

# Parallel Double Error Correcting Code Design to Mitigate Multi-Bit Upsets in SRAMs

Riaz Naseer and Jeff Draper  
Information Sciences Institute  
University of Southern California  
Marina del Rey, CA 90292  
{naseer, draper}@isi.edu

**Abstract** The range of SRAM multi-bit upsets (MBU) in sub-100nm technologies is characterized using irradiation tests on two prototype ICs, developed in 90nm commercial processes. Results reveal that MBU, as large as 13-bit, can occur in these technologies, limiting the efficacy of conventional SEC-DED error-correcting codes (ECC). A double-error correcting (DEC) ECC implementation technique suitable for SRAM applications is presented. Results show that this DEC scheme reduces errors by 98.5% compared to only 44% reduction by conventional SEC-DED ECC.

## I. INTRODUCTION

SRAM reliability faces serious challenges from radiation-induced soft errors in sub-100nm technologies [1]. SRAM cells are designed with minimum geometry devices to increase density and performance; however, a consequence is that the critical charge ( $Q_{crit}$ ) that can upset such cells has become very small, potentially increasing the upset frequency. Therefore, it has become conventional to protect memories with the application of error correcting codes (ECC) such as single-error-correcting (SEC) Hamming code, single-error-correcting-double-error-detecting (SEC-DED) extended-Hamming, or SEC-DED Hsiao codes [2][3][4]. With increasing multi-bit upset (MBU) trends [5][6], conventional single-bit correcting ECC may not be sufficient to meet reliability goals. The problem is further exacerbated for space electronics where galactic cosmic rays carry heavy-ions with much higher linear energy transfer (LET) characteristics compared to terrestrial radiation sources.

This work presents heavy-ion induced upset results for two prototype SRAM ICs designed in two characteristic 90nm processes revealing the extent of MBU in these processes. These results show that the multi-bit upsets in these processes can be as large as 13-bits, implying that current ECC architectures which use column interleaving by 2 or 4 with conventional SEC/SEC-DED codes ([7][8]) are not sufficient to mitigate the expected soft errors. This deficiency motivates the exploration of more powerful ECC implementations, such as double error correcting (DEC) BCH (Bose-Chaudhuri-

Hocquenghem) codes. Commonly employed iterative BCH decoding schemes such as Berlekamp-Massey, Euclidian and Minimum Weight Decoding algorithms require a multi-cycle decoding latency [9][10][11], which is not tolerable for embedded memories. Therefore, a new parallel implementation approach is presented for DEC BCH codes. This parallel approach is suitable for SRAM applications where data is accessed in words on every transaction. In addition, a design space for various ECC techniques has been explored by implementing SEC Hamming, SEC-DED Hsiao, DEC and DEC-TED BCH codes using IBM's 90nm standard cell ASIC technology. The implementation results demonstrate the practicality of the proposed decoding implementation approach and also offer insights for various trade-offs for selecting soft error mitigation techniques. Accelerated irradiation test results on prototype SRAM ICs demonstrate the relative reliability efficiency of SEC-DED and DEC ECC techniques.

## II. HEAVY-ION-INDUCED UPSETS

This section presents heavy-ion-induced soft error results for two SRAM ICs designed using commercial 90nm processes. One IC has been designed in a low-power process (henceforth labeled LP), while the other IC has been designed in a standard high-performance process (labeled SF), employing a foundry provided 6T SRAM cell designed for each process. To characterize the intrinsic radiation response of the processes, each IC contains a *baseline* SRAM module of 64-kbits without ECC protection and any hardening applied on peripheral logic. In addition, each IC also contains a *hardened* module that applies ECC on the memory array to mitigate upsets and triple-modular redundancy (TMR) on peripheral logic to guard against single-event transients. The size of the hardened array increases in proportion to the redundancy required by the applied ECC scheme in each IC. Both SRAMs have been extensively tested using a 10MeV cocktail beam at the Lawrence Berkeley National Laboratories (LBNL) 88-inch cyclotron. Fig. 1 shows the distribution of raw upsets for LP and SF SRAMs versus ion LET value, before any ECC correction is applied.

---

This work was supported by the Defense Advanced Research Projects Agency (DARPA) Microsystems Technology Office under award No.: N66001-04-1-8914. Any opinions, findings, and conclusions or recommendations expressed in this work are those of the authors and do not necessarily reflect the views of DARPA/MTO or the U.S. Government.

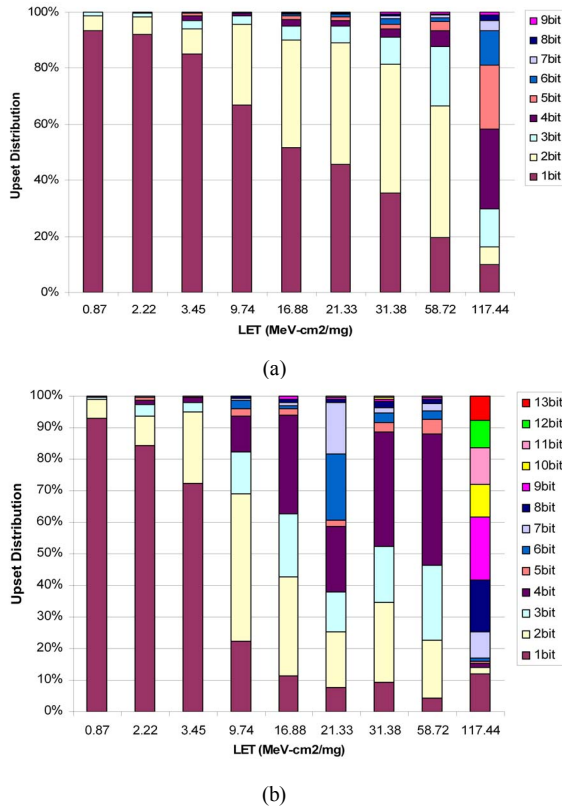


Figure 1. Single- and Multi-Bit Upset Distributions versus Effective LET (a) LP SRAM (b) SF SRAM

At onset, single-bit upsets (SBU) dominate the total soft error distribution; however MBU quickly become the main contributor as the LET increases. A wide difference in MBU distribution can be noted for the two ICs, in addition to observing that the largest MBU is 9 bits for LP versus 13 bits for SF. The difference in LP and SF upset distributions can be understood by investigating the 6T SRAM cell critical charge in each process. Our prior simulations show that the  $Q_{crit}$  of the LP cell is slightly higher than the SF (1.58fC vs. 1.23fC), due to variations in cell design and nominal operating voltages (1.2V and 1V, respectively). These MBU distributions suggest that much larger column interleaving factors must be implemented if only the conventional single error correcting ECC is employed. However, the aspect ratio of memory array and column multiplexer design may potentially restrict the interleaving factor to 8 or 16. Moreover, if the increasing MBU trend continues with technology scaling as expected, it will become necessary to utilize more powerful codes such as double error correcting ECC.

### III. PARALLEL IMPLEMENTATION APPROACH FOR BCH CODES

Having projected increasing MBU trends in SRAMs, it is important to begin exploring stronger ECC schemes. Though such schemes have been explored in communication applications, some re-design is necessary to make them suitable for memory protection. Particularly, a pure combinational logic approach has been adopted to implement double error correcting BCH codes in order to overcome the multi-cycle decoding latency of conventional BCH decoders.

This approach is constructed on a standard array based syndrome decoding procedure, where a set of syndromes is pre-computed corresponding to correctable error patterns. The error correction bits are then set according to a Boolean function mapping of syndrome patterns. This allows a Boolean function implementation using a standard cell ASIC design methodology.

For the following, recall from coding theory that a binary  $(n, k)$  linear block code is a  $k$ -dimensional subspace of a binary  $n$ -dimensional vector space. Thus, an  $n$ -bit codeword contains  $k$ -bits of data and  $r (= n - k)$  check bits. An  $r \times n$  parity check matrix  $H$ , or alternatively  $k \times n$  generator matrix  $G$ , is used to describe the code [9]. Due to the cyclic property of BCH codes, a systematic generator matrix of the form  $G_{k,n} = [I_{k,k} | P_{k,r}]$  can be generated by combining two sub-matrices.  $I_{k,k}$  is an identity matrix of dimension  $k$ , and  $P_{k,r}$  is a parity sub-matrix consisting of the coefficients of  $k$  parity polynomials of degree  $r$ . The  $k$  parity polynomials can be obtained from a polynomial division involving the generator polynomial  $g(x)$  of the BCH code as:

$$P_i = \text{remainder}(X^{n-k+i} / g(x)) \text{ mod } 2 \quad i = 0, 1, \dots, k-1 \quad (1)$$

#### A. DEC Encoder

The encoding process converts a data word (row vector  $\underline{b}$ ) into a codeword (row vector  $\underline{c}$ ) by multiplying it with the generator matrix  $G$  of the code using modulo-2 arithmetic, i.e.,  $\underline{c} = \underline{b} * G$ . With systematic generator matrix  $G$ , data bits are passed as-is in the encoding process and only the check bits need to be computed. The computation of check bits is accomplished through XOR trees, e.g. an encoder is shown in Fig. 2 for DEC (26, 16). The inputs to each XOR tree are data bits chosen according to non-zero entries in respective columns of the parity sub-matrix which is part of the  $G$  matrix. The depth of the XOR tree has an upper bound of  $\log_2(k)$  if implemented using 2-input XOR gates.

#### B. DEC-TED Encoder

A DEC-TED encoder is similar to a DEC encoder except that an additional overall parity bit is added. Since this is an overall parity bit covering the data as well as check bits, it can only be computed after all other check bits have been computed. Therefore, this bit becomes the critical path in a DEC-TED encoder and can considerably increase the latency of the encoder.

#### C. DEC Decoder

For decoding purposes, a parity check matrix  $H$ , of the

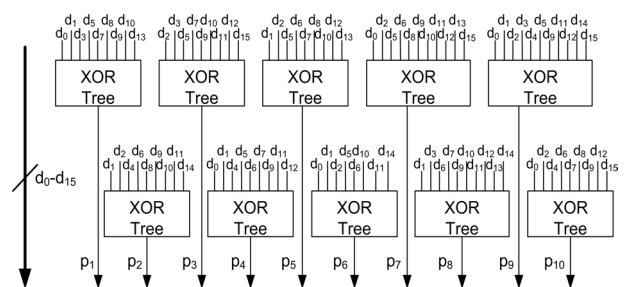


Figure 2. Encoder Circuit for DEC (26, 16)

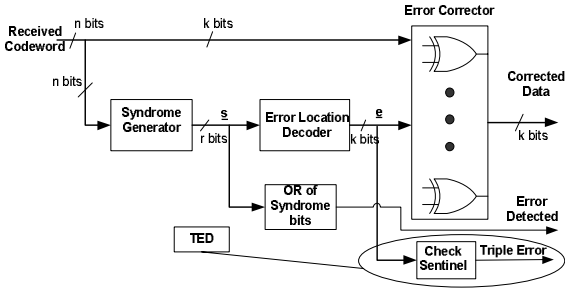


Figure 3. Block Diagram of BCH Decoder

form:  $H_{r,n} = [P^t_{r,k} | I_{r,r}]$ , is required where  $P^t$  is the transpose of the parity sub-matrix in systematic  $G$ . The input to the decoder is the read codeword vector  $\underline{v}$  which may contain errors in data or check bit locations. A block diagram of the decoder is shown in Fig. 3 (ignore the TED portion for now) containing three main parts: 1) Syndrome Generator, 2) Error Location Decoder and 3) Error Corrector. The circuit for the syndrome generator is similar to the encoder circuit. Essentially, it re-computes the check bits and compares those with the received check bits. In algebraic form, this process translates to  $\underline{s} = H * \underline{v}^t$ ; where  $\underline{v}^t$  is the transpose of the read codeword  $\underline{v}$  and  $\underline{s}$  is the syndrome vector. A non-zero syndrome implies the presence of errors and is checked by ORing the syndrome bits to flag error detection. The erroneous bit positions are identified by feeding the syndrome into an error location decoder. The error location decoder circuit is implemented using combinational logic that maps the respective pair of syndromes and correctable error patterns. This mapping is pre-computed by multiplying all correctable error patterns with the parity check matrix  $H$ . For binary vectors, an erroneous bit is corrected merely by complementing it; therefore, the error corrector circuit is simply a stack of XOR gates.

#### D. DEC-TED Decoder

The decoder for a DEC-TED code is similar to the decoder for DEC with modifications necessary to handle triple-bit error detection, as shown in Fig. 3. In particular, an all-0 column and an all-1 row are added to the DEC  $H$  matrix to obtain the parity check matrix for DEC-TED. This increases the syndrome vector by 1-bit, doubling the number of

syndromes. The error location decoder then maps the syndromes for 3-bit errors to a sentinel pattern. A simple sentinel value of the least three bits being set in the error pattern  $\underline{e}$  can be ANDed to flag triple error detection.

## IV. RESULTS AND DISCUSSION

For demonstrating the practicality of the parallel implementation approach, DEC and DEC-TED encoder and decoder circuits have been implemented for typical memory word sizes of 16, 32 and 64 bits. For analyzing trade-offs with conventionally used ECC schemes, SEC and SEC-DED codes have also been implemented. Since the Hsiao code is an optimal SEC-DED code [3], we have included only the Hsiao code for SEC-DED ECC for our comparisons. Synopsys Design Compiler (DC) has been used for synthesizing all encoder and decoder circuits targeted to an IBM 90nm standard cell library. Table I shows the latency and area results for encoder circuits while Table II shows the latency and area results for decoder circuits.

A major inference from the synthesis results is that the decoding latency for the DEC codes is reasonably small, and it is much better compared to multi-cycle shift register based decoders used in communication systems [9][10][11]. Therefore, this parallel implementation of DEC codes makes it feasible for memory applications.

As expected, the overall parity bit results in increasing the latency penalty of the DEC-TED encoder by 80% to 85% as compared to the DEC encoder. Therefore, DEC is preferred wherever possible as compared to DEC-TED. Looking at the decoder results, we see that the latency of SEC and Hsiao SEC-DED decoders is identical. On the other hand, the decoding latency for DEC and DEC-TED varies significantly, between 21% to 36% for 16 and 64 bit decoders respectively. The latency increases between DEC and DEC-TED because the corresponding syndrome and correctable & detectable error patterns are doubled by adding the overall parity bit. For DEC-TED, the computed overall parity cannot simply be compared with the received overall parity to infer triple bit error detection since the computed overall parity will be the same for single- and triple-bit errors, but single-bit errors are correctable while triple-bit errors are not. In contrast, in the Hsiao SEC-DED case, the number of syndromes and correctable error pattern pairs remains the same as for SEC. If

TABLE I ECC ENCODER LATENCY AND AREA RESULTS

Data Width	Ham. SEC		Hsiao SEC-DED		DEC		DEC-TED	
	Latency (ns)	Area (m <sup>2</sup> )	Latency (ns)	Area (m <sup>2</sup> )	Latency (ns)	Area (m <sup>2</sup> )	Latency (ns)	Area (m <sup>2</sup> )
16	0.4	296	0.4	291	0.5	496	0.9	786
32	0.5	598	0.5	605	0.6	1250	1.1	1424
64	0.65	1302	0.7	1168	0.7	2335	1.3	2546

TABLE II ECC DECODER LATENCY AND AREA RESULTS

Data Width	Ham. SEC		Hsiao SEC-DED		DEC		DEC-TED	
	Latency (ns)	Area (m <sup>2</sup> )	Latency (ns)	Area (m <sup>2</sup> )	Latency (ns)	Area (m <sup>2</sup> )	Latency (ns)	Area (m <sup>2</sup> )
16	0.9	576	0.9	935	1.4	4288	1.7	5432
32	1.1	1303	1.1	1376	1.8	11735	2.2	13757
64	1.3	2412	1.3	2681	2.2	37279	3	42976

TABLE III REDUNDANCY FOR ECC

Data bits	SEC Check bits	DEC Check bits
16	5	10
32	6	12
64	7	14
128	8	16

TABLE IV ACCELERATED TEST RESULTS

	LP IC-SEC-DED	SF IC-DEC
Errors Observed before ECC	13,004	12,117
Errors Observed after ECC	7,305	195

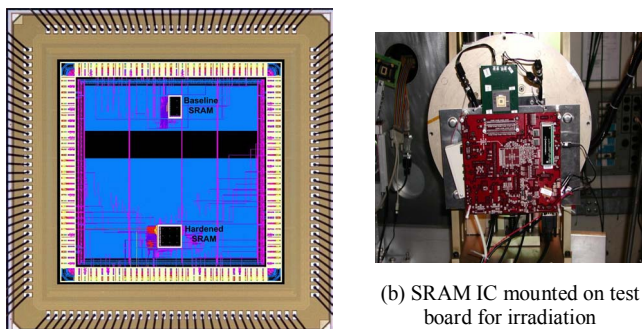


Figure 4. (a) SF IC layout

we compare the latency penalty between Hsiao SEC-DED and DEC-TED, it is almost double both for encoder and decoder. However, in comparing SEC-DED to DEC, the percentage decoder latency penalty varies by 55% to 69% for different block sizes, implying again that DEC is strongly preferred over DEC-TED unless there is strong evidence that the target application requires triple-error detection.

Another important implication can be made from the almost identical latencies of the Hsiao SEC-DED and DEC encoders. Since the syndrome generator circuit is similar to the encoder circuit, error detection can be accomplished with quite similar latency for both SEC-DED and DEC codes. The ECC implementation architecture can benefit remarkably from this observation. In particular, as most memory accesses would be error-free, data can be passed for processing to the next stage without the full decoder delay. In erroneous cases, when a non-zero syndrome is detected, only then is the full decoder latency needed to correct the errors. For these cases only, the next processing stage can be stalled for a cycle or two depending on the speed of the processing stage, minimizing the overall performance penalty.

The spread of the area results for the encoder and decoder circuits is quite large for various ECC schemes, but the overall area for each is very small compared to typical ASIC sizes. Thus, these ECC encoders and decoders can be implemented without a significant area impact. Notice that the extra redundancy required within the memory array for a particular code is a function of the error detection and correction capability of that code and block size. Table III lists the redundancy requirements for SEC and DEC codes for typical memory word sizes, showing that DEC requires twice as many check bits as SEC ECC.

For evaluating the relative reliability efficiency of the ECC techniques, we implemented the Hsiao SEC-DED code on the LP prototype SRAM IC and the DEC code on the SF IC following our parallel decoding approach. Fig. 4 (a) shows the layout of the SF IC embedded in the chip micrograph, and Fig. 4(b) shows the SRAM chip mounted on the tester board for irradiation. As can be seen from the layout, the hardened array is larger compared to the baseline array due to extra redundancy required for the DEC BCH code. The accelerated heavy-ion irradiation testing, performed on the prototype SRAM ICs according to a JEDEC standard, showed that the SEC-DED code reduced the error count only by 44%. On the

other hand, the implemented DEC code reduced the error count by more than 98%, as shown in Table IV. This increased error coverage of DEC ECC easily justifies its associated implementation cost for many applications.

## V. CONCLUSION

Heavy-ion-induced soft error results for SRAM ICs designed in two characteristic 90nm processes have been presented. The upset distributions exhibit that MBU are the dominating contributor to overall soft error rate, and these MBU can range as large as 9-bits for LP and 13-bits for SF necessitating the usage of more powerful ECC schemes. Implementations of DEC and DEC-TED ECC using a parallel implementation approach in 90nm technology demonstrate that these codes can effectively be applied for SRAM applications. Synthesis results for different ECC circuits reveal various trade-offs and provide guidelines for choosing a particular solution depending on the application requirements. Test results on prototype SRAM ICs demonstrate that DEC reduces the error count by more than 98% compared to only 44% for SEC-DED ECC.

## ACKNOWLEDGMENT

The authors would like to thank Michael Bajura and Younes Boulghassoul for design collaboration and collecting the test results. The authors also thank Jeff Sondeen for physical design of the ICs and Scott Stansberry for the tester board design.

## REFERENCES

- [1] R.C. Baumann, "Radiation-induced soft errors in advanced semiconductor technologies," *IEEE Trans. Device Mater. Reliab.*, vol. 5, no. 3, pp. 305–316, Sep. 2005.
- [2] R.W. Hamming, "Error Correcting and Error Detecting Codes," *Bell Sys. Tech. Journal*, Vol 29, pp. 147-160, April 1950
- [3] M. Y. Hsiao, "A Class of Optimal Minimum Odd-weight-column SEC-DED Codes", *IBM Journal of R & D* Vol. 14, July 1970, pp. 395-401
- [4] C.W. Slayman, "Cache and memory error detection, correction, and reduction techniques for terrestrial servers and workstations", *Device & Materials Reliability*, *IEEE Trans. on Vol. 5*, Sept. 2005 pp. 397 – 404
- [5] D. Radaelli, H. Puchner, S. Wong, S. Daniel, "Investigation of multi-bit upsets in a 150 nm technology SRAM device", *IEEE Trans. On Nucl. Sci.* Vol. 52, Issue 6, pp. 2433- 2437, Dec 2005
- [6] D. Giot, P. Roche, G. Gasiot, R. Harboe-Sorensen, "Multiple-Bit Upset Analysis in 90 nm SRAMs: Heavy Ions Testing and 3D Simulations", *IEEE Trans. Nucl. Sci.*, Vol. 54, Aug. 2007 pp.904 – 911
- [7] K. Osada, Y. Saitoh, E. Ibe, K. Ishibashi, "16.7fA/cell Tunnel-Leakage-Suppressed 16Mb SRAM for Handling Cosmic-Ray-Induced Multi-Errors", *ISSCC Dig. Tech. Papers*, pp. 302-303, Feb. 2003
- [8] T. Suzuki et al, "0.3 to 1.5V Embedded SRAM with Device-Fluctuation-Tolerant Access-Control and Cosmic-Ray-Immune Hidden ECC Scheme", *ISSCC Dig. Tech. Papers*, pp. 484-612, Feb. 2005
- [9] S. Lin, D. J. Costello, "Error Control Coding: Fundamental and Applications", Prentice-Hall, 1983
- [10] Shyue-Win Wei, Che-Ho Wei, "High-speed hardware decoder for double-error-correcting binary BCH codes", *Communications, Speech and Vision, IEE Proceedings Jun 1989* Vol. 136, Issue 3, pp. 227- 231
- [11] W.M. El-Medany, C.G. Harrison, P.G. Farrel, and C.J. Hardy, "VHDL Implementation of a BCH Minimum Weight Decoder for Double Error", *Radio Science Conf., Proc. of the 18th 2001* Vol. 2, pp. 361-368