IMPROVING ERROR CORRECTION CODES FOR MULTIPLE-CELL UPSETS IN SPACE APPLICATIONS

VEDIKA.S[1] and DR.S.UMA MAHESWARI[2]

[1]M.E Research Scholar Electronics and Communication Engineering, Coimbatore Institute of Technology, Coimbatore, Email: vedikascit@gmail.com

2 Professor Electronics and Communication Engineering, Coimbatore Institute of Technology,Coimbatore, Email: umamaheswari@cit.edu.in

***Corresponding Author**
VEDIKA .S
M.E Research Scholar, Electronics and Communication Engineering,
Coimbatore Institute of Technology, Coimbatore,
Email: vedikascit@gmail.com

ABSTRACT:

With the present aggressive scaling, the memory cell critical charge and the energy needed to provoke a Single-Event Upset (SEU) in storage have been reduced. Currently, faults suffered by SRAM memory systems have exaggerated thanks to the aggressive CMOS integration density. Thus, the likelihood of incidence of Single-Cell Upsets (SCUs) or Multiple-Cell Upsets (MCUs) augments. One in every of the most causes of MCUs in house applications is radiation, a typical resolution is that the use of Error Correction Codes (ECCs). However, once exploitation ECCs in house applications, they have to bring home the bacon an honest balance between error coverage and redundancy, and their encoding/decoding circuits should be economical in terms of space, power, and delay. Totally different codes are projected to tolerate MCUs. Recently bestowed, Column–Line–Code (CLC) has been designed to tolerate MCUs in house applications. CLC could be a changed Matrix code, supported extended performing codes and parity checks. Proposed scheme uses the codes have been designed using Flexible Unequal Error Control Methodology.

Keywords: Multiple Cell Upsets, Column Line Code, Error Correction Codes

I. INTRODUCTION

The general idea for achieving error detection and correction is to add some redundancy which means to add some extra data to a message, which receiver can use to check uniformity of the delivered message, and to pick up data determined to be corrupt. Error-detection and correction scheme may be systematic or it may be non-systematic. In the system of the module non-systematic code, an encoded is achieved by transformation of the message which has least possibility of number of bits present in the message which is being converted.[1,2] Another classification is the type of systematic module unique data is sent by the transmitter which is attached by a fixed number of parity data like check bits that obtained from the data bits.

The receiver applies the same algorithm when only detection of the error is required to the received data bits which is then compared with its output with the receive check bits if the values does not match, there we conclude that an error has crept at some point in the process of transmission. Error correcting codes are regularly used in lower-layer communication, as well as for reliable storage in media such as CDs, DVDs, hard disks and RAM.

Multiple cell upsets (MCUs) caused by neutron-induced soft errors square measure very problematic for on-chip memory systems because the size and therefore the offer voltage of semiconducting material devices shrink within the deep sub-micron regime. [3]These upsets could cause burst errors in physically adjacent memory cells as a result of the heavy-ions tracks left by nuclear reactions alter the data in storage nodes being modified. Soft errors in on-chip memory systems are also effectively protected by error correcting codes (ECCs). Single error correction and double error discovering (SEC–DED) codes correct single-bit errors and detect double-bits errors victimization little redundant bits known as confirmation bits and square measure wide used in business on-chip memory systems. [4]However, these codes cannot faithfully defend these systems against MCUs. To correct multiple adjacent errors caused by MCUs, multiple burst error correction codes (MBECCs) square measure projected. There is SEC–DED and double adjacent error correction (SEC–DED–DAEC) codes which exploit linear block coding.[14 15] These codes are shortened Hamming codes that correct both single and double adjacent errors.

However, because some syndromes for double adjacent errors and double non-adjacent errors are equal, these codes are prone to mis-correction. Double error correction Bose–Chaudhuri–Hocquenghem (DEC BCH) codes for on-chip memories are proposed and these codes correct random and burst double errors without mis-correction, they require large parity bits and decoding latency. Triple adjacent error correction codes are derived from orthogonal Latin square codes that can correct random and burst double errors with simple one-step majority logic decoding. [7 8]However, these codes require more parity bits than DEC BCH codes. In addition, low-cost burst error correction codes are more efficient for flip-flops than on-chip memories because they offer rapid decoding, but still require many parity check bits. [5 6]In this paper, we propose cost effective SEC–DED, and double–triple adjacent error correction (SEC–DED–DTAEC) codes for protecting on-chip memory systems against MCUs. In simulation experiments and comparison with conventional MBECCs, the proposed codes corrected single error and double adjacent errors without mis-correction and corrected triple adjacent errors with same number of parity check bits as SEC–DED–DAEC codes.

The main problem when memory systems employ an ECC is the redundancy required. The extra bits added are used to detect and/or correct the possible errors occurred. Also, redundant bits must be added for each data word stored in memory. In this way, the amount of storage occupied for redundant bit scales with the memory capacity. [10]For example, if an ECC with 100% of redundancy is employed in a 2-GB memory, only 1GB is available to store the payload (the clean data) and the remaining 1 GB is required for code bits. [12]In addition, the usage of an ECC implies overheads in the area, power, and delay employed by the encoder and decoder circuits. These overheads must be maintained as low as possible, especially in space applications.

II. EXISTING CODE

Error correcting codes are widely used in protecting memories against the soft errors that are occurring due to the changes in the environment and the operating point of the devices.

Hamming codes are widely used to protect memories against SEU because of the reduced area and performance. Hamming codes are used for single error deduction (SEC) and multiple error deduction. Hamming codes are capable of deducting up to two errors in a given code word.

Hamming codes can be easily built for any word length. Also, the encoding and decoding circuits are easy to implement. Their main drawback is that only one bit in error can be corrected. Nevertheless, for common data word lengths (8, 16, 32, and 64), Hamming codes can detect some double error patterns, in addition to the SEC. Exploiting this feature, it is possible to systematize the detection of 2-bit adjacent errors with the same redundancy, as presented in [14] and [15].

The main problem of Hadamard and Repetition codes is that they introduce a great redundancy for common data word lengths . This great redundancy provokes the necessity of a great memory storage capacity, which is an inconvenient for space applications. Golay code is able to correct up to 3-bit errors. Nevertheless, Golay code presents a redundancy of almost 100%. Also, this code presents a high time and power consuming ratio, as it has to execute sequentially two complementary sequences.

Although BCH and Reed–Solomon codes can correct multiple errors, their main drawbacks are the great complexity and difficulty to implement them, as well as their great latency and speed. These weaknesses can be very problematic in space applications. Concerning Reed–Muller codes, although vastly used in critical applications, they present a great complexity. In this way, the overheads introduced are higher than the overheads introduced by Matrix or CLC codes.

Multidimensional codes are a class of matrix codes that uses parity bits to detect and correct errors. With a low redundancy, these codes present several drawbacks. When more than two errors must be corrected, the code design is very complicated. Also, it is very difficult to adapt these codes to standard data word sizes (i.e., 16, 32, or 64 bits).

III. PROPOSED CODE

FLEXIBLE UNEQUAL ERROR CONTROL METHODOLOGY

An algorithm developed by the authors is employed to solve it and to obtain a parity-check matrix, which defines the code to be designed. After defining the values of n and k for the code, the first step is the selection of error patterns to be corrected and detected. For instance, single errors are represented with error vectors (... 1 ...), and error vectors for double random errors show the pattern (... 1 ... 1 ...), where 1's represent the bits in error, and the dots represent the correct bits. The next step is to find the parity-check matrix H that satisfies the conditions (4) and (5), where E+ represents the set of error vectors to be corrected, and E is the set of error vectors to be detected. That is, each correctable error must generate a different syndrome.

To find the matrix, a recursive backtracking algorithm is used. It checks partial matrices and adds a new column only if the previous matrix satisfies the requirements. In this way, the algorithm starts with an empty partial matrix. New columns, with n–k rows, are added, and the new partial matrices are checked recursively. The added columns must be nonzero, so there are $2^{n-k}-1$ combinations for each column. The complete execution of the algorithm is commonly unfeasible. Nevertheless, the first solutions are usually found quickly, if the code exists. Once selected the H matrix, it is easy to determine the logic equations to calculate each parity and syndrome bit, as well as the syndrome lookup table. They are required for the encoder and decoder implementation.

IV. Experimental RESULTS

Simulation Result of Corrector

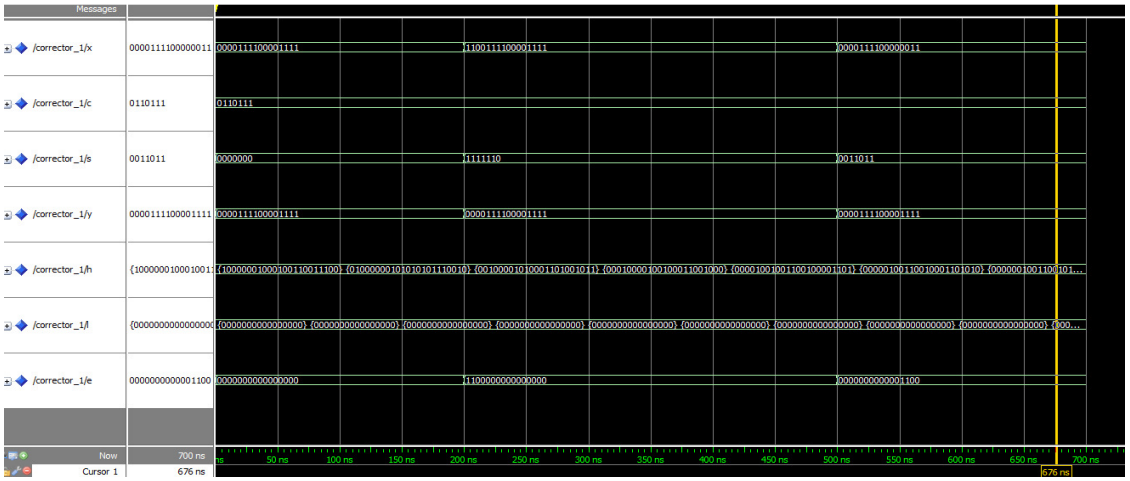

Fig 4.1 Simulation Output for Corrector

Simulation Result of Detector

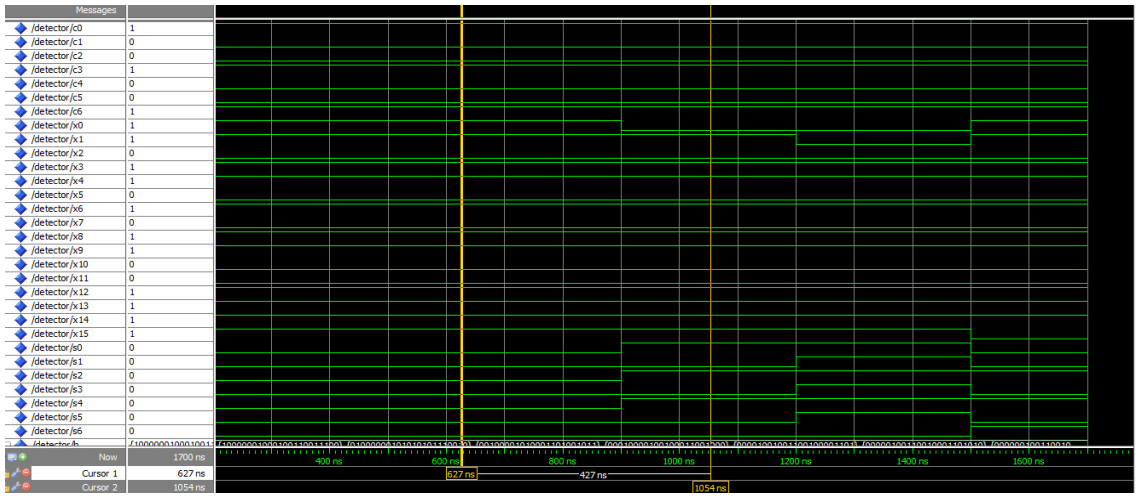

Fig 4.2 Simulation Output for Detector

Simulation Result of Encoder

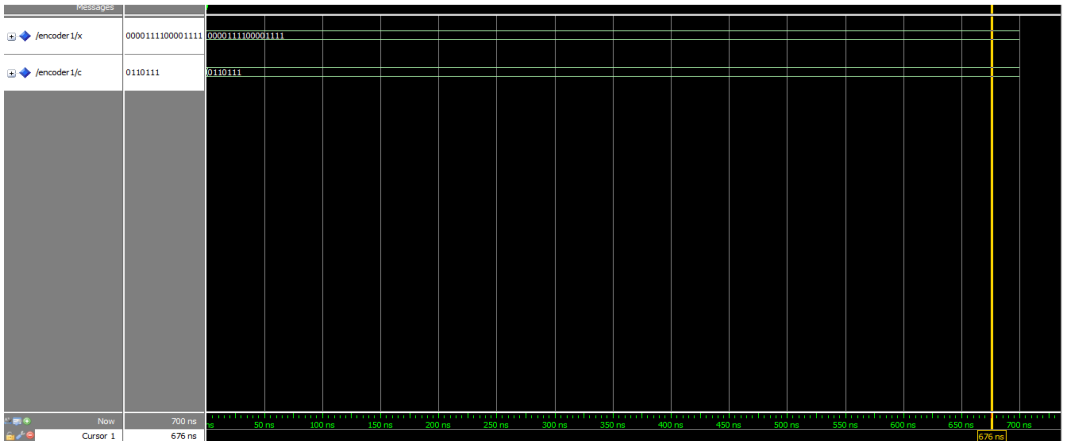

Fig 4.3 Simulation Output for Encoder

Simulation Result of FUEC

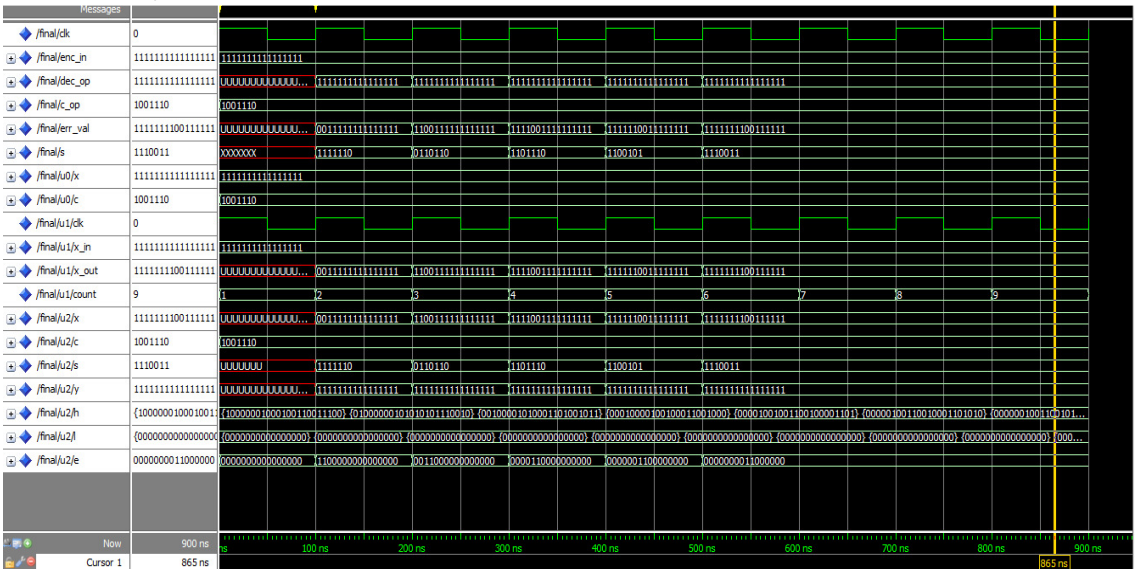

Fig 4.4 Simulation Output of FUEC

V. CONCLUSION

The insertion of an ECC in SRAM memory also provokes the introduction of area, power, and delay overheads. Relating to the area overhead, proposed code introduces a much lower area overhead than the CLC and Matrix codes, while FUEC–TAEC code presents a similar area overhead than Matrix and CLC codes. With respect to power the flexible unequal error control codes takes much lower power than other codes. In case of delay, proposed code corrects error much faster than the existing codes. our codes become an appropriate option for critical applications in embedded systems. Beyond 4-bit burst errors, the performance of our codes decreases notably due to their low redundancy. If these errors are expected to occur, more powerful ECCs must be employed.

REFERENCES

[1]. N. B. Anne, U. Thirunavukkarasu, and S. Latifi, Three and four dimensional parity-check codes for correction and detection of multiple errors, in Proc. Int. Conf. Inf. Technol. Coding Comput. (ITCC), Apr. 2004: 837–842.

[2]. J. Barak, M. Murat, and A. Akkerman, SEU due to electrons in silicon devices with nanometric sensitive volumes and small critical charge, Nucl. Instrum. Methods Phys. Res. B, Beam Interact. Mater. At, 2012:287:113-119.

[3]. Y. Bentoutou, Program memories error detection and correction on board earth observation satellites, Int. J. Elect. Comput. Eng, 2010:4(6):933-936.

[4]. A. Mukati, Review: A survey of memory error correcting techniques for improved reliability, J. Netw. Comput. Appl, 2011:34(2):517-522.

[5]. J. M. Berger, A note on error detection codes for asymmetric channels, Inf. Control,1961:4:68-73.

[6]. M. J. Gadlage, A. H. Roach, A. R. Duncan, A. M. Williams, D. P. Bossev, and M. J. Kay, Multiple-cell upsets induced by single high-energy electrons, IEEE Trans. Nucl. Sci, 2018:65(1):211-216.

[7]. O. Goldreich and L. A. Levin, A hard-core predicate for all one-way functions, in Proc. 21st ACM Symp. Theory Comput., 1989, pp. 25–32.

[8]. M. J. E. Golay, Notes on digital coding, Proc. IRE, 1949:37:657.

[9]. E. Ibe, H. Taniguchi, Y. Yahagi, K. Shimbo, and T. Toba, Impact of scaling on neutron-induced soft error in SRAMs from a 250 nm to a 22 nm design rule, IEEE Trans. Electron Devices, 2010:57(7):1527-1538.

[10]. S. K. Kurinec and K. Iniewski, Nanoscale Semiconductor Memories: Technology and Application. Boca Raton, FL, USA: CRC Press, 2014.

[11]. N. N. Mahatme, B. L. Bhuva, Y.-P. Fang, and A. S. Oates, Impact of strained-Si PMOS transistors on SRAM soft error rates, IEEE Trans. Nucl. Sci, 2012:59(4):845-850.

[12]. D. E. Müller, Application of Boolean algebra to switching circuit design and to error detection, IRE Trans. Electron. Comput, 1954:3(3):6-12.

[13]. L. J. Saiz-Adalid, P. Gil, J.-C. Baraza-Calvo, J.-C. Ruiz, D. Gil-Tomas, and J. Gracia-Moran, Modified Hamming codes to enhance short burst error detection in semiconductor memories, in Proc. IEEE Eur. Dependable Comput. Conf. (EDCC), Newcastle, U.K., May 2014:62–65.

[14]. A. Sánchez-Macián, P. Reviriego, and J. A. Maestro, Hamming SECDAED and extended Hamming SEC-DED-TAED codes through selective shortening and bit placement, IEEE Trans. Device Mater. Rel., 2014:14(1):574-576.

[15]. G. Tsiligiannis et al., Multiple cell upset classification in commercial SRAMs, IEEE Trans. Nucl. Sci, 2014:61(4):1747-1754.