Implementation of PMC to Enhance Memory Reliability against MCU

1T.Maheswari, 2V.Logeswari
1,2Assistant Professor/ECE
Nandha Engineering College (Autonmous)
tmaheswariece@gmail.com
vlogesivamece@gmail.com

***Abstract*—This paper presents a high level technique to protect SRAM memories against multiple upsets based on correcting codes. Transient multiple cell upsets (MCUs) are becoming major issues in the reliability of memories exposed to radiation environment. To prevent MCUs from causing data corruption, more complex error correction codes (ECCs) are widely used to protect memory, but the main problem is that they would require higher delay overhead. The only drawback of the existing DMC is that more redundant bits are required to maintain higher reliability of memory. The proposed technique used parity matrix code to assure reliability in presence of multiple bit flip and reduce more redundant bit and its correct more error compare to existing system.**

Index Terms- Parity algorithm, error correction codes, multiple cell upsets (MCU's), Memory.

I. INTRODUCTION

As CMOS technology scales down to nanoscale and memories are combined with an increasing number of electronic systems, the soft error rate in memory cells is rapidly increasing, especially when memories operate in space environments due to ionizing effects of atmospheric neutron, alpha-particle, and cosmic rays [1].

The general idea for achieving error detection and correction is to add some redundancy (i.e., some extra data) to a message, which receivers can use to check consistency of the delivered message, and to recover data determined to be corrupted. Error-detection and correction schemes can be either systematic or non-systematic: In a systematic scheme, the transmitter sends the original data, and attaches a fixed number of check bits (or parity data), which are derived from the data bits by some deterministic algorithm. If only error detection is required, a receiver can simply apply the same algorithm to the received data bits and compare its output with the received check bits; if the values do not match, an error has occurred at some point during the transmission.

In a system that uses a non-systematic code, the original message is transformed into an encoded message that has at least as many bits as the original message. The goal of error detection and correction code is to provide against soft errors that manifest themselves as bit-flips in the memory.

Several techniques are used nowadays to midi gate upsets in memories. For example, the Bose–Chaudhuri–Hocquenghem codes [8], Reed–Solomon codes [9], punctured difference set (PDS) codes [10], and matrix codes have been used to deal with MCUs in memories. But these codes require more area, power, and delay overheads since the encoding and decoding circuits are more complex in these complicated codes. Reed-Muller code [14] is another protection code that is able to detect and correct more error than a Hamming code. The main drawback of this protection code is its high area and power penalties.

Hamming Codes are largely used to correct Single Error Upsets (SEU's) in memory due to their ability to correct single errors with reduced area and performance overhead [13]. Though excellent for correction of single errors in a data word, they cannot correct double bit errors caused by single event upset. An extension of the basic SEC-DED Hamming Code has been proposed to form a special class of codes known as Hsiao Codes to improve the speed, cost and reliability of the decoding logic [14].

One more class of SEC-DED codes known as Single-error-correcting, Double-error-detecting Single-byte-error-detecting SEC-DED-SBD codes were proposed to detect any number of errors affecting a single byte. These codes are more suitable than the conventional SEC-DED codes for protecting the byte-organized memories [15][16]. Though they operate with lesser overhead and are good for multiple error detection, they cannot correct multiple errors. There are additional codes such as the single-byte-error-correcting, double-byte-error-detecting (SBC-DBD) codes, double-error-correcting, triple error-detecting (DEC-TED) codes that can correct multiple errors as discussed in [10].

The Single-error-correcting, Double-error-detecting and Double-adjacent-error-correcting (SEC-DED-DAEC) code provides a low cost ECC methodology to correct adjacent errors as proposed

in [12]. The only drawback with this code is the possibility of miscorrection for a small subset of multiple errors. More recently, in decimal matrix codes (DMC) are proposed to provide enhanced memory reliability. In this method decimal algorithm (decimal integer addition and decimal integer subtraction) is used to detect errors. The only drawback of the proposed DMC is that more redundant bits are required to maintain higher reliability of memory.

This paper presents a new technique, Parity Matrix Code (PMC). A parity algorithm (matrix multiplication and matrix addition) is used to detect and correct multiple errors which require less number of redundant bits compared to decimal matrix codes (DMC). This paper corrects maximum number of errors in memories. This paper is divided into the following sections. The proposed PMC introduced and its encoder and decoder circuits are present in Section II. Results are given in section III and paper is concluded in the section IV.

II. PMC TECHNIQUE

This technique uses a parity algorithm (matrix multiplication and matrix addition) to detect and correct multiple errors which require less number of redundant bits compared to decimal matrix codes (DMC). It corrects maximum number of errors in memories.

A. PARITY MATRIX CODE

The parity matrix 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 parity matrix codes code itself is no different to any other block code. Indeed existing block codes can be successfully used with the parity matrix codes 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 parity matrix 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 parity matrix codes and classical block code is how they are decoded. For large block size, parity matrix codes are commonly constructed by first studying the behavior of decoders. Parity matrix codes decoders shown to have a noise threshold below which the decoding is reliably achieved and above which the decoding is not achieved.

The construction of specific parity matrix codes 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 parity matrix codes. The desirable property of parity matrix codes 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 rates 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. Figure 1 shows the block diagram of parity matrix code.

The main blocks of parity matrix codes are variable node unit (VNU), check node unit (CNU) and sparse matrix. Sparse matrix contains the connections between VNUs and CNUs.

In the proposed method sparse matrix will be reduced and in such a way that all the process will be done in parallel. Input data which need to be send will be divided into parts and will be given to sparse matrix so every part will be processed in parallel so total system complexity will be decreased.

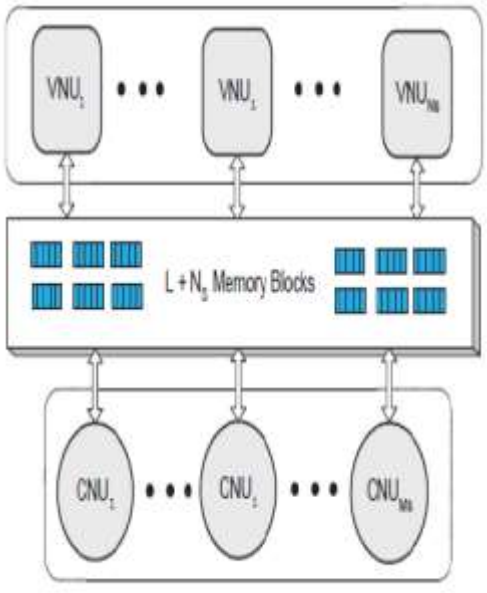

Figure 1. PMC Block Diagram

A.METHODOLOGY

LDPC is a linear error correcting code for transmitting a message over a noisy transmission channel.

1. LDPC ALGORITHM

A code word **c** is generated as

$$C = kG$$

where k is the vector of information bits and G is the generator matrix. A valid codeword can be verified using

$$HCT = 0$$

Where H is the parity check matrix. If the result in correction procedure should be used in this case. The Bit flipping method uses a vector, called syndrome , which is computed as

$$S = HYT,$$

Where Y is the invalid codeword. The syndrome indicates which row in the H is not zeroed by vector Y and some bits have to be repaired in the decoder .If the parity check matrix has low size, we can find an error floor of the LDPC code, where one erroneous bit is repaired and BER is close to zero or is zero.

B. PMC ENCODER

Encoder uses generator matrix to encode the information bits in to the code word. Both generator and parity check matrix are in inter related, parity check matrix is given by

$$H = [A \mid In\text{-}k]$$

and generator matrix is given by

$$G = [Ik \mid AT]$$

The advantage of LDPC codes is their error correcting performance. This is a reason why LDPC codes have been selected for many applications.

Initially parity check matrix is generated, using that generator matrix is created by Gaussian elimination method. There are two types of parity matrices in LDPC coding one is Regular matrix and another one is irregular matrix.

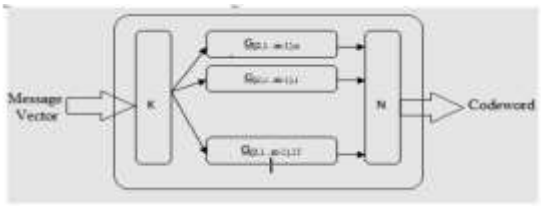

Figure 2. Encoder Block Diagram

Now the information message bits are encoded by multiplying it with above generator matrix i.e C = [U][G] to obtain the codeword. The above figure 2 shows the encoder block diagram. Each structure labeled G{0,1,,,m-1},i are XOR structures performs modulo-2 operations on the incoming message bits and the resultant code words will be of N bits.

In encoding as a first step sparse matrix H matrix will be created as the interconnection between the VNUs and CNU. Using this H matrix G matrix will be created using row column operations in the H matrix using this G matrix encoded LDPC code will be generated by multiplying divided input data parts with G matrix. This data will be combined and transmitted over the network to the receiver.

C. PMC DECODER

A decoder is a device which does the reverse operation of an encoder, undoing the encoding so that the original information can be retrieved. The same method used to encode is usually just reversed in order to decode 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.

III RESULT AND DISCUSSION

1. CONSIDER THE H MATRIX AS

This matrix shows both identity matrix and data bits called as an H matrix. The H matrix is converted to G matrix. Here column of H matrix is converted into row as shown in G martrix. In this conversion the result are shown in figure 2.

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

Figure 3. H Matrix to G Matrix

2. H-MATRIX CREATED AS G MATRIX

Here H-matrix is created as G matrix.

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

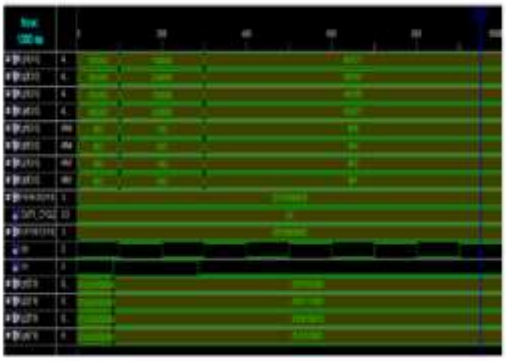

Figure 4. Transpose for G Matrix

3. ENCODING PROCESSORS

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

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

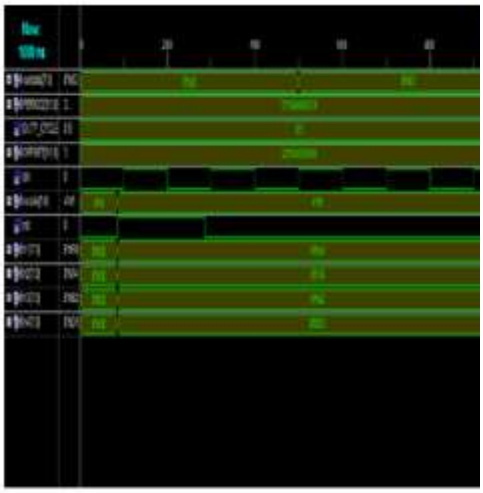

Figure 5. Multiplying (0110) with G matrix generated in 4-bit Encoding.

Multiplying (0110) with G matrix generated in 4-bit Encoding.

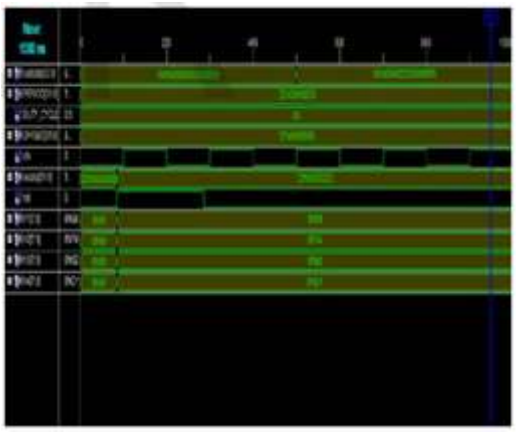

Figure 6. 32 Bit Encoding

Multiplying (0110) with G matrix generated in 32-bit Encoding. In this paper the main aim is to reduce redundant bits and increase correction capability.

Table 1 shows the Comparison of proposed results with the existing technique

Parameter	DMC	PMC	
	32 bit existing	16 bit proposed	32 bit proposed
No .of Errors Corrected	12 Bits	12 Bits	12 Bits
No. of Redundant BITS	12 Bits	12 Bits	12 Bits

Table 1. Comparison of results with the existing technique.

IV CONCLUSION

The Parity based matrix codes is proposed to increase the error handling capability. Proposed method will increase number of detectable and correctable errors and will decrease the total number of extra bits need to be stored to detect the errors. So, proposed method will increase the data accuracy of the memory.

Drawback of the existing system is rectified in this method. Encoding of the proposed system is designed and implemented. The decoding will be implemented and will be compared with existing system results.

REFERENCES

[1] Jing Guo, Liyi Xiao, Member, IEEE, Zhigang Mao, Member, IEEE, and QiangZhao,"Enhanced memory reliability against multiple cell upsets using Decimal Matrix Code" IEEE Trans. Very Large ScaleIntegr. (VLSI) Syst., Vol. 22, no. 1, pp.127-135, Mar 2013.

[2] D. Radaelli, H. Puchner, S. Wong, and S. Daniel, "Investigation of multi-bit upsets in a 150 nm technology

SRAM device," IEEE Trans.Nucl. Sci., Vol. 52, no. 6, pp. 2433–2437, Dec. 2005.
[3] E. Ibe, H. Taniguchi, Y. Yahagi, K. Shimbo, and T. Toba, "Impact of scaling on neutron induced soft error in SRAMs from an 250 nm to a 22 nm design rule," IEEE Trans. Electron Devices, Vol. 57, no. 7, pp. 1527–1538, Jul. 2010.
[4] C. Argyrides and D. K. Pradhan, "Improved decoding algorithm for high reliable reed muller coding," in Proc. IEEE Int. Syst. On Chip Conf., Sep. 2007, pp. 95–98.
[5] A. Sanchez-Macian, P. Reviriego, and J. A. Maestro, "Hamming SEC-DAED and extended hamming SEC-DED-TAED codes through selective shortening and bit placement," IEEE Trans. Device Mater. Rel., to be published.
[6] S. Liu, P. Reviriego, and J. A. Maestro, "Efficient majority logic fault detection with difference-set codes for memory applications," IEEETrans. Very Large Scale Integer. (VLSI) Syst., Vol. 20, no. 1, pp. 148–156, Jan. 2012.
[7] M. Zhu, L. Y. Xiao, L. L. Song, Y. J. Zhang, and H. W. Luo, "New mix codes for multiple bit upsets mitigation in fault-secure memories," Microelectron. J., Vol. 42, no. 3, pp. 553–561, Mar. 2011.
[8] R. Naseer and J. Draper, "Parallel double error correcting code design to mitigate multi-bit upsets in SRAMs," in Proc. 34th Eur. Solid-StateCircuits, Sep. 2008, pp. 222–225.
[9] G. Neuberger, D. L. Kastensmidt, and R. Reis, "An automatic technique for optimizing Reed-Solomon codes to improve fault tolerance in memories," IEEE Design Test Comput., Vol. 22, no. 1, pp. 50–58, Jan.–Feb. 2005.
[10] P. Reviriego, M. Flanagan, and J. A. Maestro, "A (64,45) triple error correction code for memory applications," IEEE Trans. Device Mater.
Rel., Vol. 12, no. 1, pp. 101–106, Mar. 2012.
[11] S. Baeg, S. Wen, and R. Wong, "Interleaving distance selection with a soft error failure model," IEEE Trans. Nucl. Sci., Vol. 56, no. 4, pp. 2111–2118, Aug. 2009.
[12] K. Pagiamtzis and A. Sheikholeslami, "Content addressable memory (CAM) circuits and architectures: A tutorial and survey," IEEE J.Solid-State Circuits, Vol. 41, no. 3, pp. 712–727, Mar. 2003.
[13] S. Baeg, S. Wen, and R. Wong, "Minimizing soft errors in TCAM devices: A probabilistic approach to determining scrubbing intervals," IEEE Trans. Circuits Syst. I, Reg. Papers, Vol. 57, no. 4, pp. 814–822, Apr. 2010.
[14] P. Reviriego and J. A. Maestro, "Efficient error detection codes for multiple-bit upset correction in SRAMs with BICS," ACM Trans. DesignAutom. Electron. Syst., Vol. 14, no. 1, pp. 18:1–18:10, Jan. 2009.
[15] C. Argyrides, R. Chipana, F. Vargas, and D. K. Pradhan, "Reliability analysis of H-tree random access memories implemented with built in current sensors and parity codes for multiple bit upset correction," IEEETrans. Rel., Vol. 60, no. 3, pp. 528–537, Sep. 2011.
[16] C. Argyrides, D. K. Pradhan, and T. Kocak, "Matrix codes for reliable and cost efficient memory chips," IEEE Trans. Very Large Scale Integr.(VLSI) Syst., Vol. 19, no. 3, pp. 420–428, Mar. 2011.
[17] C. A. Argyrides, C. A. Lisboa, D. K. Pradhan, and L. Carro, "Single element correction in sorting algorithms with minimum delay overhead," in Proc. IEEE Latin Amer. Test Workshop, Mar. 2009, pp. 652–657.
[18] Y. Yahagi, H. Yamaguchi, E. Ibe, H. Kameyama, M. Sato, T. Akioka, and S. Yamamoto, "A novel feature of neutron-induced multi-cell upsets in 130 and 180 nm SRAMs," IEEE Trans. Nucl. Sci., Vol. 54, no. 4, pp. 1030–1036, Aug. 2007.
[19] 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., Vol. 59, no. 4, pp. 845–850, Aug. 2012.

