Burst of 2 (i.e., 2-beat) write operations.