

# DETECTION OF MULTIPLE CELL UPSETS IN MEMORY FOR ENHANCED PROTECTION

E. LAVANYA<sup>1</sup>, J. DHANAPATHI<sup>2</sup>

<sup>1</sup>PG Student, VLSI Design, <sup>2</sup>Assistant Professor (Sr.G)<sup>1,2</sup>Department of ECE

<sup>1,2</sup>Surya Group Of Institutions, India

## Abstract:

The Multi-cell upsets are those upsets affecting multiple cells of a memory, whatever correction words those cells happen to fall in. These multiple cell upsets (MCUs) have become a serious reliability concern in some memory applications. 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. In the existing system, novel decimal matrix code (DMC) based on divide-symbol is presented which 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 existing DMC is compared to well-known codes such as the Hamming, MCs, and punctured difference set (PDS) codes. The 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 existing scheme is 73.1%, 69.0%, and 26.2% of Hamming, MC, and PDS, respectively.

Index Terms—Decimal matrix code(DMC), error correction codes (ECCs), mean time to failure (MTTF), memory, multiple cells upsets (MCUs).

## I. INTRODUCTION

A soft error is a type of error, where a signal or datum is wrong. Errors may be caused by a defect, usually understood either to be a mistake in design or construction, or a broken component. A soft error is also a signal or datum which is wrong, but is not assumed to imply such a mistake or breakage. After observing a soft error, there is no implication that the system is any less reliable than before. In the spacecraft industry this kind of error is called a single bit upsets. Usually, only one cell of a memory is affected, although high energy events can cause a multi-cell upset. Conventional memory layout usually places one bit of many different correction words adjacent on a chip. So, even a multi-cell upset leads to only a number of separate single bit upsets in multiple correction words. 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, Reed–Solomon codes, 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 decoding circuits are more complex in these complicated codes. The Multi-cell upsets are those upsets affecting multiple cells of a memory, whatever correction words those cells happen to fall in. These multiple cell upsets (MCUs) have become a serious reliability concern in some memory applications. 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. In the existing system, novel decimal matrix code (DMC) based on divide-symbol is presented which 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. Computer memories are sensitive to soft errors which can affect system reliability. Memory cells can be disturbed by high-energy neutron particles from terrestrial atmosphere or alpha particles resulted from IC package material. The way to correct is to include error correction capabilities on the memory so that some of the errors can be corrected. This is normally done by using a single error correction (SEC) code on each memory word so that single errors in a word can be corrected. The combination of SEC and scrubbing is effective against single event upsets but not against MCUs as the errors in an MCU tend to be physically close and therefore it is likely that they affect more than one bit of the same memory word. Multiple cell upsets (MCUs) in highly scaled memory arrays, CPU registers etc., which are defined as simultaneous errors in more than one memory cell induced by a single event. An encoder module is efficiently designed for the encoding of the input data. A CAM memory is also designed and a decoder module which uses the same encoder is designed. The error detection is performed using an error signal indicator which is designed and the correction is also performed.

First, during the encoding (write) process, information bits  $D$  are fed to the DMC encoder, and the horizontal redundant bits  $H$  and vertical redundant bits  $V$  are obtained from the DMC encoder. When encoding process is completed, the obtained DMC codeword is stored in the memory. If MCUs happen in the memory, these errors can be corrected in the decoding (read) method. This paper is divided into the following sections. Design and analysis of Automatic Input generator is present in Section II. Design and encoder operation analyzed in section III. Design and analysis of memory ,decoding operation and Integration process are analyzed in Section IV. Finally, some conclusions of this paper are discussed and shared in Section V.

## II. PROPOSED APPROACH

### A. Proposed Schematic of Fault-Tolerant Memory

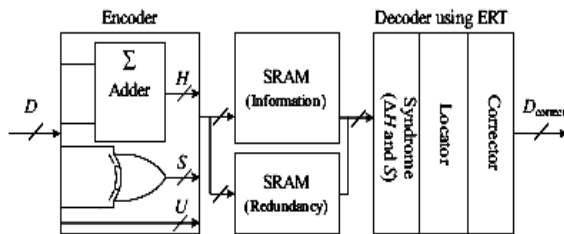


Fig. 1. Proposed schematic of fault-tolerant memory protected with DMC.

### B. Proposed DMC Encoder

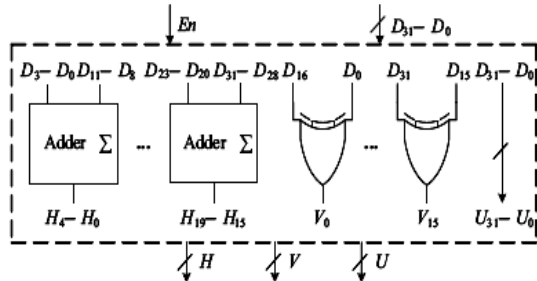


Fig. 2. 32-bit DMC encoder structure using multibit adders and XOR gates

The horizontal redundant bits  $H$  can be obtained by

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

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

For the vertical redundant bits  $V$ , we have

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

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

### C. Proposed DMC Decoder

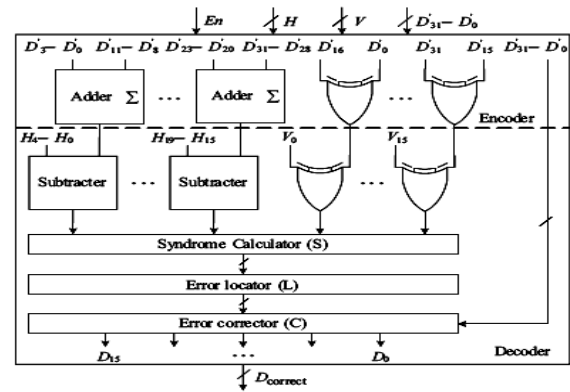


Fig. 3. 32-bit DMC decoder structure using ERT.

### D. Advantage of Decimal Error Detection

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)

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

$$= 1100 + 0110$$

$$= 10010. \quad (14)$$

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:

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

$$8 + D_3D_2D_1D_0$$

$$= 0111 + 1111$$

$$= 10110. \quad (15)$$

Then, the horizontal syndrome bits  $H_4H_3H_2H_1H_0$  can be

obtained using decimal integer subtraction

$$H_4H_3H_2H_1H_0 = H_4H_3H_2H_1H_0$$

$$H_4H_3H_2H_1H_0$$

$$= 10110 - 10010$$

$$= 00100.$$

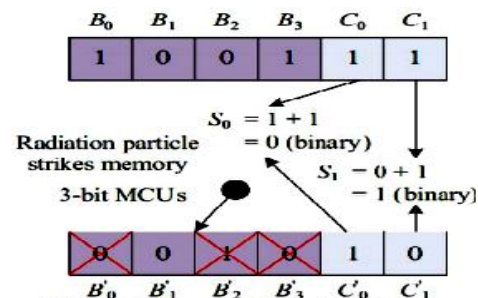


Fig .4 limits of binary error detection in simple binary operations

Besides, the figure shows the proposed decoder with an allow 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 manage of the write and read signals in memory. Therefore, in the encoding (write) mode, the DMC encoder is only an encoder to execute the encoding operations. However, in the decoding (read) mode, this encoder is employed for computing the syndrome bits in the decoder. These obviously show how the area overhead of extra circuits can be substantially decreased.

### III. RELIABILITY AND OVERHEADS ANALYSIS

#### A. Fault Injection

The correction coverage of PDS [9], MC [15], Hamming, and the proposed DMC codes are obtained from one million experiments. The results of coverage are shown in Table I. It can be seen that our proposed DMC have superior protection

$$\beta = \text{Redundant bits} / \text{Redundant bits} + \text{Information bits}$$

TABLE I

CORRECTION FOR COVERAGE (32-bit)

ECC Codes	The Number of Errors in a Word															
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
DMC (%)	100	100	100	100	100	32.6	84.7	76.0	66.7	60.9	54.5	47.7	40.0	31.6	22.3	11.8
PDS [9] (%)	100	100	100	0.8	0	0	0	0	0	0	0	0	0	0	0	0
MC [15] (%)	100	100	76.4	54.3	35.1	14.2	6.7	0.6	0	0	0	0	0	0	0	0
Hamming (%)	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TABLE II

REDUNDANT BITS (32-bit)

ECC	Information Bits	Redundant Bits	$\beta$	Note
DMC	32	36	52.9%	$k = 2 \times 4, m = 4$
DMC	32	32	50.0%	$k = 4 \times 4, m = 2$
PDS [9]	32	19	37.3%	Shorting and puncturing
MC [15]	32	28	46.7%	Correction capability is 2
Hamming	32	7	17.9%	Correction capability is 1

TABLE III

AREA, POWER, AND DELAY ANALYSIS OF ENCODER AND DECODER

ECC Codes	Area		Power		Delay	
	$\mu\text{m}^2$	%	mw	%	ns	%
DMC	41572.6	100	10.8	100	4.9	100
PDS* [9]	486778.1	1170.9	221.1	2047.2	18.7	381.6
MC [15]	77933.7	187.5	24.7	228.7	7.1	144.9
Hamming	58409.4	140.5	20.5	189.8	6.7	136.7

### IV. DECIMAL ERROR DETECTION IN CAM

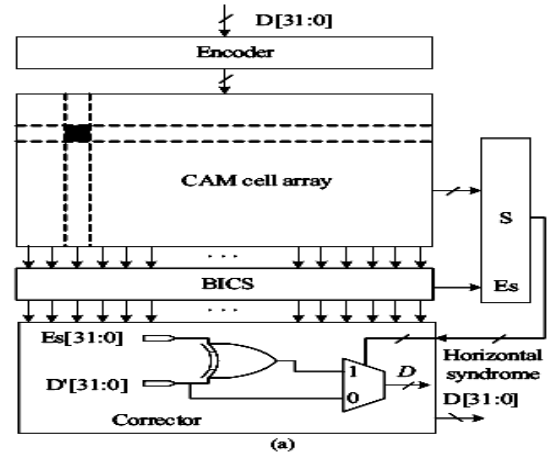
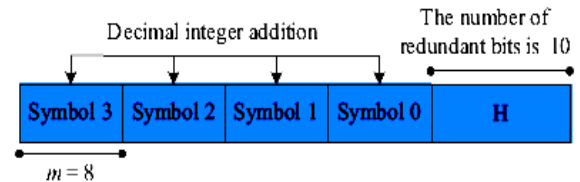


Fig. 5 (a) Proposed fault-tolerant CAM using decimal error detection

technique together with BICS. Note that when errors do not exist in CAM, the stored codeword is directly output without though error detection and correction circuits.


 (b) 32-bit word organization in CAM ( $k = 1 \times 4$  and  $m = 8$ ).

The memory is designed which consists of encoded data. Due to external disturbances the original data stored in the memory leads to bit flips. So the original data is retrieved from the memory using the read command and it is decoded. This decoder module uses the same encoder in order to avoid unnecessary area consumption. Finally the encoded and the decoded data are compared for error checking. Fig.1 An error signal detector is used for detection of errors and it is corrected.

ECC code is a very powerful technique to correct MCUs in memory, as mentioned before. However, ECC implementation in CAM is significantly different from its implementation in SRAM due to simultaneous access to all the words in CAM, so that ECC code is not suitable to directly protect CAM. In , BICS together with Hamming code is used to protect SRAM. Because BICS has zero fault-detection latency for multiple error detection, it is suitable for detection errors in CAM as well.

For the decimal error detection, this ability to detect any type of error can be useful in CAM. Let us consider that the decimal error detection technique combines with BICS to protect CAM. When MCUs occur in a word of CAM, for each error column, a momentary current pulse generated between power supply and ground. BICS can detect this current pulse and generate an error signal  $E_s$ , i.e., this  $E_s$  signal detects and locates columns which the errors occur in. At the same time, the syndrome calculation is active to detect the error row, i.e., (5) is performed row and row. Then in the error corrector these errors can be corrected. Finally, the correctable word is written back in CAM. Because the proposed decimal error detection technique can detect any number of errors in a word, the reliability of CAM has an adequate level of immunity to MCUs in a word. For example, when 32-bit errors occur in a word of CAM, the syndrome bits  $_H$  can detect these errors ( $_H$  can detect errors but cannot locate the precise upset locations; this is enough) and activate the syndrome calculation so that all errors can be corrected at the expense of least time consumption. The proposed module to increase the error correction rate compared to the 32-bit decimal matrix code. 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 include been proposed for memory protection. The main issue is that they are double error correction codes and the error correction capabilities are not enhanced in all cases. Moreover, the ERT (encoder-reuse technique) is proposed to reduce the area overhead of extra circuits exclusive of disturbing the total encoding and decoding processes. ERT use DMC encoder itself to be part of the decoder.

#### IV. SIMULATION RESULT

##### Error Detection and Correction using DMC decoder

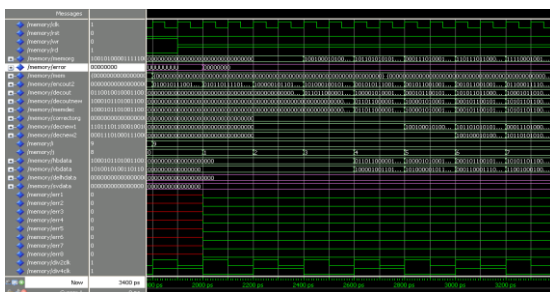
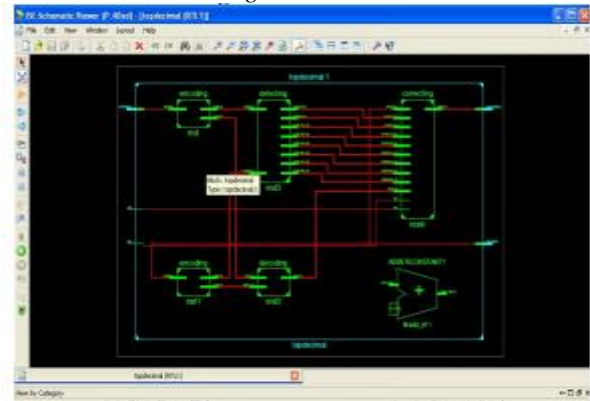


Fig 4.1 Error Detection and Correction using DMC decoder

#### 3.2 RTL Schematic diagram



#### V. CONCLUSION

Thus we had designed the encoder module which encodes the incoming original data. The encoding is done by calculating the redundant bits and combining of redundant bits and the original data. This is then stored in the memory which is also designed. The read and write signals are given to the memory operation. Then the data is retrieved from the memory and the decoding is performed which uses the same encoding logic and finally the decoded and the encoded data are compared for soft error detection and correction. All these are designed and verified successfully using Modelsim Simulator.

#### REFERENCES

- [1] S. Liu, P. Reviriego, and J. A. Maestro, "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, Jan. 2012.
- [2] 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.
- [3] K. Pagiampzis and A. Sheikholeslami, "Content addressable memory (CAM) circuits and architectures: A tutorial and survey," *IEEE J. Solid-State Circuits*, vol. 41, no. 3, pp. 712–727, Mar. 2003.
- [4] 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.
- [5] Saeed Shamshiri and Kwang-Ting Cheng, "Error-Locality-Aware Linear Coding to Correct Multi-bit Upsets in SRAMs" Jun.2010.
- [6] Jing Guo, Liyi Xiao, Member, IEEE, Zhigang Mao, Member, IEEE, and Qiang Zhao, "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.