International Journal of Advanced Education and Research
ISSN: 2455-5746, Impact Factor: RJIF 5.34
www.newresearchjournal.com/education
Volume 1; Issue 5; May 2016; Page No. 04-07

Enhanced memory against multiple errors using decimal matrix code decoding circuits are more complex in these complicated codes

Sarojini KVE, Divya Swaroopa, Sirisha IB, Triveni P, Nagendra S, Satish P

Department of E.C.E, VITS College of Engineering (VSKV), Sontyam, Andhra Pradesh, India.

Abstract

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. Recently, matrix codes (MCs) based on hamming codes have been proposed for memory protection. The main issue is that they are double error correction codes and the error correction capabilities are not improved in all cases. In this paper, novel decimal matrix code (DMC) based on divide-symbol is proposed to enhance memory reliability with lower delay overhead. The proposed DMC utilizes decimal algorithm to obtain the maximum error detection capability. Moreover, the encoder-reuse technique (ERT) is proposed to minimize the area overhead of extra circuits without disturbing the whole encoding and decoding processes. ERT uses DMC encoder itself to be part of the decoder. The proposed DMC is compared to well-known codes such as the existing Hamming, MCs, and punctured difference set (PDS) codes. The obtained results show that the mean time to failure (MTTF) of the proposed scheme is 452.9%, 154.6%, and 122.6% of Hamming, MC, and PDS, respectively. At the same time, the delay overhead of the proposed scheme is 73.1%, 69.0%, and 26.2% of Hamming, MC, and PDS, respectively. The only drawback to the proposed scheme is that it requires more redundant bits for memory protection.

Keywords: Decimal algorithm, error correction codes (ECCs), mean time to failure (MTTF), memory, multiple cells upsets (MCUs)

1. Introduction

AS CMOS technology scales down to nano scale 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 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. In order to make memory cells as fault tolerant as possible, some error correction codes (ECCs) have been widely used to protect memories against soft errors for years. For example, the Bose–Chaudhuri–Hocquenghem codes.

Hamming distance and punctured difference set (PDS) codes have been used to deal with MCUs in memories. But these codes require more area, power, and delay overheads since the encoding and interleaving technique has been used to restrain MCUs, which rearrange cells in the physical arrangement to separate the bits in the same logical word into different physical words. However, interleaving technique may not be practically used in content-addressable memory (CAM), because of the tight coupling of hardware structures from both cells and comparison circuit structures.

Built-in current sensors (BICS) are proposed to assist with single-error correction and double-error detection codes to provide protection against MCUs. However, this technique can only correct two errors in a word.

More recently, in 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 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, 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. This section also illustrates the limits of simple binary error detection and the advantage of decimal error detection with some examples. The reliability and overheads analysis of the proposed code are analyzed in Section III. In Section IV, the implementation of decimal error detection together with BICS for error correction in CAM is provided. Finally, some conclusions of this paper are discussed and shared in Section V.

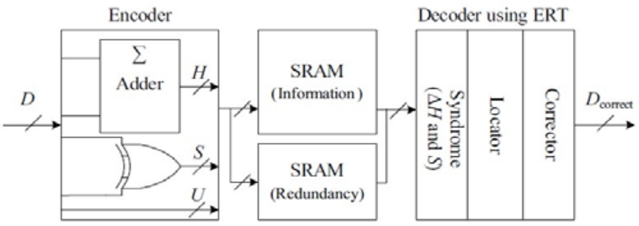


Fig 1: Proposed schematic of fault-tolerant memory protected with DMC.

2. Proposed DMC

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

A. Proposed Schematic of Fault-Tolerant Memory

The proposed schematic of fault-tolerant memory is depicted in Fig. 1. First, 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. 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 and will be introduced in the following sections.

B. Proposed DMC Encoder

In the proposed 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 ($N = k \times m$), and these symbols are arranged in a $k_1 \times k_2$ 2-D matrix ($k = k_1 \times k_2$, where the values of k_1 and k_2 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.

To explain the proposed DMC scheme, we take a 32bit word as an example, as shown in Fig. 2. The cells from $D0$ to $D31$ are information bits. This 32-bit word has been divided into eight symbols of 4-bit. $k_1 = 2$ and $k_2 = 4$ have been chosen simultaneously. H_0–H_{19} are horizontal check bits; V_0 through V_{15} are vertical check bits. However, it should be mentioned that the maximum correction capability (i.e., the maximum size of MCUs can be corrected) 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 correction capability and minimize the number of redundant bits. For example, in this case, when $k = 2\times2$ and $m = 8$, only 1-bit error can be corrected and the number of redundant bits is 40. When $k = 4\times4$ and $m = 2$, 3bit errors can be corrected and the number of redundant bits is reduced to 32. However, when $k = 2\times4$ 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 $k = 2\times4$ and $m = 4$ are utilized to construct DMC.

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

$$H_4H_3H_2H_1H_0 = D_3 D_2D_1D_0 + D_{11}D_{10}D_9D_8 \tag{1}$$
$$H_9H_8H_7H_6H_5 = D_7 D_6D_5D_4 + D_{15}D_{14}D_{13}D_{12} \tag{2}$$

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

$$V_0 = D_0 \square D_{16} \tag{3}$$
$$V_1 = D_1 \square D_{17} \tag{4}$$

and similarly for the rest vertical redundant bits.

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. The enable signal E_n will be explained in the next section.

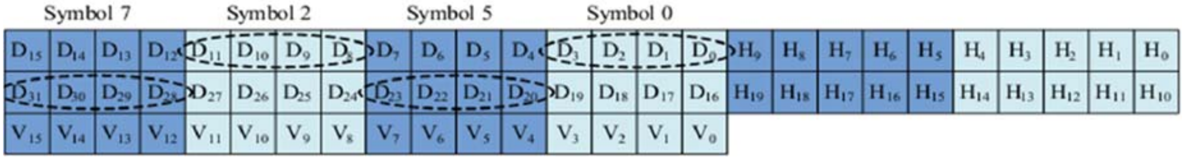
C. Proposed DMC Decoder

To obtain a word being corrected, the decoding process is required. For example, first, the received redundant bits $H_4\, H_3\, H_2\, H_1\, H_0$ and $V_0 - V_3$ are generated by the received information bits D. Second, the horizontal syndrome bits $H_4\, H_3\, H_2\, H_1\, H_0$ and the vertical syndrome bits $S_3 - S_0$ can be calculated as follows:

$$\square\, H_4H_3H_2H_1H_0 = H_4H_3H_2H_1H_0' - H_4H_3H_2H_1H_0 \tag{5}$$
$$S_0 = V_0' \square V_0 \tag{6}$$

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

When $H_4\, H_3\, H_2\, H_1\, H_0$ and $S_3 - S_0$ are equal to zero, the stored code word has original information bits in symbol 0 where no error occur.



Symbol 7				Symbol 2				Symbol 5				Symbol 0													
D_{15}	D_{14}	D_{13}	D_{12}	D_{11}	D_{10}	D_9	D_8	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	H_9	H_8	H_7	H_6	H_5	H_4	H_3	H_2	H_1	H_0
D_{31}	D_{30}	D_{29}	D_{28}	D_{27}	D_{26}	D_{25}	D_{24}	D_{23}	D_{22}	D_{21}	D_{20}	D_{19}	D_{18}	D_{17}	D_{16}	H_{19}	H_{18}	H_{17}	H_{16}	H_{15}	H_{14}	H_{13}	H_{12}	H_{11}	H_{10}
V_{15}	V_{14}	V_{13}	V_{12}	V_{11}	V_{10}	V_9	V_8	V_7	V_6	V_5	V_4	V_3	V_2	V_1	V_0										

Fig 2: 32-bits DMC logical organization (k=2x4 and m= 4). Here, each symbol is regarded as a decimal integer.

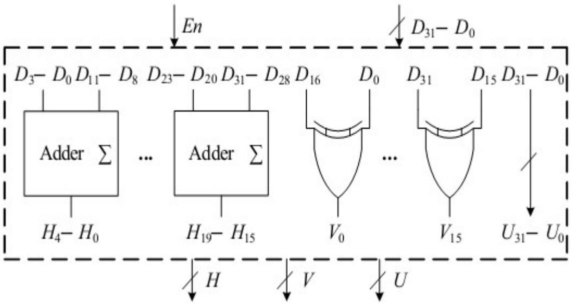


Fig 3: 32- bit DMC encoder Structure using multi bit Adders and X OR gates operations.

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

Fig 4: 32-bit DMC decoder structure using ERT.

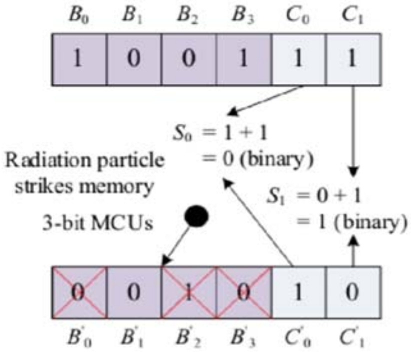


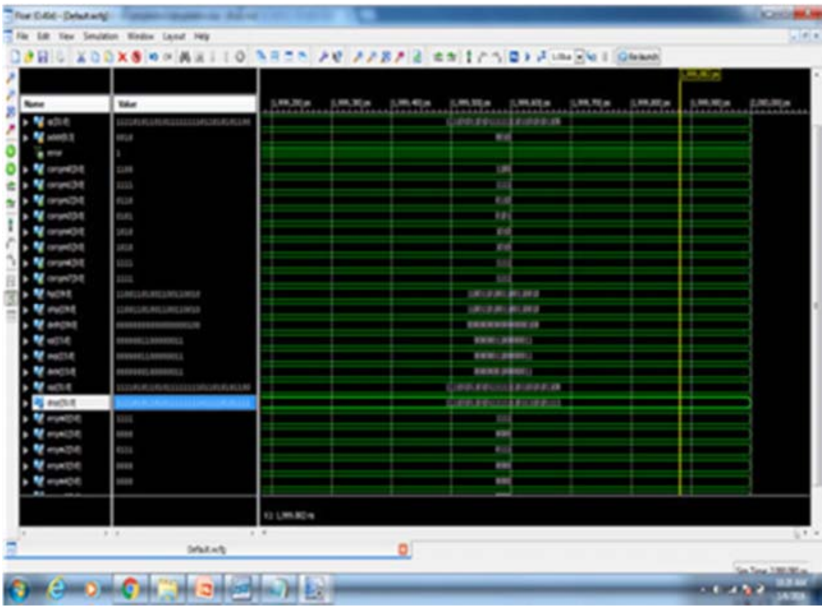
Fig 5: limits of binary error detection in simply binary

When are detected and located in symbol 0, and then these DMC encoder is also reused for obtaining the errors can be corrected by syndrome bits in DMC decoder. Therefore, the whole circuit area of DMC can be minimized as a result of

$$D \quad D \square S$$

using the existent circuits of encoder. Besides, this 0 correct =00. figure also shows the proposed decoder with an enable (7) The proposed DMC decoder is depicted in Fig. 4, signal *E n* for deciding whether the encoder needs to be which is made up of the following sub modules, and a part of the decoder. In other words, the *E n* signal is each executes a specific task in the decoding used for distinguishing the encoder from the decoder, process: syndrome calculator, error locator and error and it is under the control of the write and read signals corrector.it can be observed from this fig that the in memory. Therefore, in the encoding (write) process, redundant bits must be recomputed from the DMC encoder is only an encoder to execute the received information bits *D* and compared to the encoding operations. However, in the decoding (read) original set of redundant bits in order to obtain the process, this encoder is employed for computing the syndrome bits *H* and *S*. Then error locator uses *H* syndrome bits in the decoder. These clearly show how and *S* to detect and locate which bits some errors the area overhead of extra circuits can be substantially occur in. Finally, in the error corrector, these errors reduced. Can be corrected by inverting the values of error bits.

In the proposed scheme, the circuit area of DMC is *D. Limits of Simple Binary Error Detection* minimized by reusing its encoder. This is called the for the proposed binary error detection technique ERT. The ERT can reduce the area overhead of DMC in, although it requires low redundant bits, its error without disturbing the whole encoding and decoding detection capability is limited. The main reason for processes. From Fig. 4, it can be observed that its error detection mechanism is based on binary. We illustrate the limits of this simple binary error detection using a simple example. Let us suppose that the bits *B*3, *B*2, *B*1, and *B*0 are original information bits and the bits *C*0 and *C*1 are redundant bits shown in Fig. 5. The bits *C*0 and *C*1 are obtained using the binary algorithm



3. Advantage of decimal error detection

In the previous discussion, it has been shown that error detection based on binary algorithm can only detect finite number of errors. However, when the decimal algorithm is used to detect errors, these errors can be detected so that decoding error can be avoided. The reason is that the operation mechanism of decimal algorithm is different from that of binary. The detection procedure of decimal error detection using the proposed structure is fully described.

4. Conclusion

In this paper, novel per-word 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. Besides, the proposed decimal error detection technique is an attractive opinion to detect MCUs in CAM because it can be combined with BICS to provide an adequate level of immunity.

The only drawback of the proposed DMC is that more redundant bits are required to maintain higher reliability of memory, so that a reasonable combination of k and m should be chosen to maximize memory reliability and minimize the number of redundant bits based on radiation experiments in actual implementation. Therefore, future work will be conducted for the reduction of the redundant bits and the maintenance of the reliability of the proposed technique.

5. References

1. Radaelli D, Puchner H, Wong S, Daniel S. Investigation of multi-bit upsets in a 150 nm technology SRAM device, *IEEE Trans. Nucl. Sci.*, 2005; 52(6):2433-2437.
2. Ibe E, Taniguchi H, Yahagi Y, Shimbo K, Toba T. Impact of scaling on neutron induced soft error in SRAMs from an 250 nm to a 22 nm design rule, *IEEE Trans. Electron Devices*, 2010; 57(7):1527-1538.
3. Argyrides C, Pradhan DK. Improved decoding algorithm for high reliable reed muller coding, in *Proc. IEEE Int. Syst. On Chip Conf.*, 2007, 95-98.
4. Sanchez-Macian A, Reviriego P, Maestro JA. Hamming SEC-DAED and extended hamming SEC-DEDTAED codes through selective shortening and bit placement, *IEEE Trans. Device Mater. Rel.*, to be published.
5. Liu S, Reviriego P, Maestro JA. Efficient majority logic fault detection with difference-set codes for memory applications, *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, 2012; 20(1):148-156.
6. Zhu M, Xiao LY, Song LL, Zhang YJ, Luo HW. New mix codes for multiple bit upsets mitigation in fault-secure memories. *Microelectron. J.* 2011; 42(3):553-561.
7. Naseer R, Draper J. Parallel double error correcting code design to mitigate multi-bit upsets in SRAMs, in *Proc. 34th Eur. Solid-State Circuits*, 2008, 222-225.
8. Reviriego P, Flanagan M, Maestro JA. A (64, 45) triple error correction code for memory applications, *IEEE Trans. Device Mater. Rel.*, 2012; 12(1):101-106.
9. Baeg S, Wen S, Wong R. Interleaving distance selection with a soft error failure model, *IEEE Trans. Nucl. Sci.*, 2009; 56(4):2111-2118.
10. Pagiamtzis K, Sheikholeslami A. Content addressable memory (CAM) circuits and architectures: A tutorial and survey. *IEEE J. Solid-State Circuits*. 2003; 41(3):712-727.
11. Baeg S, Wen S, Wong R. Minimizing soft errors in TCAM devices: A probabilistic approach to determining scrubbing intervals, *IEEE Trans. Circuits Syst. I, Reg. Papers*, 2010; 57(4):814-822.
12. Reviriego P, Maestro JA. Efficient error detection codes for multiple-bit upset correction in SRAMs with BICS, *ACM Trans. Design Autom. Electron. Syst.*, 2009; 14(1):18:1–18:10.
13. Argyrides C, Chipana R, Vargas F, Pradhan DK. Reliability analysis of H-tree random access memories implemented with built in current sensors and parity codes for multiple bit upset correction, *IEEE Trans. Rel.*, 2011; 60(3):528-537.
14. Argyrides C, Pradhan DK, Kocak T. Matrix codes memory chips, *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, 2011; 19(3):420-428.
15. Argyrides CA, Lisboa CA, Pradhan DK, Carro L. Single element correction in sorting algorithms with minimum delay overhead, in *Proc. IEEE Latin Amer. Test Workshop*, 2009, 652-657.
16. Yahagi Y, Yamaguchi H, Ibe E, Kameyama H, Sato M, Akioka T, Yamamoto S. A novel feature of neutroninduced multi-cell upsets in 130 and 180 nm SRAMs, *IEEE Trans. Nucl. Sci.*, 2007; 54(4):1030-1036.
17. Mahatme NN, Bhuva BL, Fang YP, Oates AS. Impact of strained-Si PMOS transistors on SRAM soft error rates, *IEEE Trans. Nucl. Sci.*, 2012; 59(4):845-850.