

Hari Krishna Padi<sup>1</sup>, Mr. L. Narayanarao<sup>2</sup>

<sup>1</sup>Student, Dept of Electronics and Communication Engineering, Sri Venkateswara College of Engineering & Technology, Chittoor, A.P, India, *Harikrishnason@Gmail.Com*<sup>2</sup>Assistant Professor, Dept of Electronics and Communication Engineering, Sri Venkateswara College of Engineering & Technology, Chittoor, A.P, India, *Nari.Rao@Gmail.Com* 

**Abstract:** In this paper, novel per-word 64- bit DMC was proposed to assure the reliability of memory. Here to detect and correct maximum 32 errors. 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. 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. 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.

**Keywords:** 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, alphaparticle, 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, 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.

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

#### Hari Krishna Padi & Mr. L. Narayanarao

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.



Fig1. Proposed schematic of fault-tolerant memory protected with DMC.

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 simulation outputs are in section III. Finally, some conclusions of this paper are discussed and shared.

#### 2. PROPOSED DMC

In this section, DMC is proposed to assure reliability in the presence of MCUs with reduced performance overheads, and a 64-bit word is encoded and decoded as an example based on the proposed techniques.

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

The proposed schematic of fault-tolerant memory is depicted in Fig. 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.

#### 2.2. 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 × m), and these symbols are arranged in a k1 × k2 2-D matrix (k = k1 × k2, 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.

To explain the proposed DMC scheme, we take a 64-bit word as an example, as shown in Fig. 2. The cells from D0 to D63 are information bits. This 64-bit word has been divided into sixteen symbols of 4-bit. k1 = 2 and k2 = 4 have been chosen simultaneously. H0–H39 are horizontal check bits; V0 through V31 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  $k = 2 \times 2$  and m = 8, only 1-bit error can be corrected and the number of redundant bits is 80. When  $k = 4 \times 4$  and m = 2, 3-bit errors can be corrected and the number of redundant bits is reduced to 32. However, when  $k = 2 \times 4$  and m = 4, the maximum correction capability is up to 5 bits and the number of redundant bits is 72. In this paper, in order to enhance the reliability of memory, the error correction capability is first considered, so  $k = 2 \times 8$  and m = 4 are utilized to construct DMC.

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

H4H3H2H1H0 = D3D2D1D0 + D19D18D17D16	(1	)

#### H9H8H7H6H5 = D7D6D5D4 + D23D22D21D20

and similarly for the horizontal redundant bits H14H13H12H11H10, H19H18H17H16H15H16, H24H23H22H21H20, H29H28H27H26H25, H34H33H32H31H30 and H39H38H37H36H35 where "+" represents decimal integer addition.

For the vertical redundant bits V, we have

$$V0 = D0 \wedge D31 \tag{3}$$

$$V1 = D1 ^ D32$$

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. In this figure, H39 - H0 are horizontal redundant bits, V31 - V0 are vertical redundant bits, and the remaining bits U63 - U0 are the information bits which are directly copied from D31 to D0. The enable signal En will be explained in the next section.

#### 2.3. Proposed DMC Decoder

To obtain a word being corrected, the decoding process is required. For example, first, the received redundant bits H4H3H2H1H0' and V0'-V3' are generated by the received information bits D'. Second, the horizontal syndrome bits  $\Delta$ H4H3H2H1H0 and the vertical syndrome bits S3 – S0 can be calculated as follows:

## $\Delta H4H3H2H1H0 = H4H3H2H1H0' - H4H3H2H1H0$

S0=V0'^V0

And similarly for the rest vertical syndrome bits, where "–" represents decimal integer subtraction. When  $\Delta$ H4H3H2H1H0 and S3 – S0 are equal to zero, the stored codeword has original information bits in symbol 0 where no errors occur. When  $\Delta$ H4H3H2H1H0 and S3 – S0 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

```
D0correct = D0 \land S0
```

(7)

(5)

(6)

(2)

(4)

Syn	mbol	1		Sy	mbol	6		Sj	mbo	15			Symb	ol4		Sj	mbol	3		Sy	mbol	2		S	ymbo	1		Sy	mbol	0																					
d31	d30	d29	d28	ď27	d26	d25	d24	d23	d22	d21	d20	dl9	d18	d17	dló	d15	d14	d13	dl2	dll	dlØ	0	d8	ď	dó	dö	d4	d3	d2	dl	dØ	h19	h18	h17	hló	hl5	h14	hl3	h12	hll	hlO	Ð	h8	h7	hó	hS	M	hß	h2	hl	P0
d63	d62	d61	d60	d59	d58	d57	d56	<b>6</b> 55	d54	d53	d52	d51	d50	d49	d48	d47	d46	d45	d44	d43	d42	d41	d40	d39	d38	d37	d36	d35	d34	d33	d32	h39	h38	<b>h</b> 37	h36	h35	164	h33	h32	h31	130	h29	h28	h27	h26	h25	124	h23	h22	h21	h20
s31	s30	s29	s28	s27	s26	\$5	s/4	\$23	s22	s21	s20	s19	s18	sl7	sló	slö	s14	sl3	sl2	sll	slO	\$	s	sĨ	s6	ģ	s4	sð	\$2	sl	sO																				

**Fig2.** 64-bits DMC logical organization ( $k = 2 \times 8$  and m = 4). Here, each symbol is regarded as a decimal integer



Fig3. 64-bit DMC encoder structure using multi bit adders and XOR gates



Extra airauit	En s	ignal	Function
Extra circuit	Read signal	Write signal	Function
Encodor	0	1	Encoding
Encoder	1	0	Compute syndrome bits

Fig 4. 64-bit DMC decoder structure using ERT

The proposed DMC decoder is depicted in Fig, which is made up of the following sub modules, 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  $\Delta H$  and S. Then error locator uses  $\Delta H$  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.



Fig5. limits of binary error detection in simple binary operations

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, 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.

#### **3. SIMULATION RESULTS**



Block diagram

International Journal of Emerging Engineering Research and Technology





Constrained and a service of the service of th			(* *: P P B B P	2 🔁 🖻		2				
Properties       Option of the second of the s		<ul> <li>Design Overview</li> <li>Summary</li> </ul>			topdecimal F	roject Stat	us			
Model Have       Turdonal       Papelenciales       Software         Image Cardinal       Saftware       Saftware       Saftware       Saftware         Image Cardinal       Saftware       Saftware       Saftware       Saftware       Saftware         Image Cardinal       Saftware		IOB Properties	Project File:	decimalmatrix	xise	Parser Err	ors:		No Errors	
Image Front		- 👔 module Level Utilization	Module Name:	topdecimal		Implemen	tation State:		Synthesized	
Sol Adjoint       Produkt Versione       EE 14.3       ***armage:       No Warmage:		- 👔 Pinout Report	Target Device:	xc3s500e-4fg	320	•Err	ors:		No Errors	
United and the Marger Performation		- Olock Report	Product Version:	ISE 14.3		•₩a	rnings:		No Warnings	
By Mode Network       Design Strategy:       Bit Cold at Labola       Timing Constraints:         By Mode Network       Status:       Find Timing Score:       Status:         By Mode Network       Status:       Status:       Status:       Status:         By Mode Network       Status:       Status:       Status:       Status:       Status:         Status:       Status:       Carret       Wetwork       Status:       Status:       Status:       Status:         Status:       Status:       Status: </td <td>E</td> <td>Errors and Warnings</td> <td>Design Goal:</td> <td>Balanced</td> <td></td> <td>•Ro</td> <td>uting Results:</td> <td></td> <td></td> <td></td>	E	Errors and Warnings	Design Goal:	Balanced		•Ro	uting Results:			
Indian management       States Sattrag       Final Training Source:         Indian management       Indian management       Indian management         Indian management       Indian management       Indian management       Indian management         Indian management       Indian management       Indian management       Indian management         Indian management       Indian management       Indian management       Indian management         Indian management       Indian management       Indian management       Indian management		- 📔 Parser Messages	Design Strategy:	Xiinx Default	(unlocked)	•Tin	ning Constraints	5:		
Decked Space         Operating Testing Testing         Operating Testing Testing Testing         Operating Testing Testing		- Synthesis Messages	Environment:	System Settir	95	•Fin	al Timing Score	:		
Device Unknown Summary (standard values)         Image Results         Device Unknown Summary (standard values)         Image Results         Device Unknown Summary (standard values)         Device Summary         Standard Values       Device Summary         Standard Values       Device Summary		Map Messages								
Inter Process       Used       Available       IBitation         Inter of Stores       104       4458       104       4458         Inter of Stores       104       4458       104       4458       104         Inter of Stores       104       4458       104       4458       104       4458       104       4458       104       4458       104       4458       104		Place and Route Messages     Timing Messages		Device U	tilization Summary (	estimated v	values)			[
Image: Status Status       184       455       1         Image: Status Status       336       9112       1         Image: Status Status       336       9112       1         Image: Status Status       1       24       1         Image: Status       1       24       1       24         Image: Status       1       24       1       0       0       1       1       24       1       1       24       1       1       24       1       1       24       1       1       24       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1<		📄 Bitgen Messages	Logic Utilization	Use	d	Availabl	e	Utilizatio	on	
Image: Strates Stopet       Image: Stopet Stopet       Image: Stopet St	Ē	All Implementation Messages     Detailed Reports	Number of Slices		18	14	46	556		3
Invalue Root       Invalue of boodel USs       19       220       8         Invalue of boodel USs       1       24       24         Invalue of GUIS       1       24       1       24         Invalue of GUIS       1       24       0       0       0         Invalue of GUIS       Invalue of GUIS       1       24       0       0       0         Invalue of GUIS       Invalue of GUIS       Invalue of GUIS       1       24       0	Ī	- 📄 Synthesis Report	Number of 4 input LUTs		32	6	93	312		3
Image Report       Image Report         Image Report       Report Name         Standard Report       Standard Report         Standard Report		- 7 Translation Report	Number of bonded IOBs		19	14	2	232		83
Description       Description         Description       Report Name       Status       Cenerated Report       Troos       Warnings       Infos         Standar Reports       Report Name       Status       Cenerated Report       Report Name       Status       Cenerated Report       Infos         Description       Status       Cenerated Report       Report Name       Status       Cenerated Report       Infos         Description       Control       Out of Date       Prime       Status       Cenerated Report       Infos         Description       Control       Out of Date       Prime       Status       Cenerated Report       Infos         Description       Control       Out of Date       Prime       Status       Cenerated Report       Cenerated		- 👔 Map Report - 👔 Place and Route Report	Number of GCLKs			1		24		4
Bender Report       Detailed Reports         Bender Report       Report Name       Status       Central Lef 755 2014       0       0       0         Status       Control Lef 755 2014       0       0       0       0       0         Status       Control Lef 755 2014       0       0       0       0       0       0         Status       Control Lef 755 2014       0		- 🧃 Post-PAR Static Timing Report	_							
Concreter Reports     Concreter Report     Concrete Report		Bitgen Report			Detailed Repo	rts				
Deep Deckets       perdex Report         Decket Report       Current       Wed Xn 412-0755304       0       0       0         Decket Report       Decket Report       0       0       0       0       0         Decket Report       Decket Report       0       0       0       0       0       0         Decket Report       Decket Report       0	Ė	Secondary Reports	Report Name	Status	Generated		Errors	Warnings	Infos	
spond ben Somer Controls Some First Some First Som	De	esign Properties	Synthesis Report	Current	Wed Jun 4 12:4	7:55 2014	0	0	0	_
Image: Store Errors       Date Report       Out of Date       Fri May 23 1142-93 2014       0       0       21265 (2 mov)         Bow Filling Constraints       Store Errors       Date Report       Out of Date       Fri May 23 1143-53 2014       0       0       1126 (1 mov)         Power Report       Date Report       Out of Date       Fri May 23 1143-53 2014       0       0       1126 (1 mov)         Power Report       Description       Description       Out of Date       Fri May 23 1143-53 2014       0       0       1126 (1 mov)         Bage Report       Description       Description <td< td=""><td>)¢</td><td>ptional Design Summary Contents</td><td>Translation Report</td><td>Out of Dat</td><td>e Fri May 23 11:4</td><td>2:30 2014</td><td>0</td><td>0</td><td>0</td><td></td></td<>	)¢	ptional Design Summary Contents	Translation Report	Out of Dat	e Fri May 23 11:4	2:30 2014	0	0	0	
Show Kening         Show Kening         Show Kening         Show Kening         Show Kening         Power Report         Power Report         Statution Report         Statutio		- Show Clock Report	Map Report	Out of Dat	e Fri May 23 11:4	2:43 2014	0	0	<u>2 Infos (2 n</u>	<u>em)</u>
Store Errors       Power Report       Que of Date       Fir May 23 1145:05 2014       0       0       6.0756/65.00000         Bigen Report       Secondary Reports       Secondary Reports       Secondary Reports         Design Summary       Statut       To Design Summary       Statut       To DoumentLue       \$		- Show Warnings	Place and Route Report	Out of Dat	e Fri May 23 11:4	3:25 2014	0	0	<u>1 Info (1 ne</u>	<u>w)</u>
Image: Start       Design Summary         Start       Image: Start         Image: Start       Image: Start		- Show Errors	Power Report							
Bitgen Report         Design Summay         Start ()       index - mo, ? downdada ?? Zicharce () Window + ICE Project () Design Sum () Document () / / / / / / / / / / / / / / / / / /			Post-PAR Static Timing Report	Out of Dat	e Fri May 23 11:4	3:36 2014	0	0	<u>6 Infos (6 n</u>	em)
Start C Procena. C durdeda. C 2000 C C C C C C C C C C C C C C C C			Bitgen Report							
bar (P.4000) - (Petauthace)         -           ie Edit Yew Sindaton Windum Layot Help         - </th <th></th> <th>start 🤊 Inbox - ma 🄁 char</th> <th>ndraka 🔁 7.Erhance Desi</th> <th>a window</th> <th>• 🗲 ISE Project</th> <th>🐮 Des</th> <th>ign Sum 関</th> <th>Document1</th> <th>. 🔇 🕏 🛡</th> <th>11:2</th>		start 🤊 Inbox - ma 🄁 char	ndraka 🔁 7.Erhance Desi	a window	• 🗲 ISE Project	🐮 Des	ign Sum 関	Document1	. 🔇 🕏 🛡	11:2
a         b         b         c		start Sinbox-ma 🔁 char	ndraka 🔁 7.Erhance Desi	≌2 Window	• 🗲 ISE Project mmary	T Des	ign Sum 💆	Dorument I	. 🔇 🐉 🔰	11:2
Value         100100         200 nc         100 nc         800 nc           * did         100000         10000011100100001100100101010101000010000	lo	start Dirbox-ma Zichar	ndraka 🔁 7.Erhance Desi	≌² Window	• 🗲 ISE Project mmary	2 Des	ign Sum 👿	Document1	. ¢3 V	11:2
Name         Value         P00rs	lo	oat (2,40xd) - (Default.wcfg) e Edit Wew Smuldion Window Lay	otrala 727.Erhance Desi ov. Help or AA Sec. 1 (2015)	≥ 2 Window	► ISE Project mmary	I Des	ign Sum 🗹	Cocument1	<ul> <li></li> <li></li> <li></li> <li>√3 1000</li> </ul>	11:2
Note:         Fore:         Prove         Prove <td< td=""><td>lo ile</td><td>start Diber-na Constant ant (P.40xd) - [Default wcfg] e Edit Ver Smidden Window Laye P 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2</td><td>drala 🔁 7. Enhance Desi хл. Нер С (А) (Х) (С (О) (П) С (А) (Х) (С (О) (П) С (С (С</td><td>≥ 2 Window gn su</td><td>• ]&gt; ISE Project mmary ۶ № ] ۶ ۶ 8</td><td>2 Des</td><td>ign Sum 🗹</td><td>) Document 1</td><td><ul> <li>\$\$\$ \$\$</li> <li>\$\$\$\$ \$\$</li> <li>\$</li></ul></td><td><ul> <li>11:2</li> <li>-</li> <li< td=""></li<></ul></td></td<>	lo ile	start Diber-na Constant ant (P.40xd) - [Default wcfg] e Edit Ver Smidden Window Laye P 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2	drala 🔁 7. Enhance Desi хл. Нер С (А) (Х) (С (О) (П) С (А) (Х) (С (О) (П) С (С	≥ 2 Window gn su	• ]> ISE Project mmary ۶ № ] ۶ ۶ 8	2 Des	ign Sum 🗹	) Document 1	<ul> <li>\$\$\$ \$\$</li> <li>\$\$\$\$ \$\$</li> <li>\$</li></ul>	<ul> <li>11:2</li> <li>-</li> <li< td=""></li<></ul>
★ dout[ 100000] 100000000000000000000000000000	la ile	start Diber-na Ficher ast (P.40xd) - (Default wcfg) e Edi Ver Smiddin Wrdev Laye P E E E E E E E E E E E E E E E E E E E	dda ♥? f.there Desi xt. Heb CM M( () ↓ 1 (0) ♥ CM CF	≥ 2 Window gn su	ISE Project	X Des	ign Sum 😨	) Document I	<ul> <li>              x</li></ul>	<ul> <li>■ 11:2</li> <li>■ 11:2</li></ul>
li de de 0 li de ret 0 M de ries 1000000 0000000000 000000000 000000000		start Sibber-na ∮char ast (P.40xd) - [Default.wcfg] e Edt Ver Smiddin Window Lay P 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2	dala ♥ 7.trhance De si xx: Heb c: Ma & 1 1 0 € 00.ns:	2 Window 2 Window 2 Window 2 Window 2 Window 2 Window 2 Window 2 Window 2 Window		1 Des	ign Sum	) Document I	<ul> <li></li></ul>	11:2 5 ▼ €
terk 0		start         Triber - na         P. dv/           out (P,40xd) - [Default wcfg]         e         bit ver         smalletin           e         58 ver         smalletin         wrfge           p         1         1         x         x           p         1         1         x         x         x           p         1         1         x         x         x           p         4         100000         000000000000000000000000000000000000	dada ₹7.5mmarca Desi oxt Hep © M	2 Window     2 Window     300 0 S	ISE Project           Immary           Immary           Image: Ima	X Des X Des 500 ns 0000101010 0000001010	ign Sum	) Document I (* *) [ ] [ ] [ ] [ ] [ ] [ ] [ ] [ ] [ ] [	.  .	2 11.2 
	la ile	start Sibbor-na	zdrała	C Window	ISE Project           IMMARY	20 Des 2007 rs 2000 rs 20000010100 20000010100	ign Sum	) Document I	. ¢≱ ₩	2 11:2 - s V (-) 800 n
		start Sibbor-na	drata	B C C C C C C C C C C C C C C C C C C C		12 Des 500 rs 0000010100 0000001010	ign Sum	) Document I	. (c) ≥ ↓ .	<ul> <li>■</li> <li>■</li></ul>
		start Store-na	drala ₹7.frhance Desi xxt. Heb CM M 20 1 1 0 13 00 re	2 Window      300 ns		20 Des 500 ns 500 ns 00000101010 00000101010	ign Sum	Document1	. (¢)≥ ↓ . 1.000 709.000 ns 10 ns	<ul> <li>■</li> <li>■</li></ul>
		start () () () () () () () () () () () () ()	dela ₹7.frhance Desi xat. Heb CP AM & 1 1 0 13 00 ns	C 2 Window		20 Des 500 rs 500 rs 0000010100 0000001010	ign Sum	Document1	. (¢)≥ ↓↓ . 1.00u . 2000 ns . 0 ns	2 11:2
		start ( ) Diber-na ( ) of the out (P.40xd) - (Default wer(g) • Edt Ver Smidsten Wrder Lay ) 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	dela ₹7.frhance Desi oz. Heb CM M & 1 0 5 00 ns00 ns1 000 01	2 Window		22 Des 500 rs 500 rs 500 rs 60000101010 00000101010 00000101010	ign Sum 2	) Document I	. ¢3 ₩	2 11:2 5 ▼ ↓ 2 800 n
		start         Triber - na         P. dw           nat (P,40xd) - [Default wcfg]             a E8 Ver         Smidein Wrder Lay            P 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2	dala ₹7.frhance Desi xat Heb ~ △ ▲ 1 ④ 5 00ns _ 200ns _ 1 0000 / 1	2 Window     3 2 Window		12 Des 500 rs 500 rs 000001010	ign Sum 2	Document1	. () ≥ ↓ . (1.00u) . (	<ul> <li>■</li> <li>■</li></ul>
		start Circlese - na	trida	2 Window			ign Sum	Document1	. () ≥ ↓ . (1.00u) 709.000 ns 100 ns	<ul> <li>■</li> <li>■</li></ul>
		start      Sibor-na      Control - Default.wc/g      Default.w	drata			2 Des	ign Sum	Document1	. () ≥ ↓ </td <td></td>	
		start () () () () () () () () () () () () ()	dola ₹7.frhance Desi xxt. Heb c4 M 26 1 1 0 5 00 nc			2 Des	agn Sum	Doometti		<ul> <li>■ 11:2</li> <li>■ 11:2</li></ul>
		start Store-na	dola 77 finance Desi xet Heb ci M & 1 0 3 0113 10 1000 11 1000 1000 11 1000 10 1000000	2 Wroken		12 Des 12 Des	agn Sum		<ul> <li></li></ul>	■ 11:2
		start () () ibor-na () () on out (P,40xd) - () Default.wcfg) e Edt Wer Smulston Wrdow Law ) () () () () () () () () () () () () ()	dela ₹7.frhance De si ost Heb CM M & 1 0 5 00 ns0 00 ns1 00 ns1 00 ns1				ign Sum	Dourseti	. (€) (2) (1.000) (209.000 ns (209.000 ns	
		start Circler-na	zdrała			2 Des	ign Sum	Counerti	. (€) (2) (1.00. 209.000 ns 209.000 ns	
		start	trida				ign Stm	Dourseti	. (€) (2) (1000) 709.000 ns 100 ns	
		start	Adda 7 7 Erhance Desi xxt Heb CM M R I T O F TO R TO R				agn Sum	Doursetti	- C 200 1.00U	2 11:2 5 V 43 800 nr
		start () () () () () () () () () () () () ()	dola 77 Erhance Desi xxt Heb CM M R I T O R 00 re 1 0 10 10 10 10 10 10 10 10 10 10 10 10	роон роон роон роон соотолитони		20 Des	agn Sum	00000000	<ul> <li></li></ul>	
		start () () ibox-na () () ibox- at (P.40x0) - () Default.wcfg) e Edt Vere Smulaton Wrdow Lay } () () () () () () () () () () () () ()	dela 7 finance Desi xet Heb CM M & 1 0 5 00 rs 200 rs 1			20 Des	agn Sum		. (€) ≥ ↓	
		start Circler-na	trida				ign Sum	Pouretti.	. (€) (2) (1.000) 209.000 ns 00 ns	
		start	totala			12 Dec	agn Sum	Douverti.	<ul> <li></li></ul>	

Simulation output waveform

- 🔁 2 Window... - 🏷 ISE Project... 🔯 Document1... 🔝 ISim (P.40xd) 🛛 🔤 Flost (P.40... 🔇 🔊 🛡

### 4. CONCLUSION

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.

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

> <

٨.

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.

#### REFERENCES

- [1] D. Radaelli, H. Puchner, S. Wong, and S. Daniel, "Investigation of multi-bit upsets in a 150 nm technology SRAM device," IEEE Trans. Nucl. Sci., vol. 52, no. 6, pp. 2433–2437, Dec. 2005.
- [2] 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.
- [3] C. Argyrides and D. K. Pradhan, "Improved decoding algorithm for high reliable reed muller coding," in Proc. IEEE Int. Syst. On Chip Conf., Sep. 2007, pp. 95–98.
- [4] 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.
- [5] S. Liu, P. Reviriego, and J. A. Maestro, "Efficient majority logic fault detection with differenceset codes for memory applications," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 20, no. 1, pp. 148–156, Jan. 2012.
- [6] M. Zhu, L. Y. Xiao, L. L. Song, Y. J. Zhang, and H. W. Luo, "New mix codes for multiple bit upsets mitigation in fault-secure memories," Microelectron. J., vol. 42, no. 3, pp. 553–561, Mar. 2011.
- [7] 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.
- [8] 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.
- [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] S. Baeg, S. Wen, and R. Wong, "Interleaving distance selection with a soft error failure model," IEEE Trans. Nucl. Sci., vol. 56, no. 4, pp. 2111–2118, Aug. 2009.
- [11] K. Pagiamtzis 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.
- [12] S. Baeg, S. Wen, and R. Wong, "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, Apr. 2010.
- [13] P. Reviriego and J. A. Maestro, "Efficient error detection codes for multiple-bit upset correction in SRAMs with BICS," ACM Trans. Design Autom. Electron. Syst., vol. 14, no. 1, pp. 18:1– 18:10, Jan. 2009.
- [14] C. Argyrides, R. Chipana, F. Vargas, and D. K. Pradhan, "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., vol. 60, no. 3, pp. 528–537, Sep. 2011.
- [15] 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.
- [16] 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.
- [17] Y. Yahagi, H. Yamaguchi, E. Ibe, H. Kameyama, M. Sato, T. Akioka, and S. Yamamoto, "A novel feature of neutron-induced multi-cell upsets in 130 and 180 nm SRAMs," IEEE Trans. Nucl. Sci., vol. 54, no. 4, pp. 1030–1036, Aug. 2007.

#### **AUTHORS' BIOGRAPHY**

**Hari Krsihna Padi** received his degree from Department of Electronics and Communication. Currently he is working towards his maters with specialization of VLSI in Sri Venkateswara College of Engineering and Technology, R.V.S Nagar, Chittoor(Dist.), A.P., INDIA.

**Mr. L. Narayana Rao** received his degree of B.Tech from the Department of ECE. Currently working as Assistant Professor, Department of ECE in Sri Venkateswara College of Engineering and Technology, R.V.S Nagar, Chittoor (Dist.), A.P., INDIA