Minimizing MCU in SRAM Cell to Enhance Memory Reliability Using DMC

T.Maheswari[#1]
PG Scholar,
Nandha Engineering College, Erode.
Anna University, Tamilnadu, India.
E-mail: tmaheswariece@gmail.com
P.Sukumar[#2]
ASP of ECE,
Nandha Engineering College, Erode.
Anna University, Tamilnadu, India.
E-mail: sukumarwin@gmail.com
R.K.Gnanamurthy[#3]
Principal,
SKP Engineering College,
Thiruvannamalai. Tamilnadu, India.
E-mail: rkgnanam@yahoo.com

Abstract— **Computer memories are sensitive to soft errors which can affect system reliability [1]. Memory cells can be disturbed by high-energy neutron particles from terrestrial atmosphere or alpha particles resulted from IC package material.**

Keywords— **Error correction codes, multiple cell upsets, decimal matrix code, encoder reuse technique.**

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. Especially, SRAM memory failure rates are increasing significantly, therefore posing a major reliability concern for many applications. Although single bit upset is a major concern about memory reliability, multiple cell upsets (MCUs) have become a serious reliability concern in some memory applications.

A number of techniques can be used to protect circuits against soft errors. For example, in [2] the use of a new type of flip-flop is proposed to correct errors, and in [3] such protection is addressed at the transistor level. In memories, to prevent soft errors from causing data corruption, Single Error Correction (SEC) codes are typically used to protect memories [4]. However, as technology scales, it is more likely that a single particle hit changes the value of more than one memory cell [5]. This phenomenon is known as Multiple Cell Upset [6]. The cells affected by an MCU are physically close, as errors are caused by the same particle hit [7]. To prevent an MCU from causing more than one error in a coded word, interleaving is normally used [8] and punctured difference set (PDS) codes [9] have been used to deal with MCUs in memories.

More recently, in [10], 2-D matrix codes (MCs) are 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 bits per row are protected by Hamming code, while parity code is added in each column. For the MC [10] based on Hamming, when two errors are detected by Hamming, the vertical syndrome bits are activated so that these two errors can be corrected. As a result, MC is capable of correcting only two errors in all cases. In [11], an approach that combines decimal algorithm with Hamming code has been conceived to be applied at software level. It uses addition of integer values to detect and correct soft errors. The results obtained have shown that this approach have a lower delay overhead over other codes.

In this paper, novel decimal matrix code (DMC) based on divide-symbol is proposed to provide enhanced memory reliability. The proposed DMC utilizes decimal algorithm (decimal integer addition and decimal integer subtraction) to detect errors. The advantage of using decimal algorithm is that the error detection capability is maximized so that the reliability of memory is enhanced. Besides, the encoder-reuse technique (ERT) is proposed to minimize the area overhead of extra circuits (encoder and decoder) without disturbing the whole encoding and decoding processes, because ERT uses DMC encoder itself to be part of the decoder.

This paper is divided into the following sections. The proposed DMC is introduced and its encoder and decoder circuits are present in Section II. The reliability and overheads analysis of the proposed code are analyzed in Section III.

II. DMC TECHNIQUE

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

A. Block diagram of DMC

The proposed block diagram of decimal matrix code is shown in Fig. 1 Here, during the encoding (write) process, information bits *D* are fed to the DMC encoder, and then the horizontal redundant bits *H* and vertical redundant bits *V* are obtained from the DMC encoder. When the encoding process is completed, the obtained DMC codeword is stored in the memory. If MCUs occur in the memory, these errors can be corrected in the decoding (read) process. The proposed protection code utilized decimal algorithm to detect errors, so that more errors were detected and corrected. Due to the advantage of decimal algorithm, the proposed DMC has higher fault-tolerant capability with lower performance overheads. In the fault-tolerant memory, the ERT technique is proposed to reduce the area overhead of extra circuits.

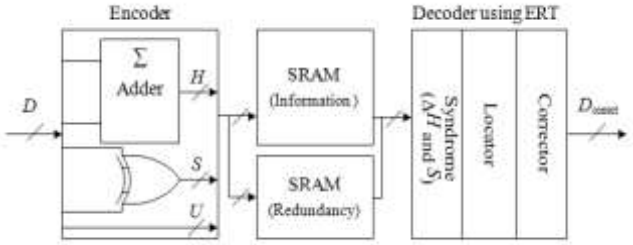


Fig.1Block diagram Decimal Matrix Code

B. Encoder of DMC

First, the divide-symbol and arrange-matrix ideas are performed, i.e., the *N*-bit word is divided into k symbols of m bits (N=k x m)) and these symbols are arranged in a k_1 x k_2 2-D matrix (k=k_1 x k_2) where the values of k1 and k2 represent the numbers of rows and columns in the logical matrix respectively). Second, the horizontal redundant bits H are produced by performing decimal integer addition of selected symbols per row. Here, each symbol is regarded as a decimal integer. Third, the vertical redundant bits V are obtained by binary operation among the bits per column. It should be noted that both divide-symbol and arrange-matrix are implemented in logical instead of in physical. Therefore, the proposed DMC does not require changing the physical structure of the memory.

We take a 32-bit word as an example, as shown in Fig. 2. The cells from D_0 to D_{31} are information bits. This 32-bit word has been divided into eight symbols of 4-bit. K_1 and k_2 have been chosen simultaneously. H_0–H19 are horizontal check bits; V_0 through V_{15} are vertical check bits. However, it should be mentioned that the maximum correction capability be corrected) and the number of redundant bits are different when the different values for k and m are chosen.

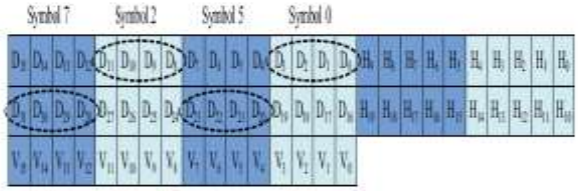


Fig. 2 32-bits DMC logical organization

Therefore, k and m should be carefully adjusted to maximize the correction capability and minimize the number of redundant bits. For example in this case, when k= 2 x 2 and m = 8, only 1-bit error can be corrected and the number of redundant bits is 40. When k=4 x 4 and, 3-bit m = 2 errors can be corrected and the number of redundant bits is reduced to 32. However, when k=4 x 4 and m= 4, the maximum correction capability is up to 5 bits and the number of redundant bits is 36. In this paper, in order to enhance the reliability of memory, the error correction capability is first considered, so k = 2 × 4 and m = 4 are utilized to construct DMC.

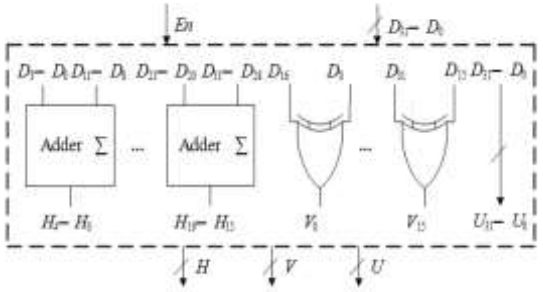


Fig. 3 Encoder of Decimal Matrix Code

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

$H_4H_3H_2H_1H_0 = D_3D_2D_1D_0 + D_{11}D_{10}D_9D_8$ (1)

$H_9H_8H_7H_6H_5 = D_7D_6D_5D_4 + D_{15}D_{14}D_{13}D_{12}$ (2)

and similarly for the horizontal redundant bits $H_{14}H_{13}H_{12}H_{11}H_{10}$ and $H_{19}H_{18}H_{17}H_{16}H_{15}$, where "+" represents decimal integer addition.

For the vertical redundant bits V, we have and similarly for the rest vertical redundant bits.

$V_0 = D_0 \oplus D_{16}$ (3)

$V_1 = D_1 \oplus D_{17}$ (4)

The encoding can be performed by decimal and binary addition operations from (1) to (4). The encoder that computes the redundant bits using multibit adders and XOR gates is shown in Fig 3. In this figure, $H_{19} – H_0$ are horizontal redundant bits, $V_{15} – V_0$ are vertical redundant bits, and the remaining bits $U_{31} - U_0$ are the information bits which are directly copied from D_{31} to D_0.

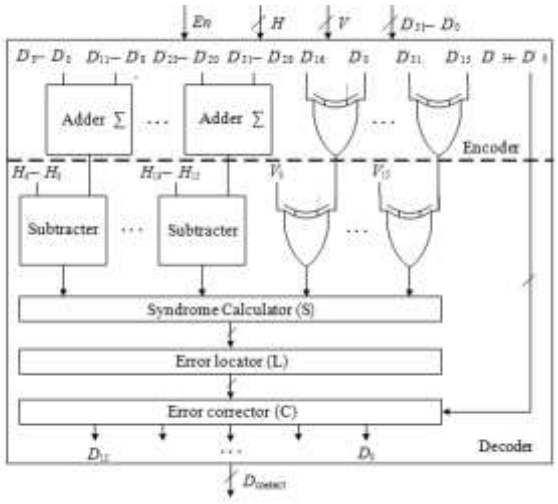
C. Decoder of DMC

To obtain a word being corrected, the decoding process is required. For example, first, the received redundant bits $H_4H_3H_2H_1H_0$' a and V_0 '- V_3' are generated by the received information bits D'. Second, the horizontal syndrome bits $\Delta H_4H_3H_2H_1H_0$ and the vertical syndrome bits $S_3 - S_0$ can be calculated as follows:

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

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

and similarly for the rest vertical syndrome bits, where "–"represents decimal integer subtraction.



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

Fig. 4 Decoder of Decimal Matrix Code using ERT

When Δ $H_4H_3H_2H_1H_0$ and $S_3 - S_0$ are equal to zero, the stored codeword has original information bits in symbol 0 where no errors occur. When Δ $H_4H_3H_2H_1H_0$ and $S_3 - S_0$ are nonzero, the induced errors (the number of errors is 4 in this case) are detected and located in symbol 0, and then these errors can be corrected by

$$D_0\text{correct} = D_0 \oplus S_0. \quad (7)$$

The proposed DMC decoder is depicted in Fig. 4 , which is made up of the following submodules, and each executes a specific task in the decoding process: syndrome calculator, error locator, and error corrector. It can be observed 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. S denotes the syndrome bits. Then error locator uses ΔH and S to detect and locate which bits some errors occur in. Finally, in the error corrector, these errors can be corrected by inverting the values of error bits.

In the proposed scheme, the circuit area of DMC is minimized by reusing its encoder. This is called the ERT. The ERT can reduce the area overhead of DMC without disturbing the whole encoding and decoding processes. From Fig. 4, it can be observed that the DMC encoder is also reused for obtaining the syndrome bits in DMC decoder. Therefore, the whole circuit area of DMC can be minimized as a result of using the existent circuits of encoder. Besides, this figure also shows the proposed decoder with an enable signal *En* for deciding whether the encoder needs to be a part of the decoder. In other words, the *En* signal is used for distinguishing the encoder from the decoder, and it is under the control of the write and read signals in memory. Therefore, in the encoding (write) process, the DMC encoder is only an encoder to execute the encoding operations. However, in the decoding (read) process, this encoder is employed for computing the syndrome bits in the decoder. These clearly show how the area overhead of extra circuits can be substantially reduced.

D. Advantage of decimal error detection

The detection procedure of decimal error detection using the proposed structure shown in Figure 2 is fully described in Fig. 5. First of all, the horizontal redundant bits $H_4H_3H_2H_1H_0$ are obtained from the original information bits in symbols 0 and 2 according to (1)

$$H_4H_3H_2H_1H_0 = D_3D_2D_1D_0 + D_{11}D_{10}D_9D_8$$
$$= 1100 + 0110$$
$$= 10010. \quad (8)$$

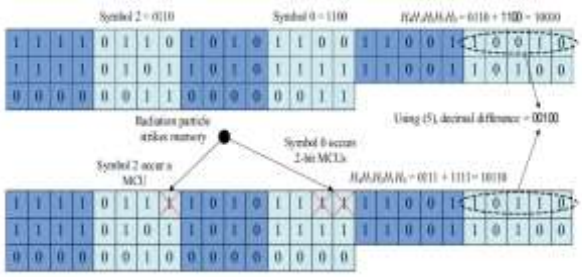


Fig. 5. Advantage of decimal error detection

When MCUs occur in symbol 0 and symbol 2, i.e., the bits in symbol 0 are upset to "1111" from "1100"($D_3D_2D_1D_0$' = 1111) and the bits in symbol 2 are upset to "0111" from "0110" ($D_{11}D_{10}D_9D_8$'= 0111). During the decoding process, the received horizontal redundant bits$H_4H_3H_2H_1H_0$ are first computed, as follows:

$$H_4H_3H_2H_1H_0' = D_{11}D_{10}D_9D_8' + D_3D_2D_1D_0'$$
$$= 0111 + 1111$$
$$= 10110. \quad (9)$$

Then, the horizontal syndrome bits $\Delta H_4H_3H_2H_1H_0$

can be obtained using decimal integer subtraction

$$\Delta H_4H_3H_2H_1H_0 = H_4H_3H_2H_1H_0' - H_4H_3H_2H_1H_0$$
$$= 10110 - 10010$$
$$= 00100. \quad (10)$$

The decimal value of $\Delta H_4H_3H_2H_1H_0$ is not "0," which represents that errors are detected and located in symbol 0 or symbol 2. Subsequently, the precise location of the bits that were flipped can be located by using the vertical syndrome bits S3 − S0 and S11 − S8. Finally, all these errors can be corrected by (7). Therefore, based on decimal algorithm, the proposed technique has higher tolerance capability for protecting memory against MCUs.

III.RESULTS AND DISCUSSIONS

In this section, the proposed DMC has been implemented in HDL, simulated with ModelSim and tested for functionality. For fair comparisions, Hamming, and Reed-solomon codes are used for references. The area, power, and critical path delay of extra circiuts have been obtained using Xilinx software.

A. Overhead analysis

The results of coverage are shown in Table I. It can be seen that our proposed DMC have superior protection level compared with other codes. These results show how our proposed technique provides single- and double-error correction, but can also provide effective tolerance capabilities against large MCUs that exceed the performance of other codes.

TABLE 1.
AREA, POWER AND DELAY ANALYSIS OF DMC, HAMMING CODE AND RS CODE

S.No	ECC Codes	Area(Gate counts)	Power(mw)	Delay(ns)
1	DMC	51958	84	4.593
2	Hamming Code	134872	138	3.995
3	RS Code	111384	34	13.603

Fig. 6 indicates that the memory with the proposed scheme performs faster than other codes. Different decoding algorithms could result in different overheads. RS code require less power when compared to proposed technique. But here multiple faults was detected but was not corrected. However, for the proposed DMC, its decoding algorithm is quite simple so that the overheads are minimal. The issue is that the proposed technique requires more redundant bits compared with other codes.

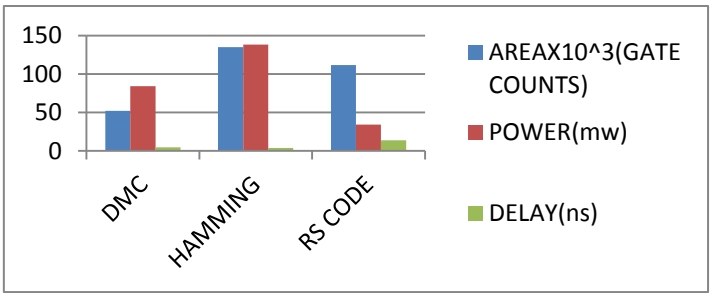


Fig. 6. Area, Power and Delay analysis of DMC, Hamming Code and RS Code

VI. CONCLUSIONS

In this project the novel per-code DMC was proposed to assure the reliability of memory. The proposed protection code utilized decimal algorithm to detect errors, so that more errors were detected and corrected. The obtained results showed that the proposed scheme has a superior protection level against large MCUs in memory. The advantage of using decimal algorithm is that the error detection capability is maximized so that the reliability of memory is enhanced. Besides, the encoder-reuse technique (ERT) is proposed to minimize the area overhead of extra circuits.

The only drawback of the proposed DMC is that more redundant bits are required to maintain higher reliability of memory. Therefore, future work will be conducted for the reduction of the redundant bits and the maintenance of the reliability of the proposed technique.

REFERENCES

[1] R. C. Baumann, "Radiation-induced soft errors in advanced semiconductor technologies," *IEEE Trans. Device Mater. Reliab.*, Vol. 5, no. 3, pp. 301–316, 2005.

[2] M. Zhang and N. R. Shanbhag, "Dual-sampling skewed CMOS design for soft-error tolerance," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, Vol. 53, no. 12, pp. 1461–1465, Dec. 2006.

[3] G. Torrens, B. Alorda, S. Barceló, J. L. Rosselló, S. A. Bota, and J. Segura, "Design hardening of nanometer SRAMs through transistor width modulation and multi-Vt combination," *IEEE Trans. Circuits Syst. II*, Exp. Briefs, Vol. 57, no. 4, pp. 280–284, Apr. 2010.

[4] C. L. Chen and M. Y.Hsiao, "Error-correcting codes for semiconductor memory applications: A state-of-the-art review," *IBMJ. Res. Develop.*, Vol. 28, no. 2, pp. 124–134, 1984.

[5] E. Ibe, H. Taniguchi, Y. Yahagi, K. Shimbo, and T. Toba, "Impact of scaling on neutron-induced soft error rate in SRAMs from a 250 nm to a 22 nm design rule," *IEEE Trans. Electron Devices*, Vol. 57, no. 7, pp. 1527–1538, Jul. 2010.

[6] R. K. Lawrence and A. T. Kelly, "Single event effect induced multiplecell upsets in a commercial 90 nm CMOS digital technology," *IEEE Trans. Nucl. Sci.*, Vol. 55, no. 6, pt. 1, pp. 3367–3374, Dec. 2008.

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

[8] S. Baeg, S.Wen, and R.Wong, "SRAM interleaving distance selection with a soft error failure model," *IEEE Trans. Nucl. Sci.*, Vol. 56, no. 4, pt. 2, pp. 2111–2118, Aug. 2009.

[9] 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.

[10] 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.

[11] 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.

