International Journal on Recent and Innovation Trends in Computing and Communication ISSN: 2321-8169
Volume: 4 Issue: 4 654 - 659

Memory Reliability Enhancement against Multiple Cell Upsets Using Decimal Matrix Code for 32-Bit Data

Satyabala
Mtech Scholar Department Of Electronics & Communication,
NRI Institute Of Information Science & Technology,
Bhopal, India
e-mail: paliwasnik_2008@yahoo.co.in

Assistant Prof. Akhilesh Jain
Assistant Prof. Department Of Electronics & Communication,
NRI Institute Of Information Science & Technology,
Bhopal, India
e-mail: akhilesh.electronics@gmail.com

***Abstract*—**An important issue in the reliability of memories exposed to radiation environment is transient multiple cells upsets (MCUs). To protect the memory data from radiations and transients many improved packaging techniques are available. But, a particular packaging provides protection from a limited variation of radiations. Today the devices are exposed to a very wide range of environment radiations due to increasing applications in the field of wireless communication. So some additional data preservation techniques are always preferred for authenticating the data before it is processed. Some of these techniques use encoded data to be stored in memories. These techniques are error correction codes (ECCs). It is always preferred to implement an error correction code that requires a less number of redundant bits to be stored and a minimized delay overhead in data correction. This paper presents an FPGA based implementation of memory data error detection and correction code that involves simple decimal addition algorithm in the encoding of data that is to be stored in memory. The decoding of the data for error detection and correction is based on the Hamming Code. This technique involves divide-symbol concept to represent the linear data in groups to make symbolic code. The length of the symbol is inversely proportional to the delay overhead of the code..

Keywords- *Decimal Addition, Error Correction Codes (ECCs), Error Syndrome, Hardware Memory, Multiple Cell Upsets (MCUs).*

*****

I. INTRODUCTION

The soft error rate in memory cells is rapidly increasing because of scaling down of CMOS technology deep nanoscale and increasing size of embedded memories in electronic systems that are exposed to space environment radiations. The radiations that have ionizing character affect the charge stored as data in semiconductor memory leading to soft error in the memories. In the memories that are exposed to the environmental radiations, the phenomena of error generation due to radiation effect are show with the help of memory block in figure 1.

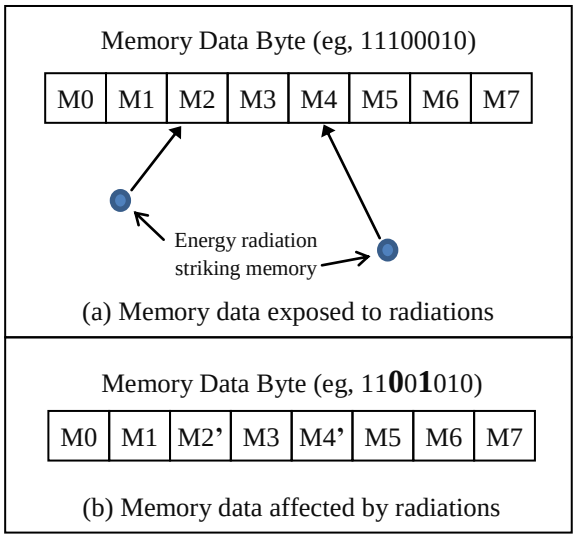


Figure1. Soft Error in Radiation Exposed Semiconductor Memory

Although a major concern about memory reliability is single-bit error but in some cases "multiple-bit error" or "multiple cell upset" (MCUs) becomes a serious reliability concern. In order to tolerant the faults in the memory up to the maximum possible extent, some error correction codes (ECCs) have been suggested by many researchers and they have been widely used for years. A simple block diagram of the fault tolerant memory encoder implementation is shown in figure 2.

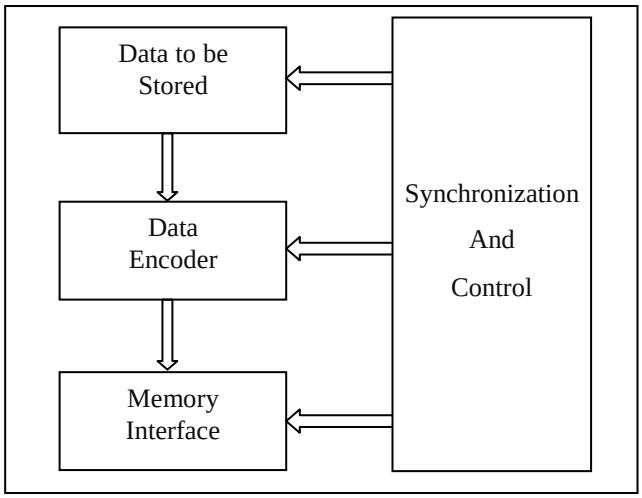


Figure2. A simple data encoder block diagram

In fault tolerant memories the data to be stored is first encoded to generate redundant bits with the help of the encoder circuit. The memory interface stores the data and the redundant bits in the memory. These redundant bits are used by the decoder in correcting the errors from the memory data. Few of the most reliable codes include Bose-Choudhary-Hocquenghem (BCH) code, Reed-Solomon (RS) code and Punctured Difference Set (PDS) code. These codes have been used to deal with MCUs in memories. Interleaving technique has also been used to restrain MCUs that rearranges cells in the physical arrangement to separate the bits into different physical words from the same logical word. The tight coupling

hardware architecture from both cells and comparison circuit structure might not practically allow interleaving to be effectively used with content-addressable memory (CAM). A recently proposed 2-D matrix code (MC) efficiently correct MCUs per word. In 2-D matrix code one word is divided into multiple rows and multiple columns, where the bits per row are protected by Hamming Code and the bits per column are protected by parity code. On detecting two errors by the Hamming the vertical syndrome bits are used to correct these errors. The 2-D MC is capable of correcting only two errors in all cases. This code has a lower delay overhead as compared to other codes. In this paper a novel decimal matrix code (DMC) is proposed. The operation of the proposed code is based on divide-symbol to provide enhanced memory reliability. The proposed DMC utilizes decimal integer addition (decimal algorithm) on the divided symbols of binary code. In the proposed work the decoder involves a logic comparator to find the error syndrome bits to detect and correct error. The decimal algorithm enhances the reliability of the error detection capability of the code. The rest of this paper is arranged as follows: section-II presents the work published by some recent scholars under the title 'Literature Review'. Section-III presents the proposed design of DMC Encoder and Decoder. The simulation and synthesis based results and comparative analysis of the proposed designs are given in section-IV. Finally the conclusion based on the proposed work is discussed in section-V.

II. LITRATURE REVIEW

In [1] a novel per-word DMC was proposed to pledge the reliability of memory by utilizing decimal algorithm to detect errors, so that more errors can be detected and corrected. The ERT based decoder architecture proposed in [1] is shown in figure 3. A comparative study of various error correction codes that defines various alternates to overcome reliability issue of radiation exposed memories is discussed in [2].

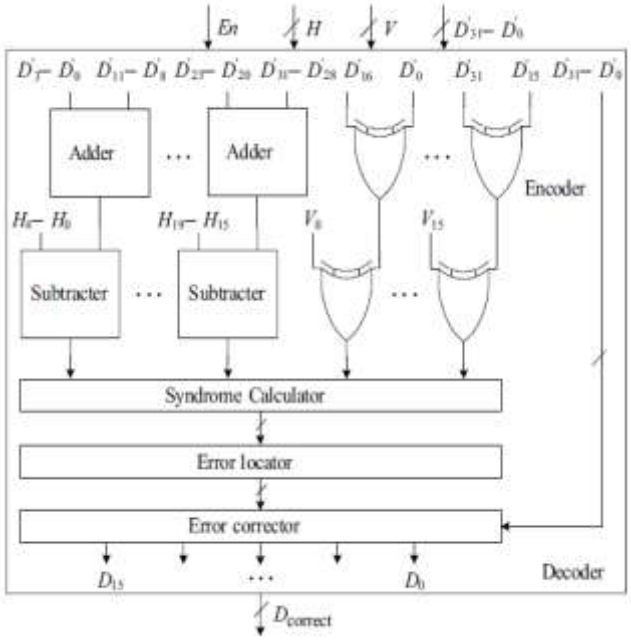


Figure3. An ERT based DMC Encoder and Decoder Design architecture in reference [1]

A high reliability decimal code for implementation of error correction technique using hamming code is proposed in [3] and [4]. Reference [5] proposes a mechanism for: (i) single error correction, double error detection triple-adjacent error detection using hamming code, and (ii) single error correction, double error detection, double adjacent error correction Codes. These are derived From Orthogonal Latin Square Codes. This paper also proposed a combination code that provides correction up to double adjacent errors. A CAM data protection scheme using DMC is proposed in [6]. References [7, 8] show implementation of decimal matrix code and parity matrix code for error correction. These references show the synthesis results against multiple bit error correction in memory data. In [9], Carry Save Adder (CSA) based decimal error detection technique is implemented on a 2-D arranged memory data. An enhanced error detection technique using Hybrid Matrix Code (HMC) is proposed in [10] with the results that shows a better quality protection level against large MCUs in memory. A decimal matrix code technique implementation is performed in [11, 12, 13] to obtain better performance of DMC against Hamming code. Reference [14] proposed a DMC architecture that is capable of correcting up to 5-bit error. An implementation of DMC and HMC are performed in [15-16] for error correction with high reliability. An Encoder Reuse Technique (ERT) based DMC implementation is performed in [17-18] to reduce the area of the proposed algorithm design.

III. PROPOSED DESIGN OF DECIMAL MATRIX CODE ENCODER AND DECODER

In the proposed work, the DMC encoder implementation follows the division of the 32-bit data and then encoding the data to generate redundant bits. The proposed architecture of fault-tolerant memory is depicted in figure 4.

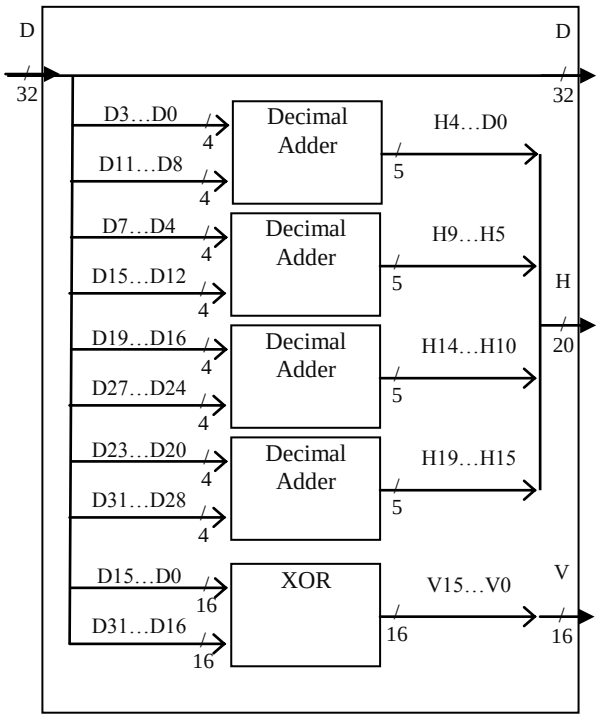


Figure4. Proposed 32-bit Architecture of DMC Encoder

In divide-symbol step N-bit data is divided into K-symbols of m-bit size, thus, N = K x m, and then the K symbols are arranged in a 2-D matrix of size k1 x k2, where k1 and k2 represents the number of row and column respectively. Each m-bit symbol is identified to generate the horizontal redundant bits by performing decimal integer addition of selected symbols per row. Each symbol is considered as a decimal integer to perform decimal addition. The vertical redundant bits are obtained by performing parity generation operation among the bits of the column. The proposed DMC does not need to change the physical structure of the memory as the steps are implemented in logical form. During the encoding process the 32-bits of data, D32…D0 are considered into symbols of 4-bit and arranged in a matrix of 2 rows x 4 column. Thus, N = 32, K = 8, m = 4, k1 = 2, and k2 = 4. The equations that are implemented to obtain the horizontal and vertical redundant bits are shown as follows:

Equations for generating horizontal bits

$$H_4H_3H_2H_1H_0 = D_{11}D_{10}D_9D_8 + D_3D_2D_1D_0$$
$$H_9H_8H_7H_6H_5 = D_{15}D_{14}D_{13}D_{12} + D_7D_6D_5D_4$$
$$H_{14}H_{13}H_{12}H_{11}H_{10} = D_{27}D_{26}D_{25}D_{24} + D_{19}D_{18}D_{17}D_{16}$$
$$H_{19}H_{18}H_{17}H_{16}H_{15} = D_{31}D_{30}D_{29}D_{28} + D_{23}D_{22}D_{21}D_{20}$$

Equations for generating vertical bits

V0	=	D0	xor	D16
V1	=	D1	xor	D17
V2	=	D2	xor	D18
V3	=	D3	xor	D19
V4	=	D4	xor	D20
V5	=	D5	xor	D21
V6	=	D6	xor	D22
V7	=	D7	xor	D23
V8	=	D8	xor	D24
V9	=	D9	xor	D25
V10	=	D10	xor	D26
V11	=	D11	xor	D27
V12	=	D12	xor	D28
V13	=	D13	xor	D29
V14	=	D14	xor	D30
V15	=	D15	xor	D31

The horizontal redundant bit H and the vertical redundant bit V are obtained from the encoder. The DMC encoded codeword consist of D, H and V. This code work is then stored in the memory. This arrangement of data and the redundant bit generation is described with the help of example as follows:

32-bit Data arranged in matrix

0	1	0	1	0	1	1	0	0	1	1	0	1	0	1	0
1	1	0	0	1	0	1	0	0	0	1	1	0	1	0	1

32-bit Data arranged in symbol matrix (2 x 4)

Symbol-3	Symbol-2	Symbol-1	Symbol-0
0101	0110	0110	1010
1100	1010	0011	0101
Symbol-7	Symbol-6	Symbol-5	Symbol-4

Generated Horizontal Redundancy Data:

Symbol 0 + 2 = 10000
Symbol 1 + 3 = 01011
Symbol 4 + 6 = 01111
Symbol 5 + 7 = 01111

Generated Vertical Redundancy Data:

1	0	0	1	1	1	0	0	0	1	0	1	1	1	1	1

If MCUs are introduced in the memory then the decoder can correct the errors. In the decoder the redundant bits are again generated from the data bits that were stored in memory. If the memory data is represented by D' then H' and V' represents the redundant bits generated by the decoder. In the decoding process, the stored redundant bits (H and V) and the generated redundant bits (H' and V') are used to generate the syndrome bits. These syndrome bits are further used to locate the position of error in the data that is fetched from memory and also to correct the error-bits. The architecture of the proposed decoder is shown in figure 5.

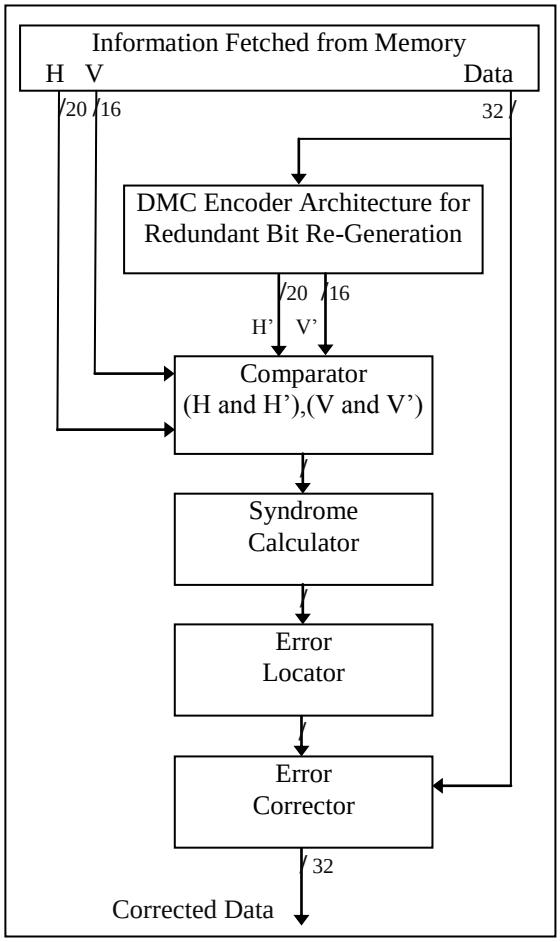


Figure5. Proposed Architecture of Fault tolerant Memory using Decimal Matrix Decoder

In the proposed work logical comparator using XOR gate is used to generate the syndrome. The conventional design of error syndrome generator logic utilizes decimal subtractor

logic to generate the syndrome data. The Syndrome generation in the proposed work is shown in fig. 6. The equations that are implemented to obtain the horizontal and vertical syndrome bits are shown as follows:

$$V_{syn} = V \text{ xor } V'$$
$$H_{syn} = H \text{ xor } H'$$

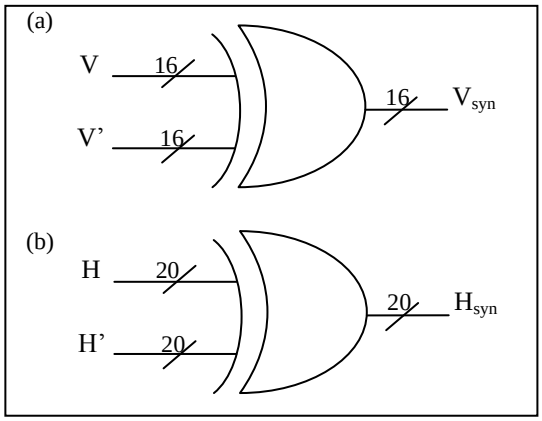


Figure6. Syndrome generation using proposed XOR-comparator (a) Vertical Syndrome, (b) Horizontal Syndrome

If any bit or bits of Hsys or Vsys is non-zero, it indicates the presence of the error in the stored data. To locate the symbol that contains error the horizontal and vertical syndrome bits are grouped with the same length as that of horizontal and vertical redundant bits. This grouping is depicted in fig. 7.

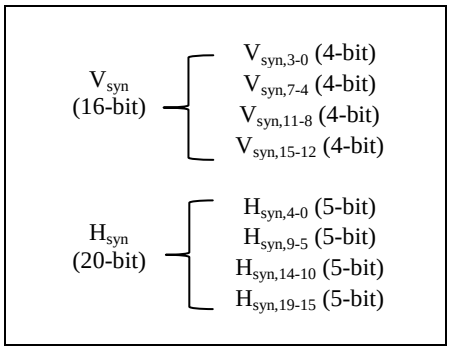


Figure7. Grouping of Syndrome Bit

The error location in the data matrix is identified by locating the common symbol in the data matrix from the non-zero syndrome horizontal and vertical symbols. This is depicted in fig. 8. Thus a non-zero value of syndrome group $H_{syn,4-0}$ and $V_{syn,3-0}$ indicates that there is error in stored data symbol D'_{3-0}. Likewise errors in other stored data symbols can also be located.

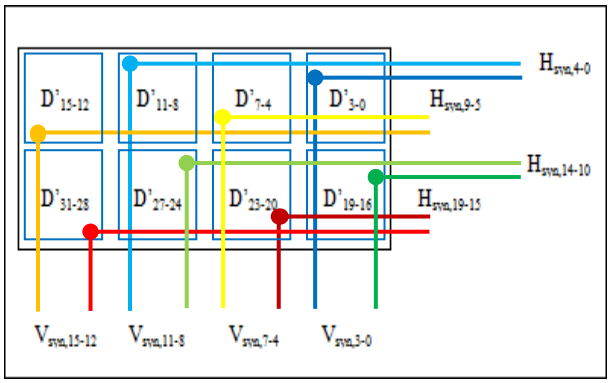


Figure8. Error Location using Syndrome Data

If the error syndrome is "zero" for a symbol then the memory retrieved symbol data is accepted as it is, else, once error is located then the error corrector logic corrects the error by performing the bit inversion from the error symbol. The bit inversion operation is performed using XOR operation which is depicted as follows.

Corrected Data		Memory Data		Syndrome Data
Symbol-0_out	=	Symbol-0	xor	$V_{syn,3-0}$
Symbol-1_out	=	Symbol-1	xor	$V_{syn,7-4}$
Symbol-2_out	=	Symbol-2	xor	$V_{syn,11-8}$
Symbol-3_out	=	Symbol-3	xor	$V_{syn,15-12}$
Symbol-4_out	=	Symbol-4	xor	$V_{syn,3-0}$
Symbol-5_out	=	Symbol-5	xor	$V_{syn,7-4}$
Symbol-6_out	=	Symbol-6	xor	$V_{syn,11-8}$
Symbol-7_out	=	Symbol-7	xor	$V_{syn,15-12}$

IV. Simulation And Synthesis Results

The Encoder and Decoder simulation waveforms are shown in Fig-9, Fig-10 and Fig-11 respectively. The present work is simulated using Xilinx. The RTL Schematic diagrams of Encoder and Decoder designs are shown in Fig-12(a) and Fig-12(b) respectively.

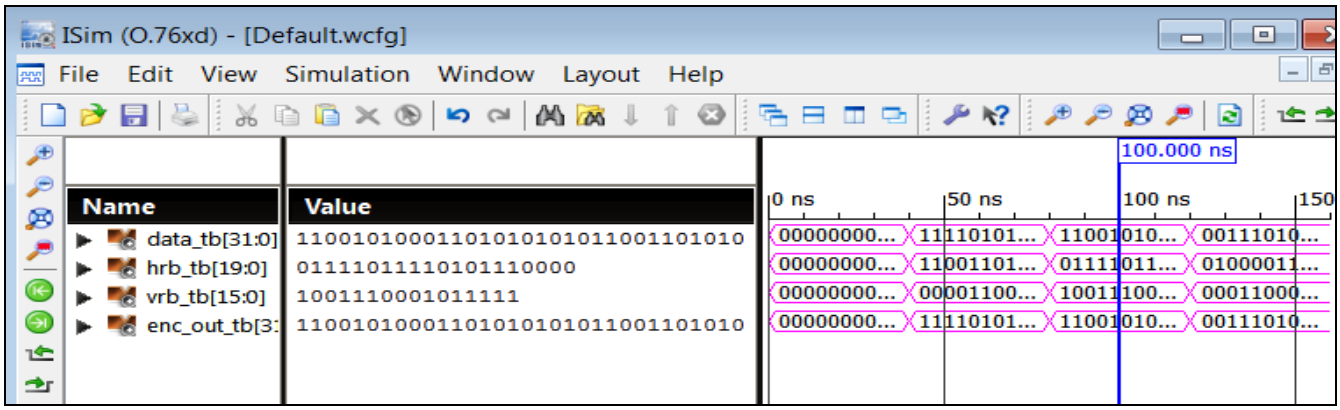


Figure9. Encoder Simulation Waveform (Example-1)

ISSN: 2321-8169
654 - 659

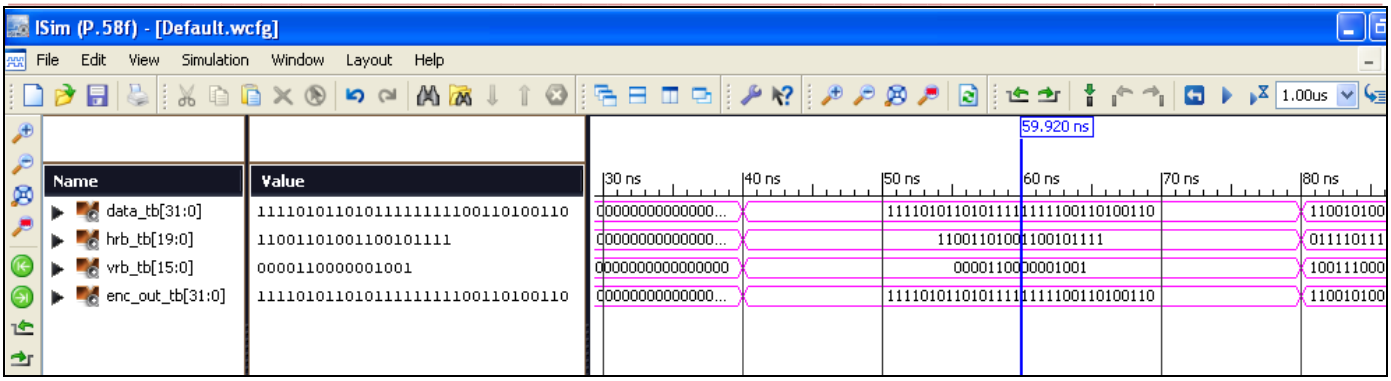


Figure10. Encoder Simulation Waveform (Example-2)

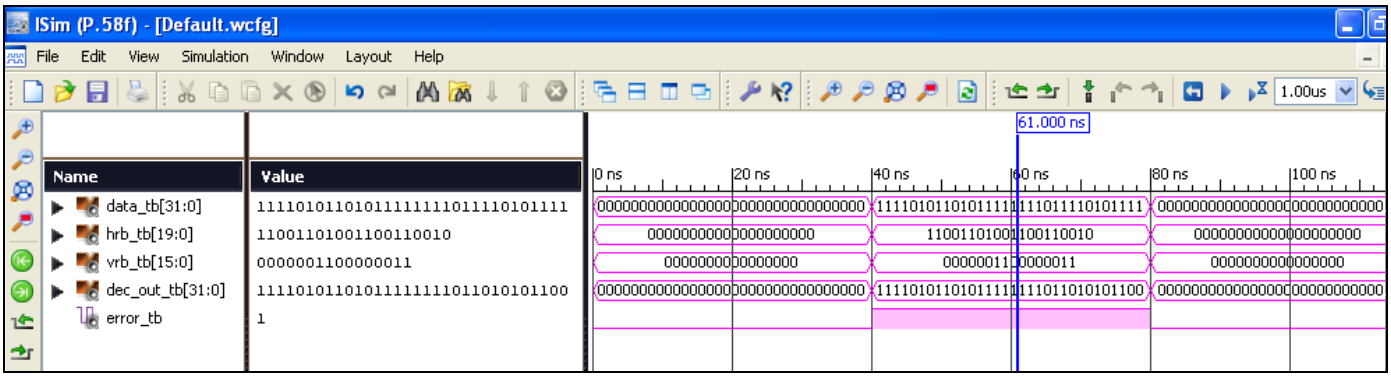
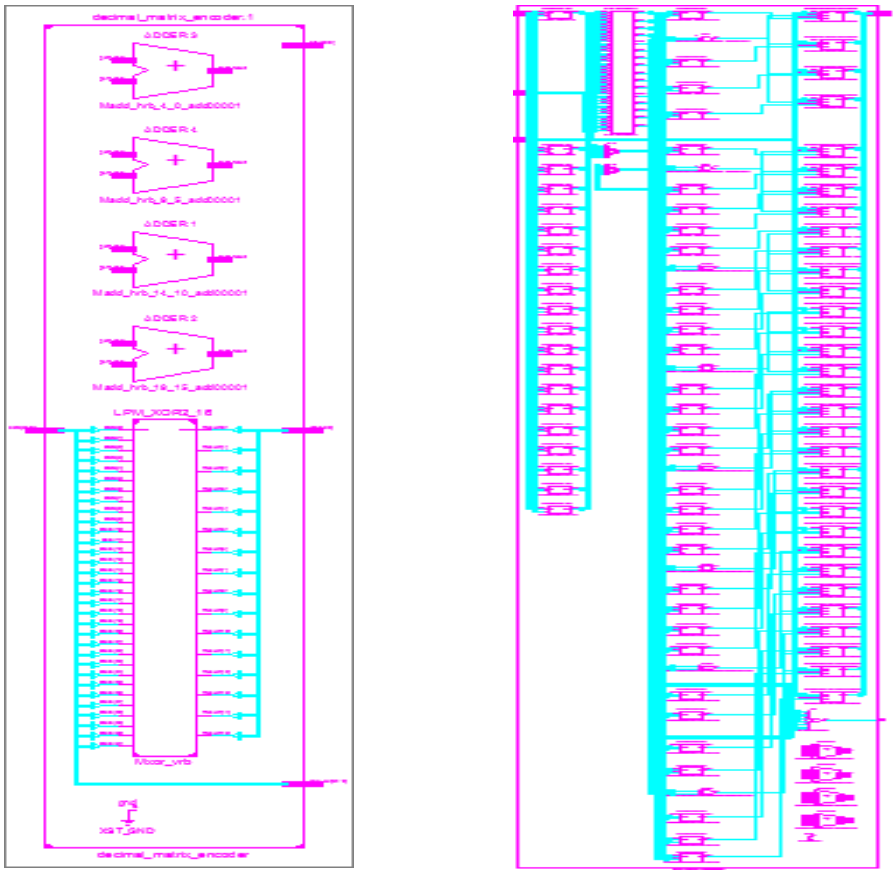


Figure11. Decoder Simulation Waveform (Example-1)



(a) (b)

Figure12. RTL Schematic of Proposed (a) DMC Encoder (b) DMC Decoder

The proposed design is simulated for communication hardware with a capability of burst error handling. The FPGA based hardware utilization summary of the proposed Encoder and Decoder designs is presented in Table-III and Table-IV respectively.

TABLE I
HARDWARE UTILIZATION SUMMARY OF 32-BIT DMC ENCODER

Spartan-3E XC3S500E-4PQ208	Total	32-bit DMC Encoder	
		Used	%
Slices	4656	26	0
LUTs 4-Inputs	9312	48	0
Bonded IOBs	158	100	63

TABLE II
HARDWARE UTILIZATION SUMMARY OF 32-BIT DMC DECODER

Spartan-3E XC3S500E-4PQ208	Total	32-bit DMC Decoder	
		Used	%
Slices	4656	70	0
LUTs 4-Inputs	9312	123	0
Bonded IOBs	158	101	63

V. CONCLUSION

In proposed DMC the presence of MCUs with modified syndrome calculation unit using XOR comparator is proposed. During the encoding (write) process, information bits are fed to the encoder, and then the horizontal redundant bits *H* and vertical redundant bits *V* are obtained. The decoding process is required to obtain the corrected word from cell upset. The area of DMC is minimized by reducing its decoder block. The proposed work replaces decimal subtractor by simple logic XOR based comparator and reduces the area overhead of the decoder.

ACKNOWLEDGMENT

The authors thank Mr. Piyush Jain (Director, Innovative Technology Design and Training Center, Bhopal India) for sharing his ideas in writing this paper.

REFERENCES

[1] Jing Guo, Liyi Xiao, Zhigang Mao and Qiang Zhao, "Enhanced Memory Reliability Against Multiple Cell Upsets Using Decimal Matrix Code", IEEE Transactions on Very Large Scale Integration Systems, Vol.22 No.1, pp.127-135, January 2014.

[2] David S, and Gayathree K., "A Comparative Study of Various Error Correction Codes", International Journal of Computer Science and Mobile Computing (IJCSMC), Vol.3 Issue 8, pp.196-200, August 2014.

[3] S. Kamalakannan, S. Karthikeyan and K. Sathyamoorthy, "Implementation of Error Correction Technique based on Decimal Matrix Code", International Journal of Advanced Research Trends in Engineering and Technology (IJARTET), Vol.2 Issue 4, pp.1-6, April 2015.

[4] K. Madhuri and V. Thrimurthulu, "Implementation of Decimal Matrix Code for correcting Cell Upsets in Static Random Access Memories", International Journal of Electrical, Electronics and Data Communication (IJEEDC), Vol.2 Issue 10, pp.72-76, October 2014.

[5] Sanilkumar N. S. and Aby Thomas, "Efficient Error Correcting Mechanism for Memories used in Radiated Environment", International Journal of Engineering Research and General Science (IJERGS), Vol.3 Issue 5, pp. 624-631, September-October 2015.

[6] Kavitha A. and Ramasathu L., "Fault Secure Memory using Modified Decimal Matrix Code", International Journal of Computer Science and Engineering Communications (IJCSEC), Vol.3 Issue 2, pp.829-834, 2015.

[7] Tintu B. Varghese and Ambika Sekhar, "Implementation of Decimal Matrix Code for Enhancing Memory Reliability", SSRG International Journal of Communication and Media Science (SSRJ-IJCMS), Vol.2 Issue 4, pp.1-5, July-August 2015.

[8] Neethu V. and Anju S. L., "A New Methodology for Error Detection and Correction to Realize Fault Tolerant Memory", International Journal of Science and Research (IJSR), pp.1689-1694, 2013.

[9] K. Janani and R. Ponni, "FPGA based protection of Soft Errors using Multiple Error Correction Codes", IJESC, January 2015.

[10] Maria Antony S. and Sunitha K., "Hybrid Matrix Codes for Enhanced Memory Reliability against Multiple Cell Upsets", International Journal for Scientific Research and Development (IJSRD), Vol.3 Issue 1, pp.114-117, January 2015.

[11] Sandeep M. D. and Rajashekhargouda C. Patil, "An approach to Reduce Number of Redundant Bits used to Overcome Cell Upsets in Memory using Decimal Matrix Code", ACEEE International Conference on Recent Trends in Signal Processing, Image Processing and VLSI (ICRTSIV), 2014.

[12] M. Sivasankaran and G. Renganayaki, "Improving Memory Reliability against Multiple Cell Upsets using Hamming based Matrix Code", International Journal of Innovative Science and Applied Engineering Research (IJISAER), Vol.13 Issue 44 Ver. 1, pp.38-43, March 2015.

[13] V. Vithya and P. Sakthivel, "Reducing Redundant Bits and Enhanced Memory Reliability using Decimal Matrix Code", International Journal of Electrical and Electronics Research (IJEER), Vol.3 Issue 3, pp.48-54, July-September 2015.

[14] E. Abinaya, D. Somasundareswari and S. Mathan Prasad, "Implementation of a Decimal Matrix Code for Correcting Multiple Cell Upsets in SRAM based FPGA Memories", International Journal of Advanced Research Trends in Engineering Technology (IJARTET), Vol.2 Special Issue 8, pp.87-93, 2015.

[15] Gayathree K. David S. and Moortheeswari M., "Reliability Enhancement using Parity Algorithm", International Journal of Innovative Science, Engineering and Technology (IJISET), Vol.2 Issue 4, pp.1176-1180, April 2015.

[16] Tarun Srivastava and Vipin Gupta, "Improved Performance of Memory Reliability against Multiple Cell Upsets using Hybrid Matrix Code", IJSET, 2015.

[17] E. Lavanya and J. Dhanapathi, "Detection of Multiple Cell Upsets in Memory for Enhanced Protection", International Journal of Innovative Trends and Emerging Technologies (IJITET), Vol.1 Special Issue 2, pp.129-132, March 2015.

[18] Gayathree K. and David S., "Analysing the Power Overhead of Decimal Matrix Code with Different Adder Architecture", International Journal of Research in Engineering and Advanced Technology (IJREAT), Vol.3 Issue 2, pp.126-132, April-May 2015.