IJCAES
ISSN: 2231-4946
*Volume III, Special Issue, August 2013*

***International Journal of Computer Applications in Engineering Sciences***
*Special Issue on National Conference on Information and Communication (NCIC'13)*
www.caesjournals.org

# Design Of Cost Efficient Memory Using Advanced Error Correction Method

M.Poornima[1], P.Rajendrian[2]

[1]*M.E VLSI DESIGN, Sree Sastha Institute of Engineering and Technology, Chennai, India.*

[1]mpoorninm@gmail.com

***Abstract*— Faults in the memory generally tested by comparing faulty circuits with test circuits. Due to this testing process cost of memory is increasing. To reduce testing cost prefect error correcting method is required. A methodology existing here is called a matrix code, combines hamming and parity code to assure the improvement of reliability and yield of the memory chip in the presence of high defects. The method is evaluated using fault injection experiments. The result are compared to well-known technique such as reed-muller and hamming code. LDPC (Low Density Parity Code) codes are find in increasing use in application requiring reliable and highly efficient information transfer over bandwidth or return channel-constrained links in the presence of data corrupting noise LDPC which will detect and correct multiple errors in memory.**

***Keywords*—Single event upset, MBU, LDPC**

## I. INTRODUCTION

As cmos process technology scales, high density, low cost, high speed integrated circuits with low voltage level and small noise margin will be increasingly susceptible to temporary faults. In very deep sub micrometer technology. Single event upsets (SEU) like atmospheric neutrons and alpha particles severely impact field-level product reliability not only for memory but for logic as well when these particles hits the silicon bulk, they create minority carriers which if the collected by the source/drain diffusion, could change the voltage level of the node.

This issue has drawn a growing attention from the fault tolerance community due to recent increase of soft error late of combinational logic circuits [3]. While effective solution to protect memory elements have already been devised [4] the low probability of soft errors affecting Cmos combinational circuits being latched at this output of the circuit keep this subject as a secondary research point. Therefore not many techniques to cope with the problem have been proposed until now. Similar concerns are also expressed for critical application such as space, where there can be potentially serious consequences for the spacecraft, including loss of information, functional failure, or loss of control [5]. Although SEU is major concern for some critical application, multiple bit upset (MBU) are also becoming important problem in designing memories mostly because of following.

1) Error late of memories are increased due to technology shrinkage [4], [5]. Therefore probability of having multiple errors increases.

2) MBU can be induced by direct ionization or nuclear recoil. After passing high energy ion [8].

3) The probability of having multiple error is increased when the size of memory is increased.

Most common approach to maintain a good level of reliability for memory cell is to use error correcting codes. Hamming and odd weight codes are largely used to protect memories against SEU because the ability to correct SEUs efficiently with reduced area and performance overhead. On the other hand, there are advanced error correcting code such as reed. Muller code [13], which can cope with multiple upsets. This is achieved by at expense of high area and power consumption. The common approach to deal with multiple errors has been the use of interleaving in physical avengement of memory cells, so that cells that belongs to same logical word are separated. As error in an MBU are physically close as discussed in [14] they will cause single error in different word that can be corrected by single error correction-double error detection (SEC-DED) code.

Interleaving cannot be used for small memories or register files and in other cases, it use may have an impact of floor planning access time and power consumption. For this reason use of more sophisticated code or combination of different code has been proposed to deal with MBUs when the use of interleaving is not a valid option. Many different technique have been proposed based on use of redundant elements to replace defective ones. This technique vary from those applied during manufacturing process, in test phase, to use in built-in-circuits, which able to operate during normal operation in field, with different trade off in terms of cost and speed.

In this paper, a high level method for detection and correction of multiple faults is proposed. This method is based on parity check matrix and generator matrix. So detection and correction of multiple faults is achieved.

## II. BACK GROUND AND RELATED WORK

Hamming and odd weight codes are largely used to protect memories against SEU because of their efficient ability to correct single upset with a reduced area and performance overhead. Hamming code implementation is composed by a combinational block responsible to code the data, inclusion of extra bits in words that indicates parity and another combinational block responsible for decoding the data. Encoder block calculates the parity bit, and it can be implemented by set of two-input XOR gate. Decoder block is more complex than encoder block, it needs not only the detect the fault but it must also correct it, which is composed of set of two-inputs XOR gats, some AND and INVERTER gate. However, it does not cope with multiple upset. Consequently more complex correcting code must be investigated.

Reed-Muller code [`13] is another protection code that is able to detect and correct more error than a Hamming code, the main draw of this protection code is its high area and power penalties.

Many different technique have been proposed, all of them are based on the use of redundant element to replace defective ones, those technique vary from those applied during the manufacturing process, in the test phase ,to the use of built in circuit is able to repair the memory chips even during normal operation in the field with different trade off in terms of cost and speed. No matter when the chip repair is performed those technique relay in a few basic redundancy schemes for instances, in redundant rows or redundant columns approach only redundant rows are included in the memory array and are used to replace defective rows detected during test also known as 1 D redundancy, the main advantage of this approach is that its implementation is very straight forward, requiring no complex redundant rows and allocation algorithm. However, its repair efficiency can low, since its defective column contain multiple defective cells cannot be replaced by a single redundant rows.

Matrix code can correct two errors in any low, assuming that we have only one error in other low. If only two errors occurs, they can be corrected without any restriction.[1]

In order to improve the efficiency of repairing embedded memories, a novel redundancy mechanism to cope not only with defects but as well as with transients is required. Our technique, in compassion with previous techniques, dramatically reduces the cost of chip while improving the overall system reliability.

## III. LDPC CODES

As their name suggests, LDPC codes are block code with parity check matrices that contains only a very small number of non—zero entries. It is sparseness of H which guarantees both a decoding complexity which increases only linearity with the code length and minimum distance which is also increases linearly with the code length. Aside from the requirement that H sparse, an LDPC code itself is no different to any other block code. Indeed existing block codes can be successfully used with the LDPC iterative decoding algorithm if they can be represented by a sparse parity-check matrix. Generally however, finding a sparse parity check matrix for an existing code is not practical. Instead LDPC codes are designed by constructing a sparse parity-check matrix first and the determining a generator matrix for the code afterwards. The biggest difference between LDPC code and classical block code is how they are decoded.

For large block size, LDPC codes are commonly constructed by first studying the behavior of decoders .LDPC decoders shown to have a noise threshold below which the decoding is reliably achieved and above which the decoding is not achieved .construction of specific LDPC code after this optimization falls into two main types of technique as pseudo-random approaches ,for large block size, a random construction gives good decoding performance but complex encoders combinatorial approach can be used to optimize the properties of small block size LDPC code.

The desirable properties of LDPC code depends on how they are to be applied. For a capacity approaching performance of low noise channel long code length and random or pseudo-random constructed irregular parity check matrices produces the performance closes to capacity. However, capacity approaching performance equate to poor word error lates and low error floors, making capacity approaching codes completely unsuitable for some application. In particular for very low error floors, a reasonably short algebraic construction with large column weight will produce the required performance.

Basically there are two different possibilities to represent LDPC codes. Like all linear block codes, they are described by matrices. The second possibility is a graphical representation.

### *A.Matrix representation*

Lets look at the for a Low density parity check matrix first. The matrix defined in equation 1 is a parity check matrix with dimension n x m for a (4,8) code. We can define two numbers describing these matrix $w_r$ for the number of ones in each rows and $w_c$ for columns. For a matrix to be called low density the two conditions $w_c << n$ and $w_r << m$ must .

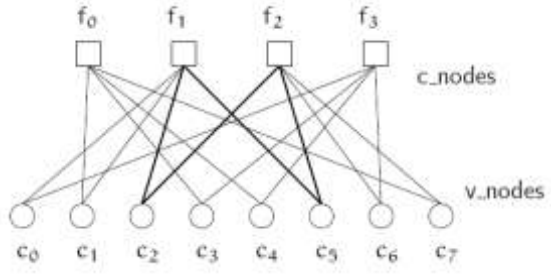
$$\mathbf{H} = \begin{bmatrix} 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \end{bmatrix} \qquad (1)$$

### B.Graphical representation

Tanner introduced an effective graphical representation for LDPC code. Not only provide this graph a complete representation of the code, they also help to describe the decoding algorithm.

Tanner graphs are bipartite graphs. That means that the nodes of the graphs are separated into two distinctive sets and edges are only connecting nodes of two different types. The two types of nodes in a tanner graph are variable nodes (v-nodes) and check nodes(c-nodes).

Example for such a tanner graph and represents the same code as the matrix in 1. The creation of such graph is rather straight forward. It consists of m check nodes (the number of parity bits) and n variable nodes (the number of bits in a codeword). Check node $f_i$ is connected to a variable node $c_j$ if the element $h_{ij}$ of H is a 1.




## IV. ALGORITHM FOR LDPC CODE

1. Create H matrix with equal number of ones in lows and columns in parity matrix such that [P/I]
2. From H matrix create G matrix such that [I/PT]
3. LDPC code is generated by multiplying G matrix with A bit input data.
4. Generated code is multiplied with H matrix. If output is zero, indicates no error.

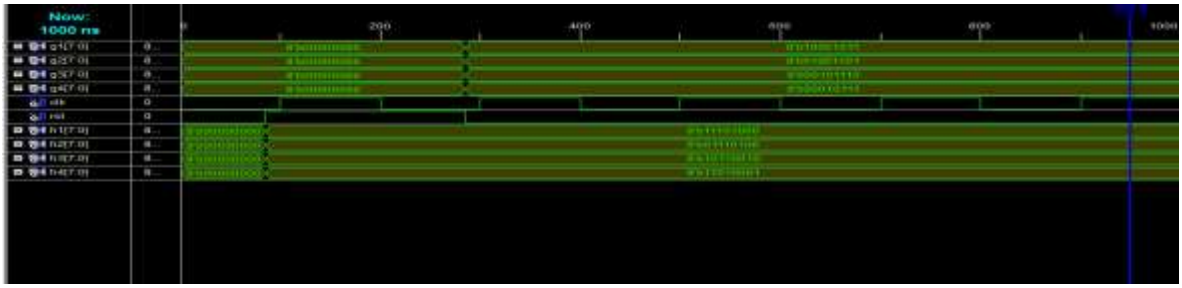
## V RESULTS AND DISCUSSIONS

1. Consider the H matrix as

$$\begin{bmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}$$

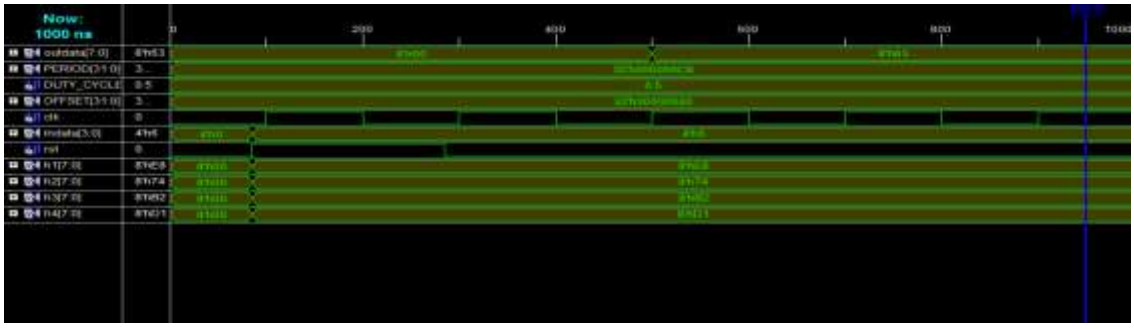
2. G matrix created as

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 \end{bmatrix}$$



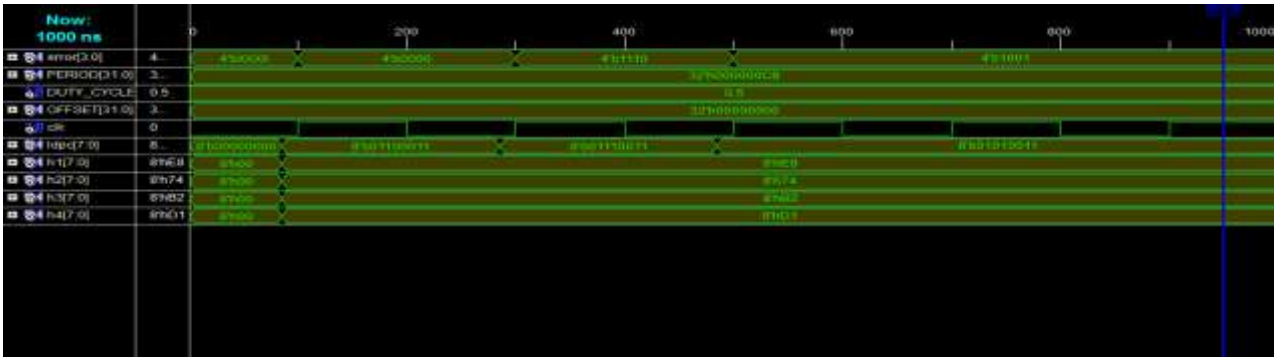
3. Multiplying (0110) with G matrix, LDPC code is generated as

$$[0\ 1\ 1\ 0\ 0\ 0\ 1\ 1]$$

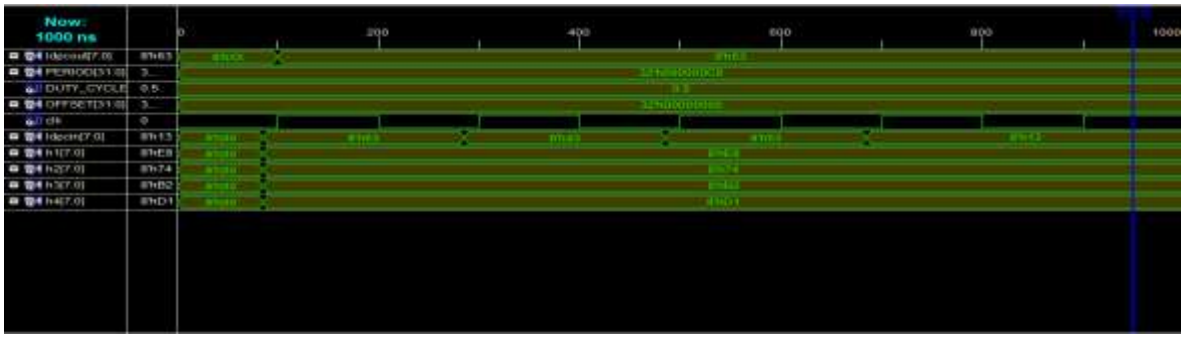


4. Multiplying generated code with H matrix gives output

$$\begin{bmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$



5. The correction for wrongly generated LDPC code can corrected using this method.



## VI.EXPERIMENTAL STUDY

A. *Error detection*

In order to estimate the error detection and correction of the proposed technique. We use advanced error correction method. The size of word can be assumed to 32 bit, both single and multiple fault can be detected and corrected.

B. *Cost per chip analysis*

The technique proposed in this paper aims to reduce the cost per chip by using coding technique , that allows us to save some of faulty memory chip with small defects instead of traditional rows and columns .The following assumption have been adopted :

1.All defective memory chip have only spot defects no global defects, wchich are those defects affecting complete sections of chip or wafer. For traditional techniques ,when no reduandant rows or column are included in the chip,any single spot defect will result in the chip being discarded.

2.Memory array is used for area calculation,which has been obtained by modeling in c programming language.

3.Each wafer can hold thousand chip without any redundancy .If redundant rows/columns are included on chip/wafer is reduced.

## VII. CONCLUSION

This paper presented a high level error detection and correction method called LDPC. The proposed code combines H matrix and G matrix so that multiple errors can be detected and corrected. Cost analysis of the proposed method is significantly better than matrix code, reed-muller code in both error detection and correction.

## REFERENCES


[1] Matrix Codes for Reliable and Cost Efficient Memory Chips, Costas Argyrides, *Member, IEEE*, Dhiraj K. Pradhan, *Fellow, IEEE*, and Taskin kocak IEEE transactions on very large scale integration (vlsi) systems, vol. 19, no. 3, march 2011

[2] ITRS 2002. [Online]. Available: http://public.itrs.net

[3] P. Shivakumar, M. Kistler, S. W. Keckler, D. Burger, and L. Alvisi,"Modeling the effect of technology trends on the soft error rate of combinational logic," in *Proc. Int. Conf. Dependable Syst. Netw. (DSN)*,2002, pp. 389–398.

[4] G. Cardarilli, A. Leandri, P. Marinucci, M. Ottavi, S. Pontarelli, M. Re,and A. Salsano, "Design of a fault tolerant solid state mass memory,"*IEEE Trans. Reliab.*, vol. 52, no. 4, pp. 476–491, Dec. 2003.

[5] B. Cooke, "Reed Muller Error Correcting Codes," *MIT UndergraduateJ. Math.*, vol. 1, pp. 21–26, 1999.

[6] P. A. Ferreyra, C. A. Marques, R. T. Ferreyra, and J. P. Gaspar, "Failure map functions and accelerated mean time to failure tests: New approaches for improving the reliability estimation in systems exposed to single event upsets," *IEEE Trans. Nucl. Sci.*, vol. 52, no. 1, pp. 494–500,Jan. 2005.

[7] P. Hazucha and C. Svensson, "Impact of CMOS technology scaling on the atmospheric neutron soft error rate," *IEEE Trans. Nucl. Sci.*, vol.47, no. 6, pp. 2586–2594, Dec. 2000.

[8] R. Hentschke, R. Marques, F. Lima, L. Carro, A. Susin, and R. Reis,"Analyzing area and performance penalty of protecting different digital modules with hamming code and triple modular redundancy," in *Proc.Symp. Integr. Circuits Syst. Des.*, 2002, pp. 95–100.

[9] J. Karlsson, P. Liden, P. Dahlgren, R. Johansson, and U. Gunneflo,"Using heavy-ion radiation to validate fault-handling mechanisms," *IEEE Trans. Microelectron.*, vol. 14, pp. 8–23, 1994.

[10] R. Reed, M. Carts, P. Marshall, C. J. Marshall, O. Musseau, P. Mc-Nulty, D. Roth, S. Buchner, J. Melinger, and T. Corbiere, "Heavy ion and proton-induced single event multiple upset," *IEEE Trans. Nucl. Sci.*, vol. 44, no. 6, pp. 2224–2229, Dec. 1997.

[11] N. Seifert, D. Moyer, N. Leland, and R. Hokinson, "Historical trend in alpha-particle induced soft error rates of the Alpha microprocessor," in *Proc. 39th Annu. IEEE Int. Reliab. Phys. Symp.*, 2001, pp. 259–265.

[12] M. Y. Hsiao, "A class of optimal minimum odd-weight column SEC-DED codes," *IBM J. Res. Development*, vol. 14, pp. 395–401, 1970.

[13] D. K. Pradhan and S. M. Reddy, "Error-control techniques for logic processors," *IEEE Trans. Comput.*, vol. C-21, no. 12, pp. 1331–1336, Dec. 1972.

[14] S. Satoh, Y. Tosaka, and S. A. Wender, "Geometric effect of multiple-bit soft errors induced by cosmic ray neutrons on DRAM's," *IEEE Electron Device Lett.*, vol. 21, no. 6, pp. 310–312, 2000.

[15] A. Dutta and N. A. Touba, "Multiple bit upset tolerant memory using a selective cycle avoidance based SEC-DED-DAEC code," in *Proc.IEEE VLSI Test Symp. (VTS)*, 2007, pp. 349–354.