International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395 -0056
Volume: 03 Issue: 08 | Aug-2016 www.irjet.net p-ISSN: 2395-0072

Comparative Analysis of DMC and PMC on FPGA

Gnanajyothi R[1], Mr Ramana Reddy K V[2], Dr. Siva Yellampalli[3]

[1] *Student, Mtech13, UTL VTU regional center , Bengaluru.*
[2] *Assistant professor, UTL VTU regional center , Bengaluru.*
[3] *Principal and HOD, UTL VTU regional center, Bengaluru.*

Abstract— *Steady reductions in device sizes by scaling it to smaller dimensions has resulted in very dense high speed memory cells and also increase of fault occurrences in the memory. Multiple cell upsets are becoming a major issues in memory exposed to extreme radiation environments like spacecraft and avionics electronics and terrestrial environments. Memory failure rates are increasing significantly, posing a reliability concern for many applications. In order to make memory cells as fault tolerant some error correction codes have been widely used to protect memory cells. Since the encoding and decoding circuits are more complex, these codes require more area, power and delay overheads. Recently Decimal matrix codes (DMCs) has been proposed to efficiently correct MCUs per word with a low decoding delay in which one word is divided into multiple rows and multiple columns in logical. The drawback of this method is that it requires more number of redundant bits. This paper presents a novel parity matrix code (PMC) based encoding and decoding method to assure reliability in presence of multiple bit upsets. PMC reduces number of redundant bits and corrects more errors compared to existing systems. The experimental results showed that the proposed method has better protection level against large MCUs.*

Key Words—Multiple Cell Upsets (MCUs), Decimal Matrix Codes (DMCs), Parity Matrix Code (PMC).

I. INTRODUCTION

Major advancement in complementary metal oxide semiconductor technology has given way to scaling of device dimensions to nanoscale. Because of scaling of device dimensions, there is a major issue in the accuracy of memory cells open to scattering (radiation) atmosphere. The velocity of soft errors in the memory cells is increased at faster rate because of neutron, cosmic rays and alpha particles ionizing effects in the atmosphere, while memories perform operations in space atmosphere. In the memory application major concern is reliability of Multiple Cell Upsets [1].

The generic solution for finding the errors and fixing is to do addition of some excess bits to the messages at the acceptor end and analyses for firmness of the messages delivered and selection of the data that is conclusive to be corrupted.
Some error correction codes are widely used to protect memories to make the memory cells defect tolerant. Compared to Hamming code memory protection codes such as punctured difference set codes (PDS), Reed Muller code are able to find and correct additional errors in the memory. The major disadvantage of these methods is its more area and power overhead. Hamming codes are capable of correcting Single error upset through reduced area and operation performance overhead. When double bit errors are caused by single event upset they are not capable to correct it. SEC-DED codes known as Single-error-correcting, Double-error- detecting are proposed to find any number of errors in a single byte. Although they produce minor overheads and are worth for detection of multiple errors, multiple errors cannot be corrected by them.

The Single-error-correction, Double-adjacent-error-correction, and Double-error-detection codes provide error correction methods to fix the errors in adjacent cells at low cost, but the disadvantage of this code is, for a small compartment of numerous errors there is possibility of miss-correction. The technique of arranging cells physically to logically separate the bits into different physical words in the same words is called interleaving techniques. But this cannot be practically used due to tight coupling of hardware structures.

To efficiently correct MCUs 2-D matrix codes are designed to correct MCUs per word with a minimum decoding delay. In 2-D matrix codes one word is logically divided into multiple rows and multiple columns. Bits per row are protected using Hamming code while each column is added by parity code. In all the cases matrix codes are capable of correcting only two errors.

II .BACKGROUND

In existing method to provide enhanced reliability of memory Decimal matrix code is used which is based on divide-symbol and arrange matrix logic. It makes use of decimal algorithm to detect errors. By using DMC the capability of error finding is maximized and memory reliability is enhanced. The only short come of DMC is increased number of extra bits (redundant bits).Decimal matrix code disadvantage can be overcome by introducing Novel Parity Matrix Code which uses matrix algorithm, to fix errors and it requires fewer number of check bits compared to DMC. It has minimum performance overhead.

I. DMC Method

A. Design of fault tolerant memory

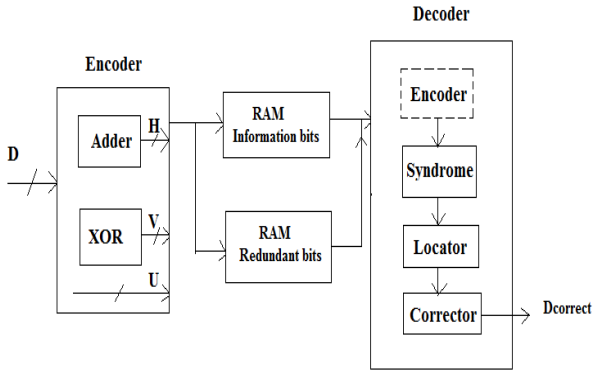

Fig. 1: Design of memory protected with DMC [2].

The schematic shown in Fig. 1 is the design of defect tolerant memory. Firstly, the DMC encoder is fed with information bits **D** during the process of encoding (write), and then DMC encoder generates horizontal and vertical redundant bits **H** and **V**. The DMC redundant bits (code word) obtained is stored in the memory on completion of encoding process. In the memory if MCUs occurs, during decoding process (read) it can be fixed. The DMC proposed has higher defect tolerant capacity because of the benefits of decimal algorithm with lesser performance overheads. Error reuse technique in defect tolerant memory is introduced to reduce extra circuits i.e., area overheads.

B. DMC Encoder

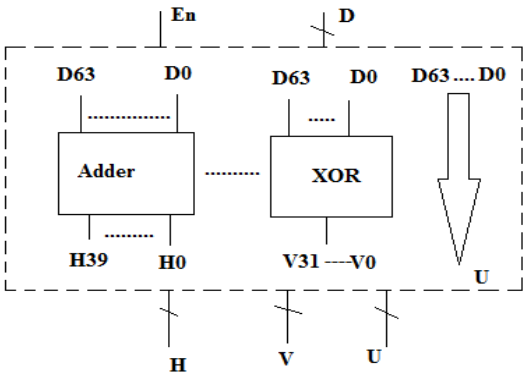

Fig. 2. DMC Encoder design (64-bits) [2].

The purpose of DMC operation is to first divide the symbol and organize the symbol to matrix form i.e. if the DMC gets N bit word it is divided into $\mathbf{k_b}$symbols with $\mathbf{m_b}$bits where $\mathbf{N = k_b \times m_b}$, and those symbols are organized in 2-D matrix of $\mathbf{k_b = k_{b1} \times k_{b2}}$, where $\mathbf{k_{b1}}$ and $\mathbf{k_{b2}}$ represents the rows and columns in matrix logically. Secondly, addition of decimal integer is performed by selecting symbols per row to generate horizontal bits **H**. N-bit word divided into symbols are noticed as

decimal integers. Thirdly, to obtain vertical bits **V** among the bits per column logical operation is performed. It can be noted that the matrix arrangement and divide –symbol are logically achieved instead of physical. Hence, the designed DMC need not change the memory structure physically.

The DMC encoder is fed with 64-bit word, this 64 bit word is divided into sixteen symbols each of four bits. $\mathbf{k_{b1}}$ = 2 two rows and k_{b2} = 8 columns is chosen concurrently. Horizontal extra (redundant) bits are from $\mathbf{H_0–H_{39}}$. By doing addition of decimal integer horizontal extra (redundant) bits **H**can be obtained, following is the decimal integer addition:

$H_4H_3H_2H_1H_0 = D_3D_2D_1D_0 + D_{11}D_{10}D_9D_8$

$H_9H_8H_7H_6H_5 = D_7D_6D_5D_4 + D_{15}D_{14}D_{13}D_{12}$

and equivalently for the horizontal extra (redundant) bits $H_{14}H_{13}H_{12}H_{11}H_{10}$ and $H_{19}H_{18}H_{17}H_{16}H_{15}$, where symbol plus "+" exhibits addition of decimal integer. The vertical excess (redundant bits) **V** is obtained as follows:

$V_0 = D_0 \oplus D_{32}$

$V_1 = D_1 \oplus D_{33}$

and similarly the rest of vertical redundant bits are obtained.

Information bits			Horizontal bits		
D_{31}	……..	D_0	H_{19}	……..	H_0
D_{63}	……..	D_{32}	H_{39}	……..	H_{20}
V_{31}	……..	V_0			
Vertical bits					

Fig. 3.Logical arrangement of 64-bit information.

C. DMC Decoder

To obtain the corrected word decoding process is required. For example, firstly, redundant bits $\mathbf{(H_4H_3H_2H_1H_0)'}$ and $\mathbf{V_0'–V_3'}$ received are created by the information bits **D'** received. Secondly, the horizontal and vertical syndrome redundant bits$\mathbf{\Delta H_4H_3H_2H_1H_0}$ and $\mathbf{S_3– S_0}$ can be computed as shown below:

$\mathbf{\Delta H_4H_3H_2H_1H_0 = (H_4H_3H_2H_1H_0)' – H_4H_3H_2H_1H_0}$

$\mathbf{S_0 = V_0' \oplus V_0}$

and similarly XOR operation is performed for the rest of vertical syndrome bits and the symbol minus "–" represents subtraction of decimal integer.

When the operation of $\mathbf{\Delta H_4H_3H_2H_1H_0}$ and $\mathbf{S_3–S_0}$ are equal to zero the code word saved in the memory has original information bits in symbol 0 where no error has occurred. When the operation of $\mathbf{\Delta H_4H_3H_2H_1H_0}$ and $\mathbf{S_3– S_0}$ are not zero, the errors induced in symbol 0 are detected and placed in the same symbol, and then by $\mathbf{D_{0correct} = D_0 \oplus S_0}$ these errors can be corrected.

The structure of DMC decoder is as shown in Fig. 4, it is made up of the above sub modules, the sub modules are defect calculator, error locator, and error corrector each performs a specific task.

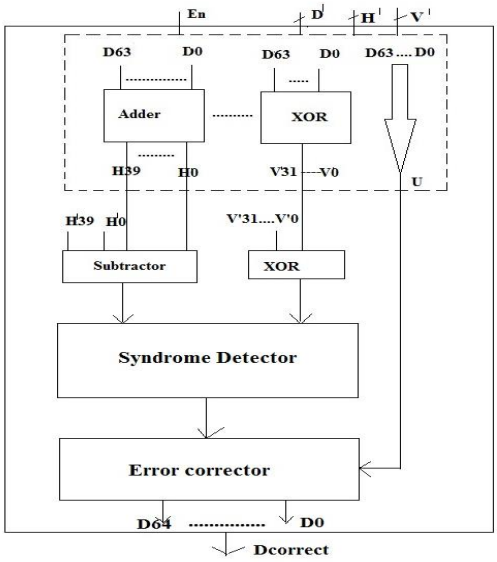

Fig. 4. 64 bit Decoder design [2].

From the Fig. 4. it can be noticed that recalculation of redundant bits from received information bits **D'** and in order to attain the syndrome bits **ΔH**and**S** the recalculated redundant bits is compared to the original set of redundant bits. Then error locator is used to locate errors when some error occurs by using **ΔH**and **S**to detect and locate. Lastly, the errors are corrected by inverting the values of error bits in the error corrector.

By using its encoder as part of decoder the circuit area of DMC is reduced. It is called as Error Reuse Technique (ERT). The whole encoding and decoding process is not disturbed even when ERT is used. It reduces the area overhead of DMC. The whole circuit area of DMC can be reduced by using existing circuits of encoder as a result of ERT.

Table 1: Structure of Error reuse Technique

Extra circuit	En Signal		Function
	Read signal	Write Signal	
Encoder	0	1	Encoding
	1	0	Compute Syndrome bits

Form the above table we can come to know that to decide whether encoder to use as part of decoder or encoder signal *En* is use i.e. the *En* signal under the control of read and write signals in memory used for discriminating the encoder from the decoder. Hence, the DMC encoder only executes the encoding operations in the encoding process (write). This encoder in decoding process (read) is employed for calculation of the syndrome bits in the decoder. This clearly shows that there is substantial reduction in area overhead of extra circuits.

D. Drawbacks of DMC

Though decimal matrix code maximizes the error detection and correction capability, it results in errors in decoding when the following factor occurs:
a)When upset occurs in both the rows.b)When the decimal integer sum value of both the original and that of flipped data is same.And also DMC requires large number of redundant bits to detect and correct errorto overcome this problem PMC is proposed.

II. PMC METHOD

A. *Design of PMC*

Parity matrix codes are the one that contains small number of non-zero values. They are block of code with parity check matrices. Decoding complexity increases linearly with length of code and minimum distance by the assurance of scantiness of H matrix. A parity matrix code itself no different from other block of codes apart from H sparse requirement. Actually existing block codes can be successfully used with the parity matrix codes iterative decoding algorithm can be actually used with existing block codes successfully if it is possible to be represented by a sparse parity-check matrix. However, finding a sparse parity-check matrix for an existing code is not possible practically. Design of parity matrix is done as follows, First sparse parity-check matrix is constructed and afterwards generator matrix is determined for code. Massive variation separating classical block code and parity matrix codes is how they are decoded.

To reduce large number of redundant bits for detection and correction of errors Parity Matrix Codes (PMC) are utilized. This new technique detects and corrects multiple errors with less number of redundant bits compared to Decimal matrix code by using parity algorithm (matrix multiplication and matrix addition). Maximum number of errors in memories is corrected by using PMC and also consumes less power, delay and area than Decimal matrix Code. Consider a 64 – bit word to which the Parity Matrix Code is activated and verified with the Decimal Matrix Code results.

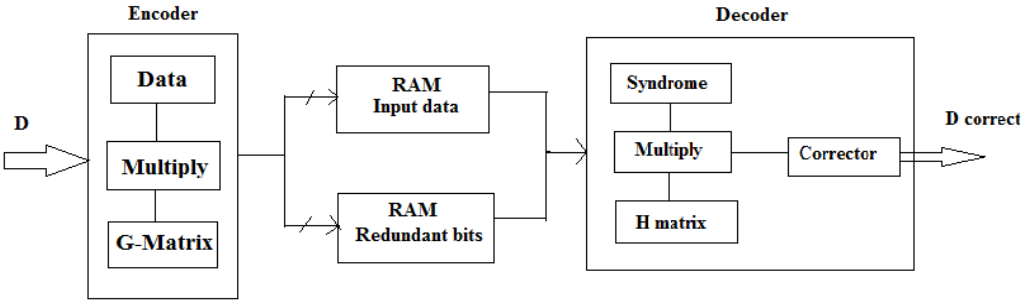

Fig. 5. PMC Block diagram [1]

B. *PMC Encoder*

Let's consider an n-bit information bit and similar to the decimal matrix code the 64-bit information is divided into $\mathbf{k_b}$ symbols of $\mathbf{m_b}$ bits each, i.e. if a 64-bits information is considered then divide the word into 32 symbols of 2 bits each. Then each data symbols are represented in column vectors. Represent each parity bit with a column vector containing a one in the row corresponding to every data bits included in the computation and a zero in all other rows. Later a generator matrix [G] is created, by organizing the column vectors into a 1×m matrix such that the columns in a code word match their corresponding bits in an order. If the bits in the columns are arranged in any other order it will just change the locations of bits in the code word. Lastly the encoder encodes the data by multiplying
it with the generator matrix. This encoded data is stored in the memory. When exposed to harmful radiations like gamma rays, x-rays etc the data/information stored in the memory may be corrupted. Example of G-Matrix:

$$G = \left[\begin{array}{cc|cc} 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 \end{array}\right]$$

C. PMC Decoder

A parity check matrix **H** is composed such that row one contains one's in the position of thefirst parity bit and all of the data bits that are included in its parity calculation. Row 2 contains 1s in the position of the second parity bit and all of the data bits that are included in its parity calculation. Row 3 contains 1s in the position of the third parity bit and all of the data bits that are included in its parity calculation and so on. Multiplying the encoded data with the **H** will result in a syndrome generation. The syndrome has two useful properties. The encoded data is error free if the syndrome is all zeros. Flipping the encoded bit that is in the position of the column in **H** that matches the syndrome will result in a valid code word, if the syndrome has non-zero value. Thus the errors are detected and corrected using parity matrix code.

$$H = \left[\begin{array}{cc|cc} 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{array}\right]$$

D. Advantages of PMC

By using PMC we can reduce the number of redundant bits compared to DMC. It increases error handling capability with less number of redundant bits.Consumption of power in PMC is less when compared to DMC.

III. SIMULATION RESULTS

Coding: Verilog
Simulation: Xilinx ISE Design Suite

A. DMC Simulation Results

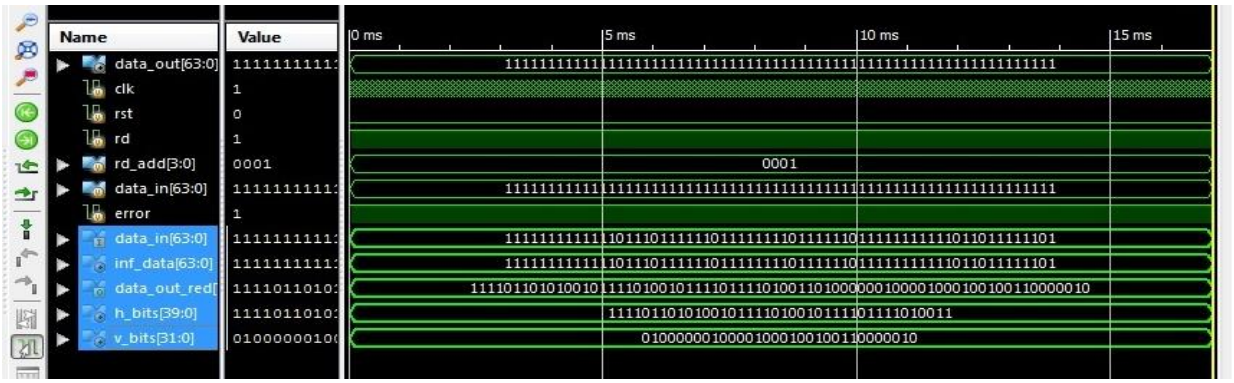

Fig. 6. Simulation of DMC

B. PMC Simulation Results

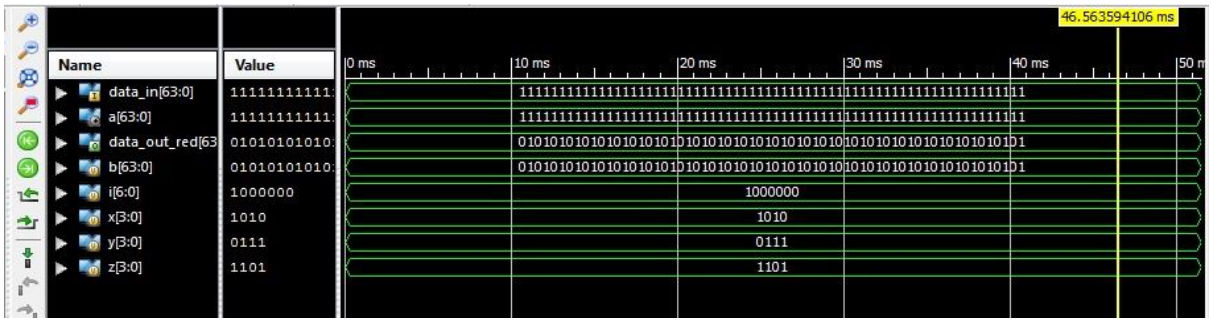

Fig. 7. Simulation of encoder

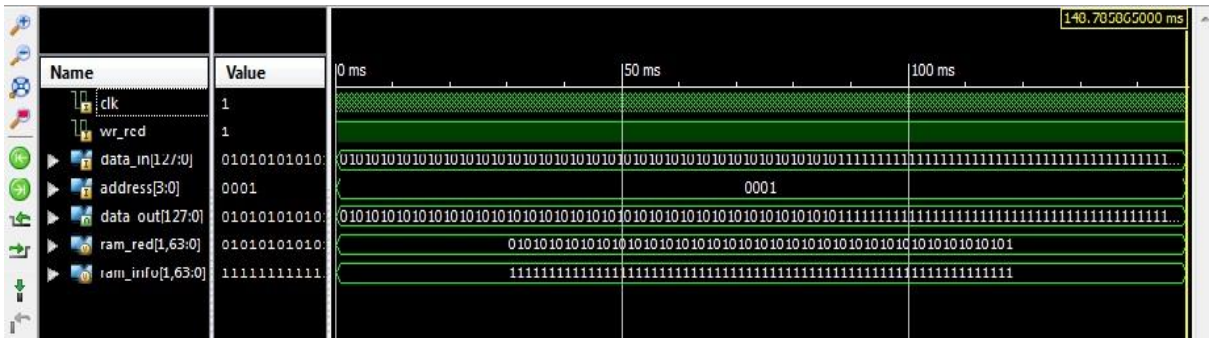

Fig.8. Simulation of RAM Block

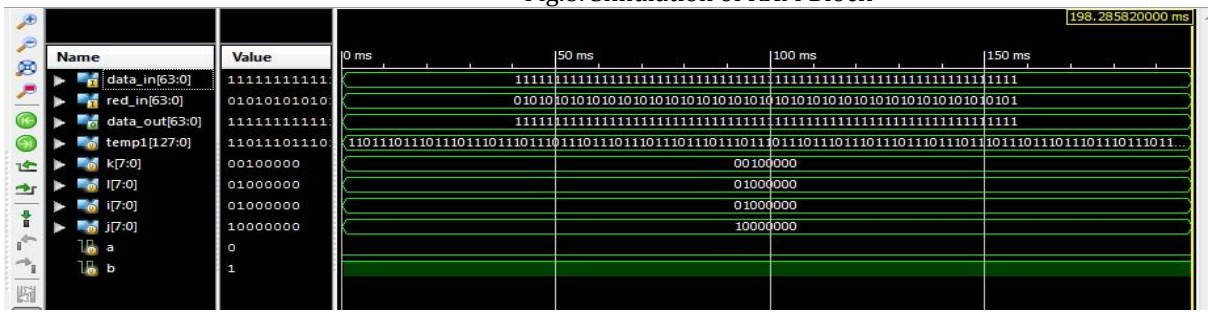

Fig. 9.Simulation of Decoder

IV. CONCLUSION

Many Error Correction Codes are used to correct and detect the errors to make the memory free from faults. Error correction and Detection capability and overheads vary based on the ECCs selected.

This project is proposed to show that PMC has superior protection level for memory against that of DMC. PMC can detect and correct maximum number of errors compared to DMC. The drawback of existing system DMC is that limited error correction capability and more redundant bits, it can be overcome by using PMC. Various parameters are compared for DMC and PMC to show that PMC has the superior protection level for memories. The table below shows the comparison of DMC and PMC.

Table 2. Comparison of DMC and PMC

Parameters	DMC(64 bits)	PMC(64 bits)
Number of redundant bits	72 bits	64 bits
Total number of bits used	136 bits	128 bits
Number of errors detected and corrected	9 bits	32 bits
Total power	33mW	32mW

REFERENCES

[1] *Neha Ibrahim, Vinoj P.G. and Sunil Jacob*, "A Novel Methodology for Detection and Correction of errors in memory using Parity Matrix Code (PMC)", volume 5, Issue 8, pp 350-360, August-2015.
[2] *Jing GuoLiyi Xiao, Zhigang Mao and Qiang Zhao*, "Enhanced Memory ReliabilityAgainst Multiple Cell Upsets Using Decimal Matrix Code", IEEE Transactions onVery large scale integration (VLSI) systems, VOL. 22, NO. 1, JANUARY 2014.
[3] *P.Reviriego, J. Martinez, S.Pontarelli, J.A.Maestro*, "A method to Design SEC-DED-DAEC codes with optimized Decoding", IEEE Transactions on Device and MaterialsReliability, volume 14,Issue 3, pp. 884-889, 2014.
[4] *A. Sanchez-Macian, P. Reviriego, and J. A. Maestro*, "Hamming SEC-DAED and extended hamming SEC-DED-TAED codes through selective shortening and bitplacement," *IEEE Trans. Device Mater. Rel.*, volume 14, Issue 1, pp. 574-576, 2014.
[5] *Y Cui, M. Lou,J. Xiao, X. Zhang,S. Shi,P. Lu*,"Research and Implementation of SEC-DED Hamming code algorithm", TENCON 2013 IEEE Region 10 Conference(31194),Year 2013, pp. 1-5, DOI:10.1109/TENCON.2013.6718953.
[6] *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.
[7] *S. Liu, P. Reviriego, and J. A. Maestro*, "Efficient majority logic fault detection withdifference-set codes for memory applications," IEEETrans. Very Large Scale Integration (VLSI)Syst., vol. 20, no. 1, pp. 148–156, Jan. 2012.
[8] *M. Zhu, L. Xiao, S. Li, Y. Zhang,* "Efficient Two- Dimensional Error Codes for Multiple Bit Upsets Mitigation in Memory", Defect and fault Tolerance in VLSISystems(DFT), 2010 IEEE 25th International Symposium, pp. 129-135, Year 2010.
[9] *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.
[10] *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.