ISSN: 2278 – 7798
International Journal of Science, Engineering and Technology Research (IJSETR)
Volume 5, Issue 11, November 2016

DETECTION AND CORRECTION OF CELL UPSETS USING MODIFIED DECIMAL MATRIX

ENDREDDY PRAVEENA [1] M.T.ech Scholar (VLSID),

Universal College Of Engineering & Technology, Guntur, A.P

M. VENKATA SREERAJ [2] Associate Professor in E.C.E Department,

Universal college of Engineering and Technology Guntur, A.P

ABSTRACT: Scaling down of the CMOS technology is facing many issues. One of them is soft errors.These errors occur when memories are exposed to radiation environment. To protect memories from soft errors, advanced coding techniques like Error Correction Codes (ECC) are used. But these ECC codes require very complex encoder and decoder structures and also have higher delay overheads. Recently, a novel Decimal Matrix Code (DMC)[1] method based on decimal algorithm is used to protect memories from multiple cell upsets(MCU) [6]. But it can correct only 5 bit errors and also requires more redundant bits for error correction. In this paper, a modified-DMC is proposed to improve memory reliability. The proposed method uses DMC and Hamming codes for generating check bits. The proposed method can correct more bit errors with less number of redundant bits when compared to existing DMC. The proposed method also uses the encoder-reuse technique (ERT) to minimize the area overhead of extra circuits without modifying the encoding and decoding processes. The obtained results showed that the proposed modified-DMC has better protection level against large MCUs.

KEYWORDS: Error Correction Codes (ECC), Multiple Cell Upsets (MCU), Encoder Reuse Technique (ERT), Decimal Algorithm, Hamming Codes.

I.INTRODUCTION

Soft errors are the major issues in the reliability of memories. Soft errors damages only the data stored in memories but not physically. Computer memories are sensitive to soft errors which affects reliability [5]. Memory cells can be disturbed by high-energy neutron particles from terrestrial atmosphere or alpha particles resulted from IC package material. As CMOS technology scales down to nano scale, multiple cell upsets have become a vital reliability concern. To avoid these cell upsets in memories, advanced error correction codes have been widely used. Some of them are Hamming codes [2], BCH codes [3], and RS codes [4] . These codes uses sophisticated encoding & decoding architectures and require more area, power, and delay overheads. Recently, a new method called Decimal Matrix Code(DMC) is proposed to deal with MCUs. But DMC requires large number of redundant bits, which is more than double the information data and also its correction capability is limited. In this paper, a modified-DMC method based on Decimal Algorithm and Hamming codes is proposed. It divides the given data into symbols and arranges them in a matrix form. The proposed method employs Hamming Codes to calculate horizontal redundant bits, XOR operations to calculate vertical redundant bits and decimal algorithm to calculate syndrome bits. The proposed method uses encoder reuse technique which avoids the need for a separate circuit for decoding, which in turn reduces the area overhead of extra circuits. The proposed method uses Decimal Algorithm to detect and correct the errors. Decimal algorithm uses decimal integer addition & decimal integer subtraction. This paper is divided into 7 sections. Section II provides related work. The existing DMC is discussed in section III. The proposed modified-DMC and its encoder & decoder circuits are presented in section IV. The advantage of using the proposed modified-DMC is explained with an example in section V. The overhead analysis is discussed in section VI. At the end, some conclusions drawn from this paper are presented in section VII.

II.RELATED WORK

One of the major approaches for transient errors mitigation is Concurrent Error Detection (CED), also called duplication with comparison [7]. It can only detect errors but cannot correct them. It requires additional techniques for error correction. The CED needs an area overhead higher than 100% and hence it is not preferred. Hamming codes and odd weight codes are also used to mitigate Single Cell Upsets (SCU). These codes can correct single errors with reduced area and performance overhead [8]. The implementation of Hamming code involves two combinational blocks, one is responsible to code the data using parity bits generated from XOR

gates and the other for decoding the data using same logic plus a decoder that gives the address of erroneous bits and an inverter gate. However it fails if more than one error occurs. Interleaving technique [9] is used to deal with multiple errors in the physical arrangement of the memory cells, so that cells that belong to the same logical word are separated. The main drawback of using interleaving is that it affects floor planning, access time and power consumption as discussed in [10]. Recently Matrix codes [11] using redundant bits is used to deal with multiple cell upsets. It detects and corrects the cell errors using redundant bits. But its efficiency is low because a column containing multiple errors cannot be corrected using one redundant row. In order to overcome these drawbacks modified-DMC using Hamming code is proposed in this paper. The proposed method gives improved results when compared to other techniques. The area, power, and delay overhead analysis are presented in section VI.

III.EXISTING DMC

In this section, the existing DMC is discussed. This approach uses decimal algorithm, which enhances the error detection capability. Decimal algorithm uses decimal integer addition and decimal integer subtraction.

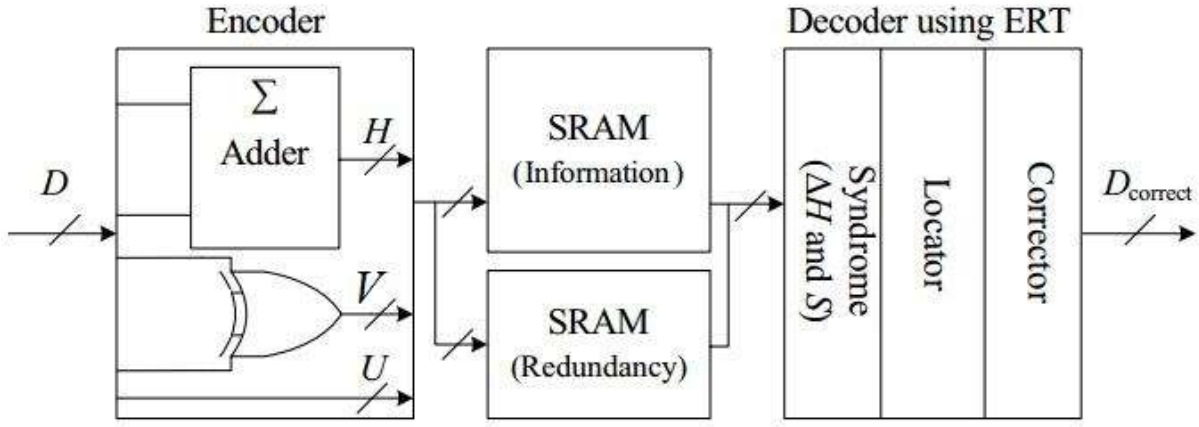

Fig. 1 Block diagram of existing DMC

Fig 1 shows the block diagram of existing DMC. In the encoding process, the information bits D are fed to the DMC encoder which uses decimal adder to compute the horizontal check bits H and exclusive-OR operation to calculate the vertical check bits V. Then the encoded data along with a copy of information data are stored in SRAM cells. When these cells are hit by radiation particles, MCUs may be induced. The decoder corrects these MCUs with the help of syndrome bits. The decoder uses horizontal syndrome bits to detect errors and uses vertical syndrome bits to correct those errors.

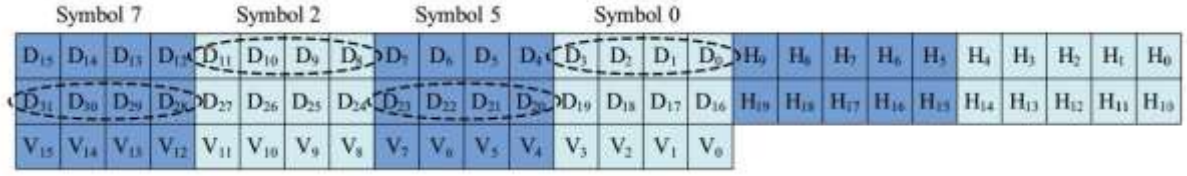

Fig. 2 Logical representation of 32-bit DMC code. Here, each symbol is regarded as a decimal integer

To understand the existing DMC scheme, logical representation of a 32-bit DMC code is shown in fig 2. The cells numbered from D_0 to D_{31} are information bits. This 32-bit word (N = 32) is divided into eight symbols (k = 8) of 4 bits (m = 4) each. The cells numbered from H_0 to H_{19} are horizontal check bits and from V_0 to V_{15} are vertical check bits. The maximum correction capability of DMC codes and the number of redundant bits are different when different values for k and m are chosen. For example, if k = 2x2 and m = 8, it can correct only one error and the number of redundant bits are 40. If k = 4x4 and m = 2, it can correct three errors and the number of redundant bits are reduced to 32. However, if k = 2x4 and m = 4, it can correct five errors and the number of redundant bits are 36. Therefore, the values of k and m should be selected carefully.

The following equations represent the decimal integer addition to find the horizontal check bits.

H[4:0] = symbol2 + symbol0

H[9:5] = symbol7 + symbol5

H[14:10] = symbol3 + symbol1

H[19:15] = symbol6 + symbol4 ….and so on

The following equations are used to compute the vertical check bits.

$V_0 = D_0 \wedge D_{16}$

$V_1 = D_1 \wedge D_{17}$ …..and so on

The symbol " ^ " indicates binary exclusive OR operation.

The horizontal syndrome bits are obtained as follows:

$$\blacktriangle H[4:0] = H[4:0]' - H[4:0]$$

$$\blacktriangle H[9:5] = H[9:5]' - H[9:5]$$ …..and so on

The vertical syndrome bits are obtained as follows:

$$S_0 = V_0' \wedge V_0$$

$$S_1 = V_1' \wedge V_1$$ ……and so on

The errors can be corrected by $D_{0correct} = D_0' \wedge S_0$

The drawback of the existing DMC is that it requires large number of redundant bits which is more than double the information bits. The correction capability of this method is also low. It cannot detect type 3, type 4 and type 5 errors discussed later in section V.

IV. PROPOSED MODIFIED-DMC

In this section, modified-DMC is proposed to enhance memory reliability. This approach uses Hamming codes and decimal algorithm, which enhances the error detection and correction capability.

A. Block diagram of proposed modified-DMC:

The proposed block diagram of modified- DMC is shown in fig 3. Here, the encoding process may be interpreted as writing the data into SRAM cells and the decoding process as reading the data from SRAM cells. The information bits D are fed to the DMC encoder which computes the horizontal check bits H and vertical check bits V.

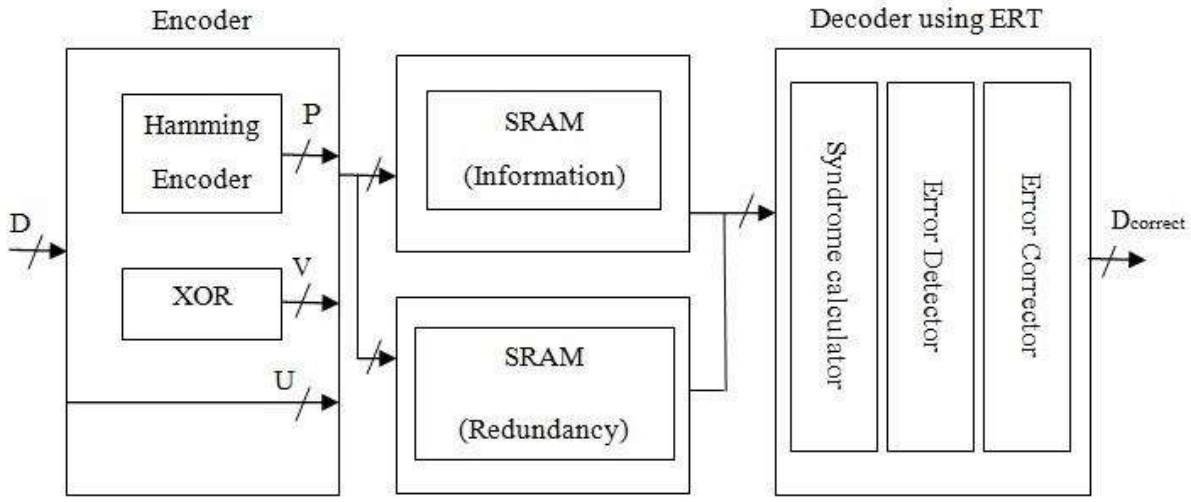

Fig. 3 Block diagram of modified-DMC

Once the encoding process is completed, the obtained modified-DMC code is stored in SRAM cells. Along with these bits, a copy of information bits D is stored in the register U. Now, when these cells are exposed to radiations, MCUs may occur. These MCUs are corrected in the decoding process. The proposed protection code utilized decimal algorithm to detect errors. Since the decoder uses ERT technique, the area overhead of extra circuits is also reduced significantly.

B.Encoder of modified-DMC:

The proposed modified-DMC encoder is similar to existing DMC encoder but employs Hamming codes for calculating horizontal redundant bits. The circuit of encoder using Hamming encoder and XOR gates is shown in fig 4. First, an Nbit information word is taken as the source data. This N-bit word is divided into k-symbols of m-bits each which gives N = kxm. These k-symbols are arranged systematically in a k_1xk_2 2-D matrix form such that k = k_1xk_2. Here k_1 indicates number of rows and k_2 indicates number of columns. Second, the horizontal check bits are computed by employing Hamming codes for each symbol of first row. Here, each symbol is considered as a decimal integer. Third, the vertical check bits are computed by employing binary exclusive OR operation for every column.

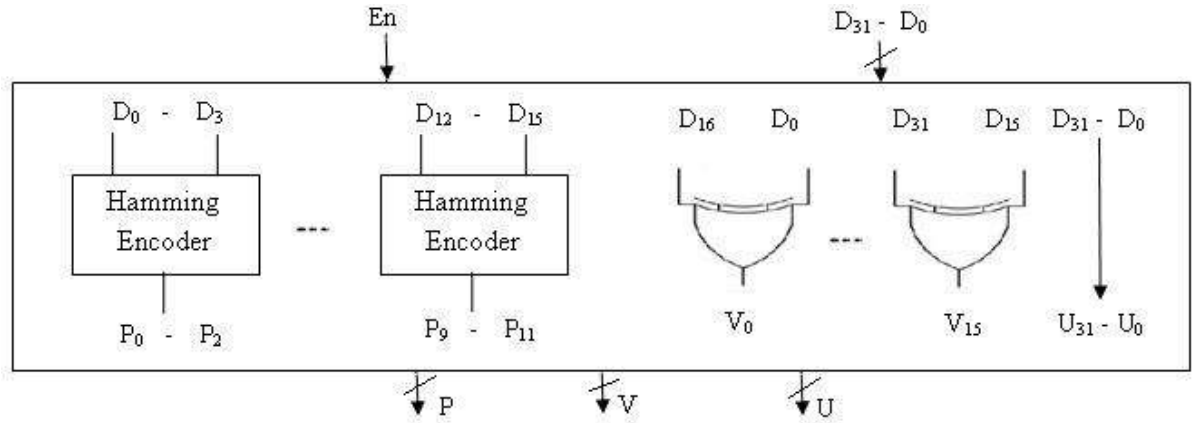

Fig. 4 Modified-DMC encoder

To understand the proposed DMC scheme, implementation of a 32-bit DMC code is explained as an example in fig 5. The cells numbered from D_0 to D_{31} are information bits. This 32-bit word (N = 32) is divided into eight symbols (k = 8) of 4 bits (m = 4) each. The cells numbered from P_0 to P_{11} are horizontal check bits and from V_0 to V_{15} are vertical check bits. The maximum correction capability and the number of redundant bits are different when different values for k and m are chosen. For example, if k = 2x4 and m = 4, it can correct maximum of 16-bit errors and the number of redundant bits are 28. Therefore, the values of k and m should be selected carefully.

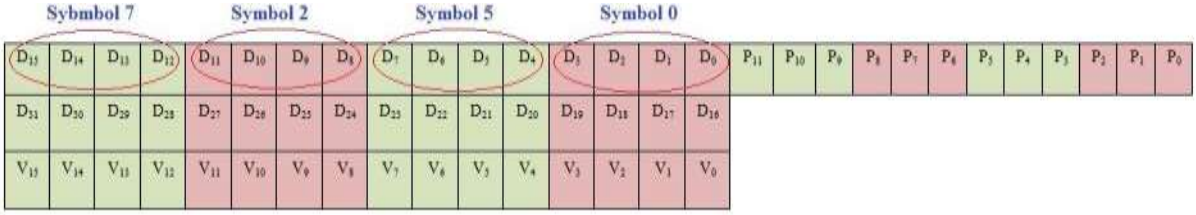

Fig. 5 Logical representation of 32-bit modified-DMC

The horizontal redundant bits are calculated for all the symbols in only one row. The following equations represent the Hamming equations to calculate the horizontal check bits.

$$P_0 = D_3 \wedge D_1 \wedge D_0$$

$$P_1 = D_3 \wedge D_2 \wedge D_0$$

$$P_2 = D_3 \wedge D_2 \wedge D_1 \ldots\ldots \text{and so on}$$

The following equations are used to compute the vertical check bits.

$$V_0 = D_0 \wedge D_{16}$$

$$V_1 = D_1 \wedge D_{17} \ldots\ldots\ldots \text{ and so on}$$

The symbol " ^ " indicates binary exclusive-OR operation.

C.Decoder of modified-DMC

In the decoding process, we need to detect the errors if any and correct them accordingly. Here foremost step is using encoder re-use technique. The received information bits D' are applied to the in-built encoder block in the decoder circuit to obtain the horizontal check bits P_0' to P_{11}' and vertical check bits V_0' to V_{15}'.The schematic of the decoding circuit is shown in fig 6.

The decoding process goes through a step by step process i.e., syndrome calculator, error-locator and error corrector.The decimal integer subtraction is used to compute horizontal syndrome bits and exclusive OR operation to compute vertical syndrome bits. The non-zero horizontal syndrome bits indicates error detection and the non-zero vertical syndrome bits gives the location of errors. These errors are corrected by error corrector block using xor operations.

The horizontal syndrome bits are obtained as follows:

$$\blacktriangle\ P_0P_1P_2D_3D_2D_1D_0 = P_0P_1P_2D_3D_2D_1D_0' - P_0P_1P_2D_3D_2D_1D_0$$

$$\blacktriangle\ P_3P_4P_5D_7D_6D_5D_4 = P_3P_4P_5D_7D_6D_5D_4' - P_3P_4P_5D_7D_6D_5D_4 \ldots\text{.and so on}$$

The vertical syndrome bits are obtained as follows:

$$S_0 = V_0' \wedge V_0$$

$$S_1 = V_1' \wedge V_1 \ldots\ldots\text{.and so on}$$

The errors can be corrected by $D_{0correct} = D_0' \wedge S_0 \ldots$ and so on

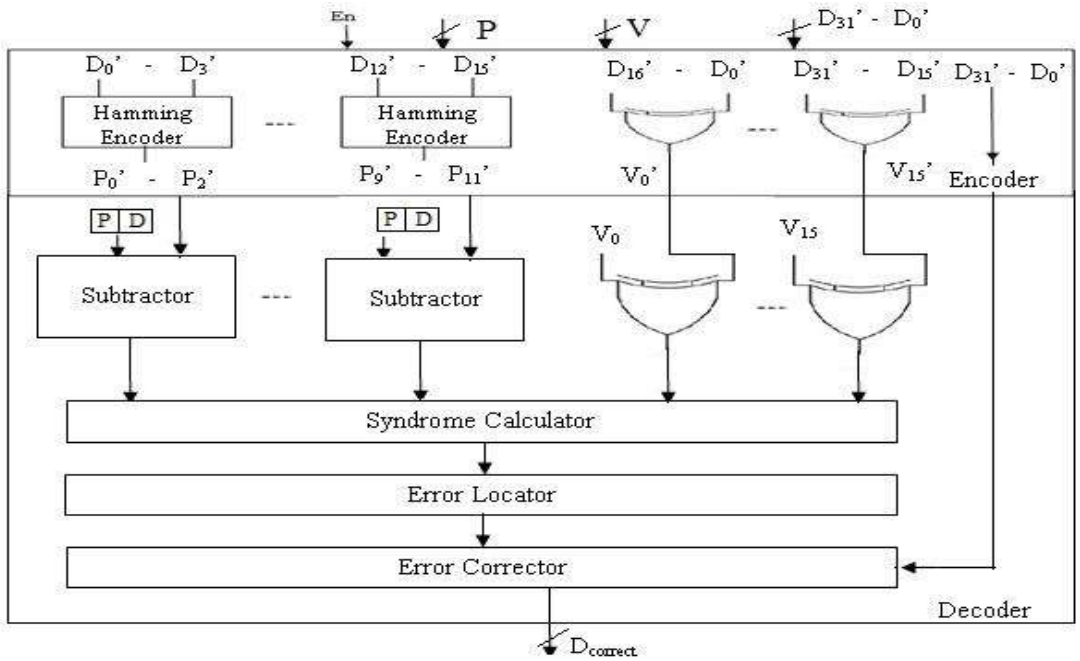

Fig. 6 Modified-DMC decoder

The table I shows the functions of En signal. The enable signal En can be used to determine whether the encoder needs to be a part of the decoder. Hence, the En signal distinguishes the encoder from the decoder and it will be under the control of the write and read signals in memory.

Table I
Functions of Enable signal

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

V.ADVANTAGE OF MODIFIED-DMC

The detection procedure of decimal error detection using the proposed modified-DMC structure is as shown in fig 7. First error detection is carried out and if errors are found then error location is calculated and accordingly correction of those errors will be done. And fig 8 shows the different types of errors.

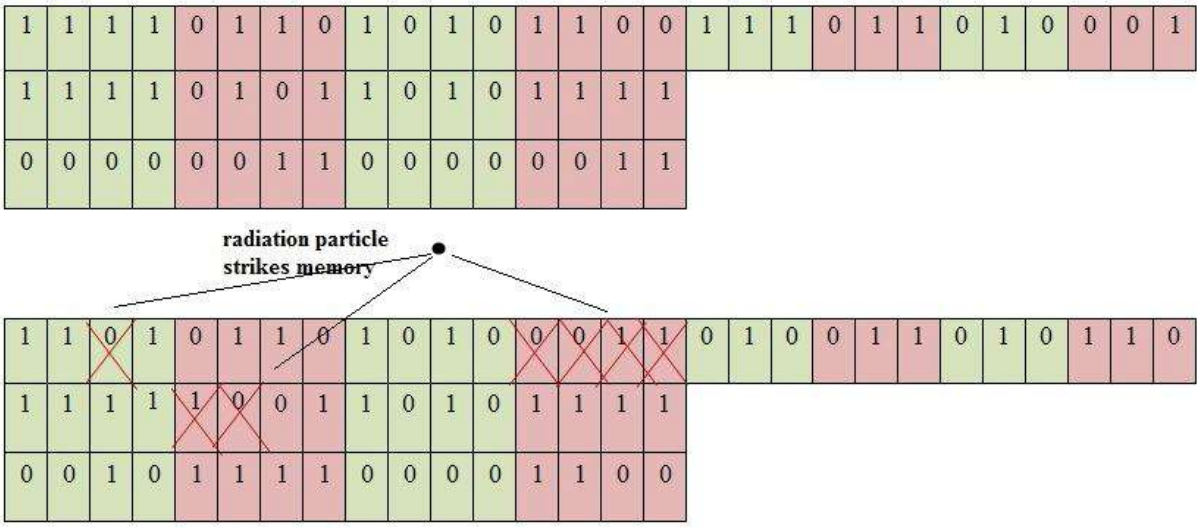

Fig. 7 Detection process with an example

Error Detection:

1- If $\blacktriangle P_0P_1P_2D_3D_2D_1D_0 \neq 0$ then error occurred in symbol 0 and no error in symbol 1.

2- If $\blacktriangle P_0P_1P_2D_3D_2D_1D_0 = 0$ and $S_3S_2S_1S_0 \neq 0$ then error occurred in symbol 1 and no error in symbol 0.

3- If $\blacktriangle P_0P_1P_2D_3D_2D_1D_0 = 0$ and $S_3S_2S_1S_0 = 0$ then no errors in symbol 0 and symbol 1.

Similarly the rest of the errors can be detected.

Here,

$\blacktriangle P_0P_1P_2D_3D_2D_1D_0 = P_0P_1P_2D_3D_2D_1D_0' - P_0P_1P_2D_3D_2D_1D_0$

$= 0110011 - 1001100$

$= 1100111 \neq 0$ => error occurred in symbol 0.

Error Location:

$S_0 = V_0' \wedge V_0 = 0 \wedge 1 = 1$

$S_1 = V_1' \wedge V_1 = 0 \wedge 1 = 1$

$S_2 = V_2' \wedge V_2 = 1 \wedge 0 = 1$

$S_3 = V_3' \wedge V_3 = 1 \wedge 0 = 1$ …..and so on

Error Correction:

$D_{0\ correct} = D_0' \wedge S_0 = 1 \wedge 1 = 0$

$D_{1\ correct} = D_1' \wedge S_1 = 1 \wedge 1 = 0$

$D_{2\ correct} = D_2' \wedge S_2 = 0 \wedge 1 = 1$

$D_{3\ correct} = D_3' \wedge S_3 = 0 \wedge 1 = 1$……and so on

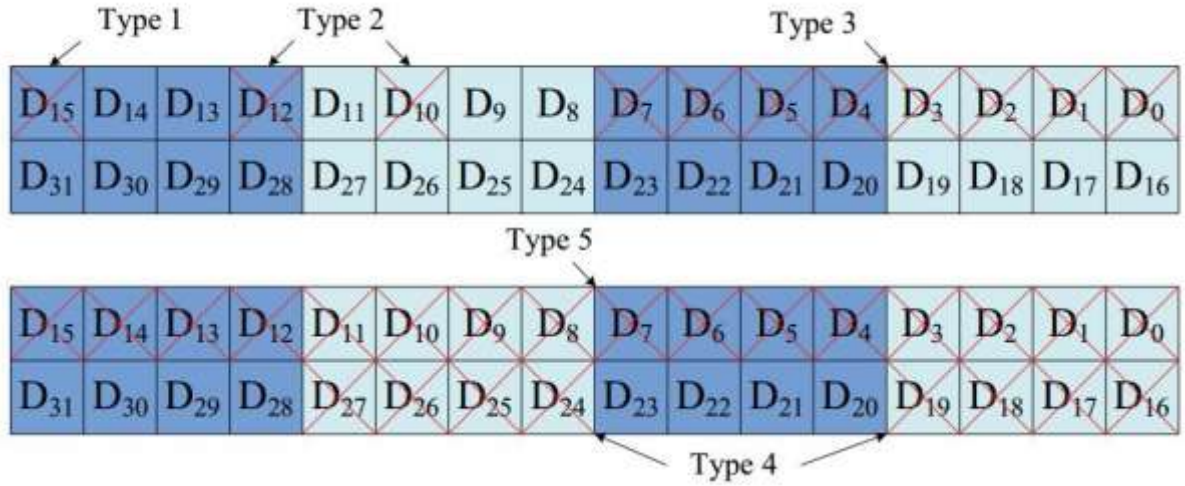

Fig. 8 Type 1 is a single error, type 2 is an inconsecutive error in two consecutive symbols, type 3 is a consecutive error

in two consecutive symbols, type 4 is an inconsecutive error in two inconsecutive symbols, and type 5 is a consecutive error in four consecutive symbols. For a 32-bit data, existing DMC requires 36 redundant bits to deal with MCUs and can correct up to five errors. Whereas for the same 32-bit data the proposed modified DMC need just 28 redundant bits to correct MCUs. Also the proposed technique can correct a maximum of 16 errors, provided no two errors occur in the same column. The existing DMC cannot correct type 4 and type 5 errors. It also cannot detect type 3 errors when following conditions occur:

(1) The decimal integer addition of bits in symbols 0 and 2 is equal to 2_m-1

(2) All the bits in symbols 0 and 2 are upset

The proposed modified-DMC successfully detects and corrects type 3 and type 4 errors provided no two errors occur in the same column.

VI. RESULT AND DISCUSSION

The complete proposed method is coded in VHDL. The design was simulated using ModelSim and Xilinx ISE is used to estimate the area required for the design, power consumption and delay overhead. From fig 7, it can be observed that the encoder is re-used for obtaining the syndrome bits in the decoder. Therefore, the overall circuit area for the implementation of the proposed method is reduced.

Table II
Area, Power, Delay analysis for 32-bit data

TECHNIQUE	Area (Gate count)	Power (mW)	Delay (ns)
Modified DMC	65970	84	8.876
DMC	74857	103	11.29

The area, power and delay overhead analysis of encoder and decoder circuits are given in table II. It shows the area in terms of number of gates required to implement the design, the total estimated power consumption in mW and the overall delay to get the output when the input is applied to the design.

Table III
Reduction in redundant bits

TECHNIQUE	Information bits	Redundant bits
Modified DMC	32	28
DMC	32	36

The differences in the number of redundant bits required are given in table III. It shows that for 32-bit information the proposed method requires 28 redundant bits while the existing method requires 36 bits. From this table, it can be observed that the proposed modified-DMC technique reduced the number of redundant bits required to detect and correct the MCUs.

VII.CONCLUSION

In this paper, modified-DMC has been proposed to protect memories from radiation induced errors. The proposed method uses a combination of Hamming coding and decimal algorithm which allows detection and correction of large MCUs. Also, the use of Encoder re-use technique reduced the area overhead of extra circuits. The obtained results show that the proposed method provides more reliable data than the existing methods. The proposed method still requires reasonably good amount of redundant bits to provide reliable data. Hence, this paper may be researched in order to reduce the number of redundant bits keeping a higher error correction capability.

REFERENCES

1. Jing Guo, Liyi Xiao, “Enhanced memory reliability against multiple cell upsets using decimal matrix code,” IEEE Transactions on VERY
LARGE SCALE INTEGRATION (VLSI) SYSTEMS, vol. 22, No.1, January 2014.
2. 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.
3. R. Naseer and J. Draper, “Parallel double error correcting code design to mitigate multi-bit- upsets in SRAMs,” in Proc. 34th Eur. Solid-State Circuits, Sep. 2008,, pp. 222-225.
4. 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.
5. R. C. Baumann, “Radiation-induced soft errors in advanced semiconductor technologies,” IEEE Trans. Device Mater. Reliab., vol. 5, no. 3, pp. 301-316, 2005.
6. 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.
7. Wakerly, J. F. Error detecting codes, self-checking circuits and applications. New York: North-Holland, 1978.
8. A. D. Houghton, The Engineer’s Error Coding Handbook. London, U.K.: Chapman and Hall, 1997.
9. S. Baeg, S. Wen, and R. Wong, “Interleaving distance selection with a soft error failure model,” IEEE Trans. Nucl. Sci., vol. 814-822, Apr. 2010.
10. 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.
11. 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.

ENDREDDY PRAVEENA [1] **M.T.ech Scholar (VLSID**), Universal College Of Engineering & Technology, Guntur, A.P

M. VENKATA SREERAJ [2] Associate Professor in E.C.E Department**,** Universal college of Engineering and Technology Guntur, A.P