International Journal of Electrical and Electronics Research ISSN 2348-6988 (online)
Vol. 3, Issue 3, pp: (48-54), Month: July - September 2015, Available at: www.researchpublish.com

REDUCING REDUNDANT BITS AND ENHANCED MEMORY RELIABILITY USING DECIMAL MATRIX CODE

[1]V.Vithya, [2]Dr. P. Sakthivel

[1]M.E- Embedded System Technologies, Velalar College of Engineering and Technology, Erode, India
[2]M.E. Ph.D., Associate Professor (EEE), Velalar College of Engineering and Technology, Erode, India

Abstract: **Memories are affected by errors in radiation environment it causes data corruption. Some of the error correcting codes are widely used to protect the memory from the errors but the main disadvantage is this codes make encoder, decoder very complex and lead to delay and area overhead, so to maintain the memory reliability, it necessary to protect memory by using protection codes. Decimal Matrix Code (DMC) is used to correct the error in the memory, protect the memory from the Multiple Cell Upset (MCU) and it maintain the memory reliability. This coding technique is maximize the error correction capability, this is the advantage of this code, and it uses Encoder Reuse Technique (ERT) to minimize the area overhead. ERT uses DMC encoder to be a part of decoder and it reduces the redundant bits.**

Keywords: **Multiple Cell Upset (MCU), Decimal Matrix Code (DMC), Encoder Reuse Technique (ERT), Memory.**

I. INTRODUCTION

Single bit upset is a major issue in the memory reliability also multiple cell upset became a major problem in memory. some error correction codes are widely used to protect the memory. Reed Solomon codes, punctured difference set codes (PDS) have been used to protect the memory from the MCU, but these codes require more memory, power, area, and delay overhead. It make encoder and decoder part as very complex and complicated.

The Interleaving Technique has been used to stop the MCU, but it not suitable in the Content addressable memory (CAM). Built-in current sensors method can only correct two error in word. Matrix codes are used to protect the memory from the MCU per word very efficiently. In this a word is divided into multiple rows and columns in the logical method. Hamming code is protecting bits per row, when parity code is added in each column. Advantage of this code is lower delay overhead when compared to other codes, disadvantage is it correct only two error in all cases.

The DMC method is uses decimal algorithm for detect the error. It protect the memory from the MCU, and it maintain the memory reliability. It uses ERT to minimize the area. Advantage of using decimal algorithm is maximize the error correction capability, so the reliability of memory is enhanced also DMC recover the memory from soft error rate.

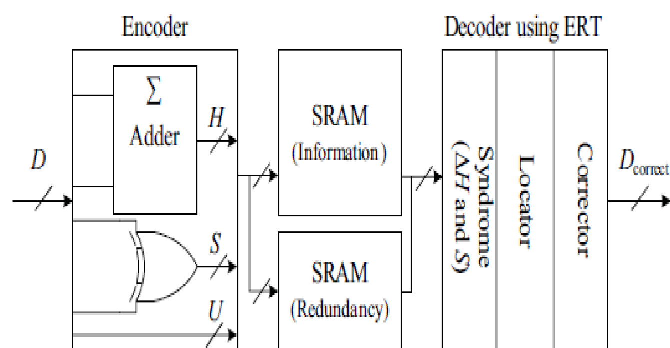

Figure.1. Proposed schematic of fault-tolerant memory protected with DMC

II. PERFORMANCE OF DMC

In this section, DMC is proposed to assure the reliability in the presence of MCUs with reduced performance overheads, and a 32bit word is encoded and decoded as an example based on the proposed technique.

A. Schematic of Fault-Tolerant Memory:

From the fig.1 we know the process of encoding and decoding technique. The information bits D are fed into the DMC encoder while encoding (write) process. From the DMC encoder can obtain the horizontal and vertical redundant bits H, V respectively. The DMC code word is stored, which is obtained from the encoder, after completing the encoder process. If MCUs occur in the memory, these errors can be corrected in the decoding (read) process. The proposed DMC has higher fault-tolerant capability with lower performance overheads.

B. DMC Encoder:

The divide-symbol and arrange-matrix ideas are performed in the first step of DMC, i.e., the N-bit word is divided into k symbols of m bits (N=k * m), and these symbols are arranged in a k_1*k_2 2-D matrix, k_1 is represent the rows in the logical matrix, k_2 is represent the columns in the logical matrix. Second the horizontal redundant bits H are calculated from the decimal integer addition of selected symbols per row. In this each symbol is regarded as a decimal integer. Third, the vertical redundant bits V are calculated from the binary operation among the bits per column. Both the divide-symbol and arrange matrix are implemented in logical order instead of in physical order. Therefore, the DMC no need to changing the physical structure of the memory.

To explain the DMC scheme, for example we take a 32-bit word, as shown in Fig. 2. The information bits are denoted by cells from D_0 to D_{31}. This 32-bit word is divided into eight symbols of 4-bit. K1 = 4 have been chosen simultaneously. H0-H19 are horizontal check bits; the vertical check bits are V0-V7. Also, it meant that the maximum error correction capability and the number of redundant bits are different when the different values for k and m are chosen. Therefore, k and m should be carefully adjusted to maximize the error correction capability and minimize the number of redundant bits. For example, in this case, when k = 2 * 2 and m = 8, only 1-bit error can be corrected and the number of redundant bits is 36. However, when k = 2 * 4 and m = 4, the maximum error correction capability is up to 5 bits and the number of redundant bits is 36. In this paper, the error correction capability is consider as most important one to maintain the performance of the memory, so k = 4 * 2 and m = 4 are used to make DMC.

The horizontal redundant bits H can be calculate by decimal integer addition as follows:

H4H3H2H1H0 = D7D6D5D4 + D3D2D1D0 (1)

H9H8H7H6H5 = D15D14D13D12 + D11D10D9D8 (2)

And similarly for the horizontal redundant bits H14H13H12H11H10 and H19H18H17H16H15, where "+" represent Decimal integer addition. The vertical redundant bits V can be calculate as follows,

$V0 = D0 \oplus D8 \oplus D16 \oplus D \oplus 24$ (3)

$V1 = D1 \oplus D9 \oplus D17 \oplus D25$ (4)

And similarly for the rest of the vertical redundant bits

The encoding can be performed by decimal and binary addition operations from (1) to (4). The encoder calculate the redundant bits using multibit adders and XOR gates is shown in Fig. 3. In this figure, H19 – H0 are horizontal redundant bits, V7 – V0 are vertical redundant bits, and the remaining bits U31 – U0 are the information bits which are directly copied from D31 to D0.

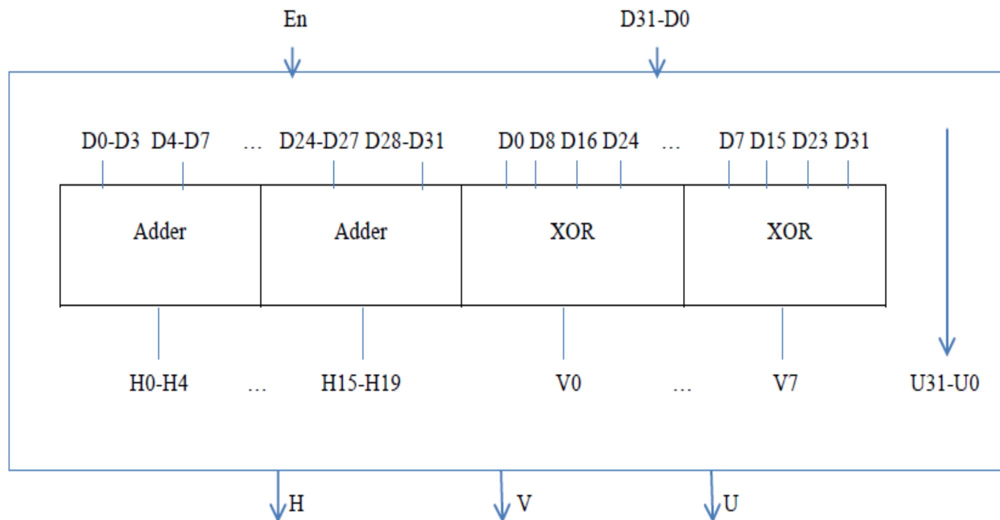

Figure.2. 32-bit DMC encoder structure using multibit adders and XOR gates

C. DMC Decoder:

The decoding process is need to know a correct word. For example, first, the received redundant bits H4H3H2H1H0' and V0' – V3' are generated by the received information bits D'. Second, the horizontal syndrome bits ΔH4H3H2H1H0 and the vertical syndrome bits S3 – S0 can be calculated as follows:

$$\Delta H_4H_3H_2H_1H_0 = H_4H_3H_2H_1H_0' - H_4H_3H_2H_1H_0 \qquad (5)$$

$$S_0 = V_0' \oplus V_0' \qquad (6)$$

Like for the rest of the vertical syndrome bits, where "-" represent decimal integer subtraction.

The stored codeword is original information bits in symbol 0 where no errors occur while ΔH4H3H2H1H0 and S3 – S0 are equal to zero, when ΔH4H3H2H1H0 and S3 – S0 are nonzero, the induced errors (the number of errors is 4 in this case) are detect and locate in symbol 0, and then these errors can be corrected by

$$D0correct = D0 \oplus S0. \qquad (7)$$

Table.1. 32 Bit Logical Organisation

D7	D6	D5	D4	D3	D2	D1	D0	H4	H3	H2	H1	H0
D15	D14	D13	D12	D11	D10	D9	D8	H9	H8	H7	H6	H5
D23	D22	D21	D20	D19	D18	D17	D16	H14	H13	H12	H11	H10
D31	D30	D29	D28	D27	D26	D25	D24	H19	H18	H17	H16	H15
V7	V6	V5	V4	V3	V2	V1	V0					

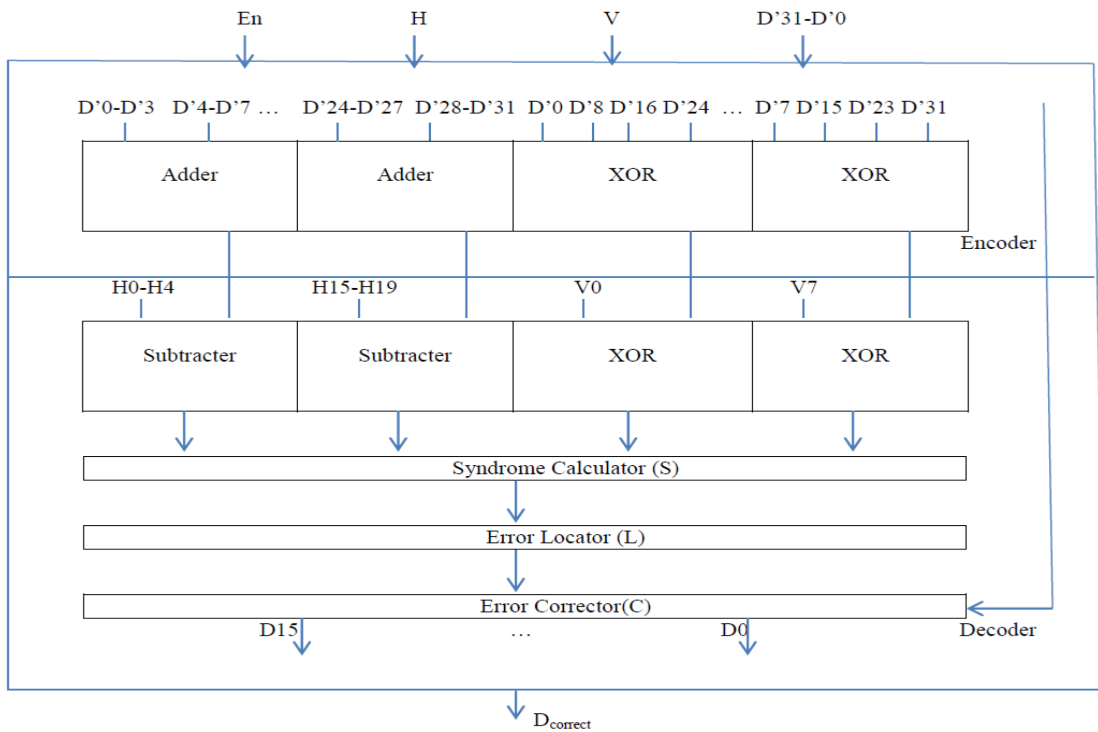

Figure.3. 32-bit DMC decoder structure using ERT

The Fig 3 shows the DMC decoder, which is made by some of the sub modules, and each execute a particular task in decoding section. syndrome calculator, error locator, and error corrector. It can be known from this figure that the redundant bits must be recomputed from the received information bits D' and compared to the original set of redundant bits in order to obtain the syndrome bits ΔH and S. Error locator is used to detect and locate the bits which is affect by error in this, by using ΔH and S. finally, the error corrector is used to correct these errors by inverting values of error bits.

In this, the reusing encoder is used to minimize the circuit area of DMC. This is calling the ERT. The ERT can decrease the area overhead of DMC without disturbing the whole encoding and decoding processes. From Fig, the DMC encoder is also again used for obtaining the syndrome bits in the DMC decoder section. Therefore, the entire circuit area of DMC can be decreased by using the existent circuits of encoder.

Besides, the figure shows the decoder with signal En for deciding whether the encoder needs to be a part of the decoder. The En signal is used to differentiate the encoder from the decoder, and it is manage the write and read signals in memory. Therefore, in the encoding (write) section, the DMC encoder is only an encoder to execute the encoding operations. However, in the decoding (read) section, this encoder is used for calculating the syndrome bits in the decoder section. These easily show how the area overhead of extra circuits can be substantially decreased.

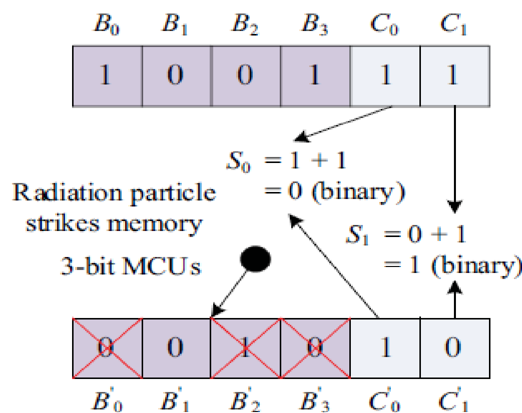

Figure.4. Limits of binary error detection in simple binary operations

III. SIMULATION RESULT ANALYSIS

A. Simulation Output:

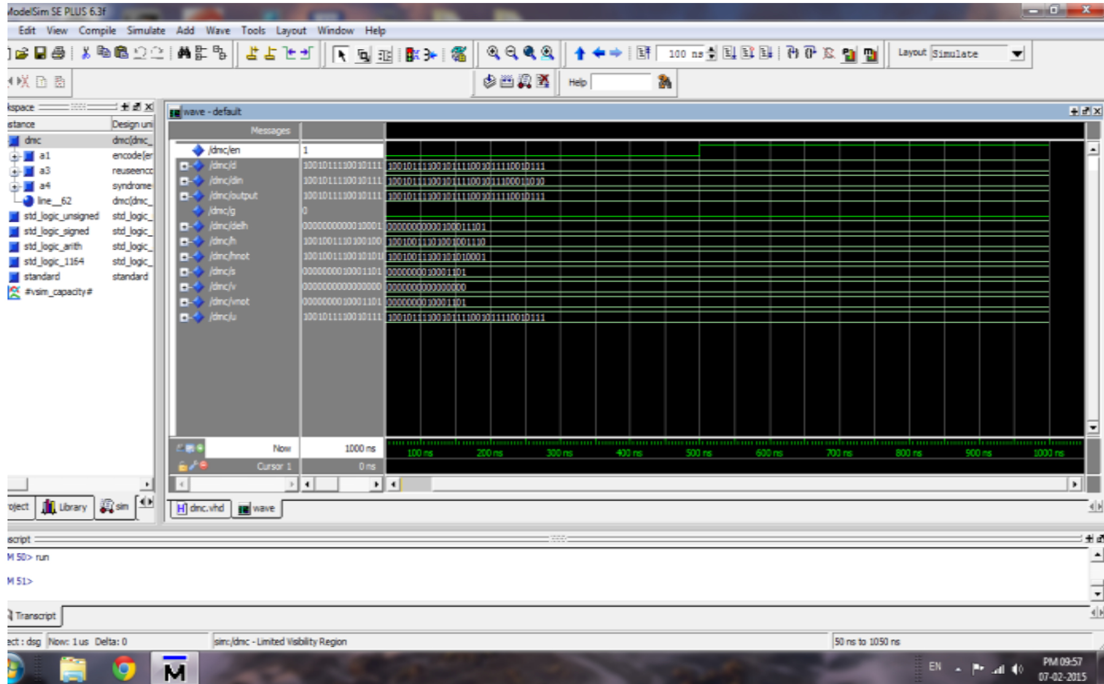

Figure.5. Simulation Output

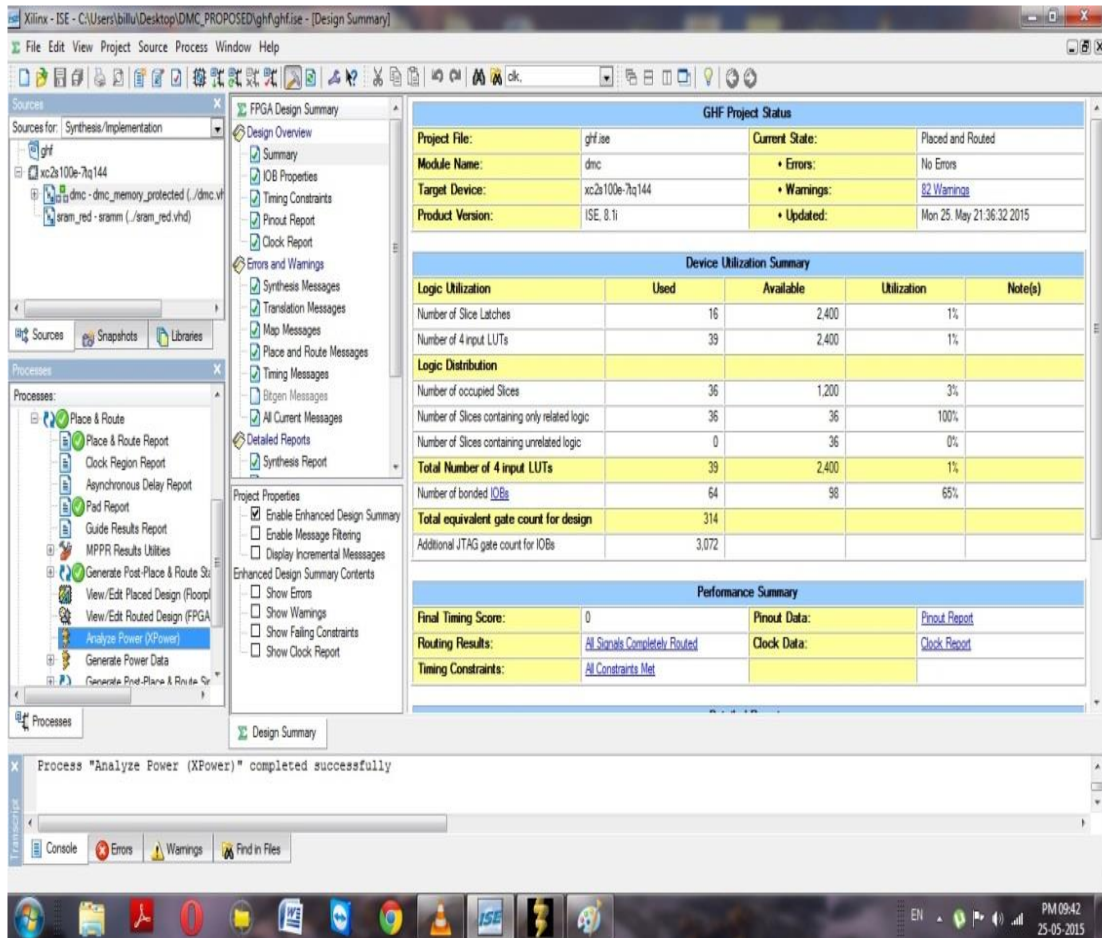

Figure.6. Area Analysis

Table.2. Comparison of Area Analysis

S N O	Logic Utilization	Used		Available		Utilization	
		E (Existing)	P (Proposed)	E (Existing)	P (Proposed)	E (Existing)	P (Proposed)
1	No of slice latches	32	16	2400	2400	1%	1%
2	No of 4 i/p LUTs	203	39	2400	2400	8%	1%
	Logic Distribution						
3	No of occupied slices	109	36	1200	1200	9%	3%
4	No of slices containing only related logic	109	36	109	36	100%	100%
5	No of slices containing only unrelated logic	0	0	109	36	0%	0%
	Total No of 4 i/p LUTs	203	39	2400	2400	8%	1%
6	No of bonded IOBs	96	64	98	98	97%	65%
	Total equivalent Gate count for design	1378	314				
7	Additional JTAG Gate count for IOBs	4608	3072				

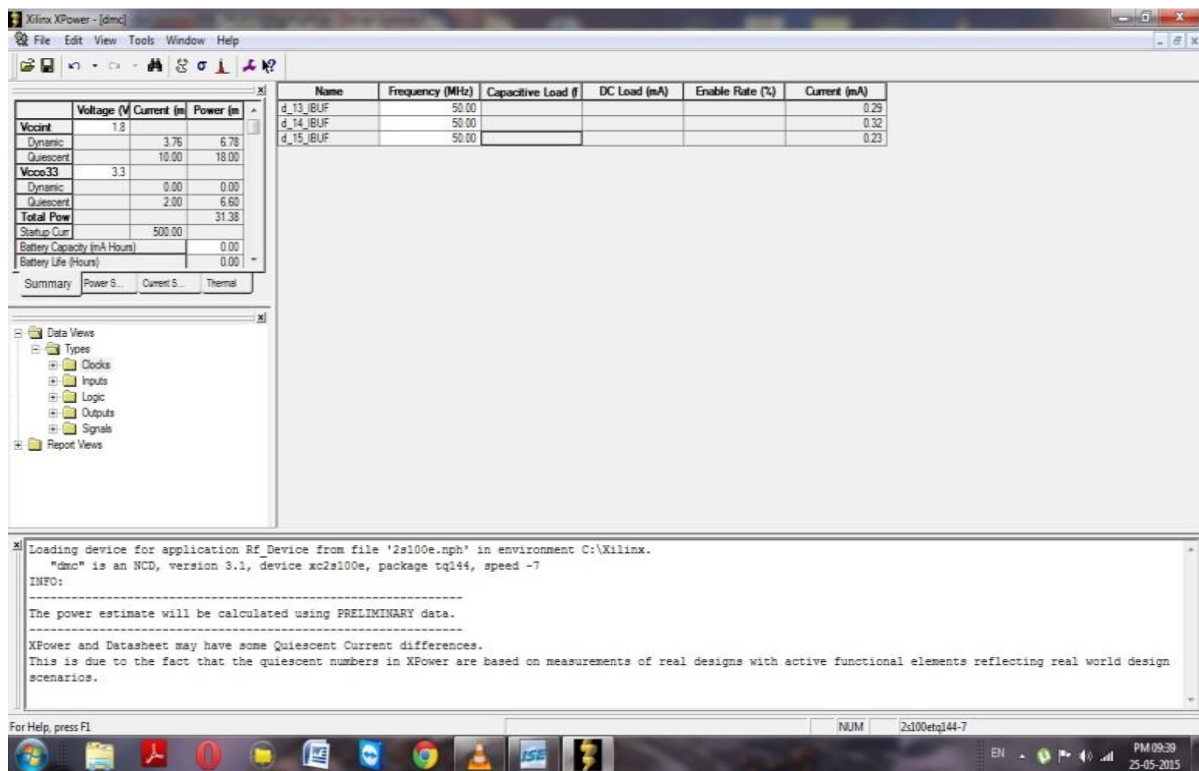

Figure.7. Power Analysis

Table.3. Comparison of Power Analysis

S N O		Voltage(V)		Current(mA)		Power(mW)	
		E (Existing)	P (Proposed)	E (Existing)	P (Proposed)	E (Existing)	P (Proposed)
1	V_{ccint}	1.8	1.8	--	--	--	--
2	Dynamic	--	--	5.33	3.76	9.60	6.78
3	Quiescent	--	--	10.00	10.00	18.00	18.00
4	$V_{cc0}33$	3.3	3.3	--	--	--	--
5	Dynamic	--	--	0.00	0.00	0.00	0.00
6	Quiescent	--	--	2.00	2.00	6.60	6.60
7	Total Power	--	--	--	--	34.20	31.38

IV. CONCLUSION

To prevent the memories from the MCUs, more complex error correction codes are used, but the major difficulty of that they would require high delay overhead. In the proposed method, we were implement the 32 bit Decimal Matrix code for detection and correction of errors in the memories and maintaining memory reliability. The proposed DMC increase the error detection and correction capability, decrease the area, power, maintain the memory reliability, and redundant bits will be reduced.

REFERENCES

[1] Jing Guo, Liyi Xiao, Zhigang Mao,Qiang Zhao, 2014 "Enhanced memory Reliability Against Multiple Cell Upset Using Decimal Matrix Code" IEEE Trans.VLSI,vol.22,no.1.

[2] D.Radaelli, H. Puchner, S. Wong, and S. Daniel, 2005, "Investigation of multi-bit upsets in a 150 nm technology SRAM device," IEEE Trans.Nucl. Sci., vol. 52, no. 6, pp. 2433– 2437.

[3] E.Ibe, H.Taniguchi, Y.Yahagi, K.Shimbo, and T.Toba, 2010, "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.

[4] C.Argyrides and D. K. Pradhan, 2007, "Improved decoding algorithm for high reliable reed muller coding," in Proc. IEEE Int. Syst. On Chip Conf., pp. 95–98.

[5] A.Sanchez-Macian, P. Reviriego, and J. A. Maestro, 2007, "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, 2012,"Efficient majority logic fault detection with difference-set codes for memory applications," IEEE Trans.Very Large Scale Integr. (VLSI) Syst., vol. 20, no. 1, pp. 148–156.

[7] M.Zhu,L.Y.Xiao,L.L.Song,Y.J.Zhang, and H. W. Luo, 2011, "New mix codes for multiple bit upsets mitigation in fault-secure memories,"Microelectron. J.,vol42,no.3, pp. 553–561,.

[8] R.Naseer and J.Draper, 2008, "Parallel double error correcting code design to mitigate multi-bit upsets in SRAMs," in Proc. 34th Eur. Solid-State Circuits,pp.222–225.

[9] G.Neuberger,D.L.Kastensmidt,and R. Reis, 2005, "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.

[10] P.Reviriego, M. Flanagan, and J. A. Maestro, 2012, "A(64,45) triple error correction code for memory applications," IEEE Trans. Device Mater. Rel.,vol.12,no.1,pp.101-106,.

[11] S.Baeg, S. Wen,and R.Wong, 2009, "Interleaving distance selection with a soft error failure model," IEEE Trans. Nucl. Sci., vol. 56, no. 4,pp. 2111–2118.

[12] K.Pagiamtzis and A. Sheikholeslami, 2003, "Content addressable memory(CAM)circuits and architectures: A tutorial and survey," IEEE J.Solid-State Circuits,vol.41,no.3,pp.712–727,.

[13] S.Baeg, S. Wen, and R. Wong, 2010, "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,.

[14] C.Argyrides, R.Chipana, F.Vargas, and D.K.Pradhan, 2011, "Reliability analysis of H- tree random access memories implemented with built in current sensor and parity codes for multiple bit upset correction," IEEE Trans.Rel.,vol.60, no.3, pp. 528–537,.

[15] C.Argyrides,D.K.Pradhan, and T. Kocak, 2011, "Matrix codes for reliable and cost efficient memory chips," IEEE Trans. Very Large Scale Integer.(VLSI) Syst., vol. 19, no. 3,pp. 420–428,.

[16] C.A.Argyrides, C. A. Lisboa, D. K. Pradhan, and L. Carro, 2009, "Single element correction in sorting algorithms with minimum delay overhead," in Proc. IEEE Latin Amer.Test Workshop,pp. 652–657.

[17] Y.Yahagi,H.Yamaguchi,E.Ibe,H.Kameyama,M.Sato,T.Akioka,andS.Yamamoto, 2007, "A novel feature of neutron-induced multi-cell upsets in 130 and 180nm SRAMs,"IEEETrans.Nucl.Sci.vol.54,no.4,pp.1030–1036,.

[18] N.N.Mahatme, B.L.Bhuva, Y.P.Fang, and A. S.Oates, 2012, "Impact of strained-Si PMOS transistors on SRAM soft error rates," IEEE Trans. Nucl.Sci., vol. 59, no. 4, pp.845–850,.