Design of High Performance EDAC Unit Using Decimal Matrix Code for 2-Way Set Associative Cache

L. Prathyusha Reddy, Hemanjaly Ojha, Rohit Kumar Singh

Abstract—Soft errors due to radiation are becoming the major concern sometimes leading to an irrepairable damage. Traditional methods to reduce these errors dealt with mitigation of SEU but as a consequence of decrease in node capacitance MBUs can occur. Some fault tolerant techniques based on matrix codes by combining various techniques together have been proposed which are complex involving much computational delay with effective correction of 2 to 3 faults. So when these types of codes are used for improving the reliability of memories such as cache memory, delay involved in accessing the data increases which is undesirable. This project is focused on implementation of decimal algorithm based EDAC unit involving lower computational delay overhead which can offer better fault tolerance and its application to an SRAM memory and 2-way Set Associative Cache memory of LEON-3FT processor prototype which used SEU protection based EDAC Unit to show that it can offer better fault tolerance. It is compared to other EDAC techniques to clearly show the improvement of its fault tolerance and reduced computational delay overhead. The analysis is done for 32 bit data and is observed that the proposed technique can detect 32 errors and can efficiently correct up to 8 adjacent errors and 5 random errors. Two dimensional error correcting code can detect up to 8 errors and can correct up to 4 adjacent errors and 3 random errors, SEC-DED code can detect up to 2 errors and Hamming can correct only 1 error code can detect and correct 1 error only involving 4 addition or subtraction operations and 32 EX-OR operations, 80 EX-OR operations, 257 EX-OR operations and 187 EX-OR operations with total delays 15.115ns, 16.116ns, 18.848ns and 19.289ns respectively.

Index Terms— Sotf errors, MBU, EDAC, Decimal Matrix Code, fault tolerance, computational delay, 2-Way set associative cache.

IJSER

1 INTRODUCTION

The effects due to radiation are of very important concern because ignorance of these can result in serious problems like incorrect results or sometimes the entire device failure causing the huge losses. Soft errors can be of many types like SEU, MBU, SEL, SET, and TID depending upon their effect on various components.

Modern electronics is fuelled by Moore's law and the electronic components are scaled to nanometers with the aim of reducing the area and power requirements and thus improving the performance as large number and complex circuits can be placed on a single die. The challenges related to the radiation effects on electronics were presented in [2].

But such scaling has resulted in the increase in Soft Error Rate and it has been reported over four technology generations (0.35µm, 0.25 µm, 0.18 µm, 0.13 µm) in [1]. The radiation effect has resulted in the adverse effects and two problems related to the same reported in [3]. They are:

1. After three years of successful operations communications between European Space Agency and Hipparcos astronomy satellite were terminated in August 1993.

2. Japan Engineering test satellite ETS-6 did not reach its original GEO orbit due to apogee kick motor problem.

Therefore the failures related to radiation are of much concern especially in memories they result in single event and multiple event upsets and hence results in incurrent data access. There are many fault tolerant techniques available to mitigate the risks due to radiation [13].

These include physical techniques like shielding, triple modular redundancy, silicon on insulator are used but they cannot effectively correct the errors because the some of the particles can easily penetrate through such structures. [13]

Many logical techniques are also available in which a software based code is written and the blocks are developed so as to correct the errors introduced into the memories. Interleaving is also one of the methods where correction is achieved by dividing the word with physical defects into different words but it is not practically possible [13].

The other logical techniques can be used to achieve protection for such kind of memories using various software based error detection and correction codes some of them can detect and correct single error whereas some of them can detect and correct multiple errors and thus can provide superior protection to memories.

As a part of this project various EDAC techniques are compared and a superior technique in terms of fault tolerance and

less computational delay is found and is implemented for two types memory, a simple SRAM or main memory and a LEON-3FT prototype based high speed 2-way set associative cache memory to prove that the proposed fault tolerant memory based on an efficient fault tolerant technique can provide superior protection thus achieving a higher reliability.

2 BACKGROUND AND RELATED WORK

Various EDAC techniques include Hamming code [17] and SEC-DED code which can be used to correct the single error. These are very simple techniques and are employed in memories for SEU detection and correction. But due to technology scaling MBU's are increasing due to penetration of errors. These can be either adjacent or random errors and detailed studies on the same are presented in [18].

For MBU detection and correction various techniques are employed which includes BICS based codes described in [4] in which Built In Current Sensors are placed in columns of each memory block and other codes in the rows to detect and correct the errors depending upon the type of code used in the rows of memory block but the BICS are prone to radiation effects and hence using this for radiation hardening is not a best option.

Matrix based codes are based upon the technique of dividing the given N bit codeword into matrix format consisting of l rows and m columns i.e. N = (l x m). There are many codes based on the same technique combining different codes together to combine the advantages of both the techniques so as to achieve the better performance.

Hamming + Parity: It includes data arranged in matrix format with hamming bits in rows with parity bits in columns which can be able to detect and correct 2 bits in all the cases. [8] Another code similar to this involves SEC-DED code in the rows with parity bits in rows which can be used for MBU detection and correction. [9]

Matrix based codes for Adjacent Error Detection: It is based on L-distance parity codes in rows and columns of the matrix which can correct up to L-1 consecutive errors or adjacent errors. [5]

Two dimensional codes: It is concerned with the arrangement of given N bit data into l rows and m columns in such a way to improve the reliability and reduce the area and delay involved by reducing the number of redundant bits. [6]

In this parity codes are applied in the columns after arranging data in the format specified in above table and in the rows either hamming or SEC-DED or L-distance parity codes which can correct faults up to L-1 per row are applied and MED bits are used to detect multiple errors and assert ME signal in case of multiple error and parity bits in columns are used to correct these errors [6].

The computational complexity of this code is high as it performs XOR operation on the binary data and reliability also decreases as the bits whose XOR operation results in the parity bits are if changed, it results in the same parity bit. Hence in such cases the error cannot be detected and corrected.

The basics of memory system, its hierarchy and different types of mapping techniques is described in [11] from which it can be known that the set-associative cache memory embeds advantages of both direct mapped and fully associative cache which is most widely employed technique in the recent processors like ARM Series and LEON-3FT etc.

LEON 3-FT processor is a fault tolerant processor which uses a SEC-DED based EDAC Unit for providing protection to Cache memory, FPU and Other memory interface [12].

The locality principle based on which the next referenced location in memory can be decided is given in [14] which explains the two main principles temporal and special localities.

Verilog, a simple hardware description language importance, its syntax details which is used for coding various hardware blocks of the project were clearly described in [10],[15], Xilinx ISE design Suite 14.5 tool installation and other details are presented in [16].

The remainder of the paper is organised as follows. Section 3 provides the explanation of Decimal Matrix Code technique based EDAC Unit used for improving fault tolerance of memories. In section 4 the proposed fault tolerant memories using DMC are presented and experimental analysis of the DMC code by comparing it with other EDAC technique to emphasize its superiority in terms of fault tolerance and performance thus justify it as a very good technique to be applied to improve memory reliability design summary of implemented blocks is provided in section 5 and final conclusion is given in section 6.

3 DECIMAL MATRIX CODE BASED EDAC TECHNIQUE

3.1 Decimal Matrix Code

This code is proposed to enhance capability to correct the MBUs and reduce delay involved in the calculations. In DMC based EDAC technique decimal integer operation is adopted [7].

If binary operation is used the delay involved in bit by bit computation is more, also correction capability is limited. Hence this problem can be solved using DMC code based EDAC technique which uses decimal algorithm. In this code the data is logically arranged matrix format which can be clearly understood from the TABLE 1.

TABLE 1

Arrangement of data in DMC based EDAC technique (32 bit data)

Data bits (D15...D0)	Horizontal Parity (H9...H0)
Data bits(D31...D15)	Horizontal Parity (H19...H10)
Vertical Parity (V15...V0)	

3.2 DMC Encoder:

Encoder is used to generate the Horizontal and vertical parity bits for the data that is to be written to a particular memory location.

Given N bit data is divided into l symbols of m bits and then the l symbols are arranged in two dimensional matrix format logically.

Horizontal parity bits can be computed using the decimal integer addition of the symbols following a specific format shown below.

(H4...H0) = (D3...D0) + (D11...D8) (1)

(H9...H5)=(D3...D0)+ (D11...D8) and so on. (2)

Vertical parity bits are computed using bit by bit XOR operation as shown below.

Vi = Di ^ Di+16 (3)

While dividing the symbols and arranging them focus should be on the correction capability i.e. that particular arrangement should have superior correction capability.

3.3 DMC Decoder:

Decoder is used to detect and correct the single and multiple bit upsets present in the memory block. The Decimal matrix code decoder consists of 4 individual blocks.

Encoder: It is the same encoder used to encode the original data bits which is reused to compute the parity bits for the data read from the memory as shown below.

(H4′...H0′) = (D3′...D0′) + (D11′...D8′) (4)

(H9′...H5′) = (D3′...D0′) + (D11′...D8′) and so on. (5)

Vi′ = Di′ ^ D′(i+16) (6)

Syndrome Calculator: The inputs to this are horizontal and vertical parity bits of actual data and the data read from the memory that can have faults in it. Here Horizontal syndrome bits are calculated by using the decimal integer subtraction and vertical syndrome bits using XOR operation. Outputs of this block are horizontal and vertical syndrome bits.

(HS4′...HS0′) = (H4′...H0′) - (H4...H0) and so on (7)

(Vsi) = (Vi′ – Vi) (8)

Error Locator: Syndrome bits generated from the Syndrome calculator are inputs to the error locator. Here first Horizontal syndrome bits for the corresponding are checked to identify if any of them are non zero. If any of them is non zero then vertical syndromes corresponding to that particular data are checked. By this data with error can be located.

Error Corrector: Data with error is located using the error locator block and this will be input to the error corrector block and the data with error can be corrected as shown below.

Dicorrect = Di ^ Si (9)

4 PROPOSED FAULT TOLERANT MEMORY USING DMC CODE

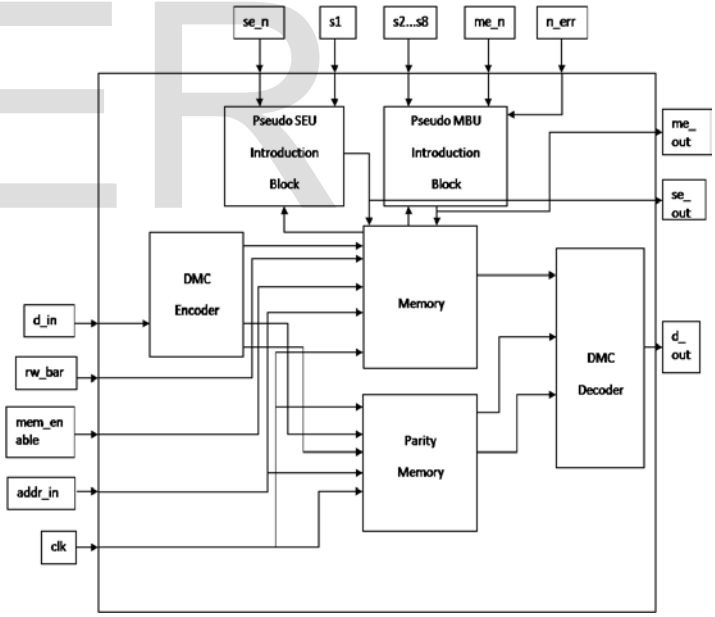

Fig1: Block diagram of proposed Rad-Hard Memory using Decimal Matrix Code based EDAC Unit

The proposed Rad-hard memory using DMC code can provide superior protection to memory with better performance compared to other techniques. For fault injection Pseudo SEU and MBU introduction blocks are employed in contrast with the previous methods where radiation environment is created which is very expensive and need specialised environment. Moreover the data after correction is written back to the memory to avoid accumulation of errors.

IJSER

4.1 Main Memory:

The main memory consists of 32 data lines and 12 address lines, a rw-bar signal and a positive edge clock signal, Memory enable signal. Any operation can be done on the memory only if the Memory enable signal is held high. Two operations read and write can be performed on this memory depending upon the rw-bar signal.

4.2 2-Way Set Associative Cache Memory:

The cache memory is 2-way set associative of LEON- 3FT Prototype. The cache memory is divided into sets and each set has two blocks of cache. It has 32 data lines, 8 tag bits and 4 set bits, a rw-bar signal, a cache-enable signal, a way-select signal and a positive edge clock signal. Any operation can be done only if cache-enable signal is high, tag and set bits help in finding the block of cache in which required data is present in the cache memory.

If rw-bar signal is held high then way-select signal is checked to decide to which way data is to be written, then data and tag are written to the particular location pointed by the set bits.

If it is low, valid bit is checked, if it is high tag is compared with the corresponding data in tag memory and then data should be read from the way which has the same tag requested and hit signal is set to high. If valid data is not present then a miss occurs and the miss signal is set to high.

4.3 Encoder and Decoder:

Encoder and decoder blocks are based on Decimal Matrix Code help in detecting and correcting the faults introduced into the memory.

4.4 Pseudo SEU Introduction Block:

The inputs to this block are the 32 bit data and a 5 bit select line. 5 bit select line is used to select one of the bits from the 32 bit data and then invert the selected bit. Using this SEU can be introduced into the memory.

4.5 Pseudo MBU Introduction Block:

This block is used to introduce multiple bit faults into the memory. The inputs to this block are the 32 bit data, a 4 bit signal and 8 5 bit select lines. Number of errors to be introduced is indicated by using the 4 bit signal and depending upon the number of errors to be introduced the select lines are used to introduce multiple bit errors into the memory.

4.6 Operation of the Fault Tolerant Memory:

All the blocks should be connected as shown in the Fig1.

Write cycle: If the memory-enable is high, rw-bar signal should be high and data is first given to the DMC encoder block which generates horizontal and vertical parity bits and these should be stored in memory along with the original data.

Read cycle: rw-bar signal should be low, the data from memory, horizontal and vertical parity bits are given as inputs to the DMC decoder block and the corrected data is read out.

To check the fault tolerance the fault injection is done using pseudo SEU and MBU introduction blocks which can be activated using s-en and m-en signals and then the output from these is written back to the memory, when this data is read, it is corrected by decoder block and thus corrected data can be obtained which should be written back to the memory.

Similar operation is valid in case of cache memory but in this both tag and data has to be written to the memory and depending on the way-select signal, it should be written to the specific way and while reading data tags of two ways are compared and then valid bit is checked to identify whether the valid data is present in that particular block and then depending on that hit or miss signals are generated. While introducing errors valid bit is checked to see if the valid data is present and then the data is extracted from that particular block and then error is introduced.

5 EXPERIMENT AND ANALYSIS

First the encoder and decoder of the Decimal Matrix Code are implemented for 32 bit data, an SRAM Memory which is considered as a main memory and a 2-way set associative cache memory LEON-3FT processor prototype along with the controller were implemented in Verilog and then the EDAC unit is applied to the main memory and 2-way set associative cache memory.

Pseudo SEU and MBU introduction blocks are implemented and then fault injection is done using these blocks. Various types of faults are introduced for testing the fault tolerance.

Other codes Hamming, SEC-DED and Two-dimensional matrix codes were implemented for the memories and are compared with the Decimal Matrix Code in terms of correction coverage, computational complexity, delay and area overheads and the verified the functionality of each block using Xilinx ISE Design suite 14.5.

The results are analysed and the various aspects like area overhead, delay, error detection and correction coverage, computational complexities are compared using tables and graphs as given below to prove the superiority of DMC code and then it is applied to an SRAM memory and cache memory and then analysed the number of slices occupied by the design on XC6SLX45 device of Spartan-6 FPGA family.

IJSER

TABLE 2

Comparison of Computational complexities (32-bit data)

EDAC Technique	Encoder	Decoder	
		Detection	Correction
DMC	4 addi-tions 16 XOR	4 subtractions	32 XOR
Two di-mensional	32 XOR	16 XOR	32 XOR
SEC-DED	128 XOR	128 XOR	1 XOR
Hamming	93 XOR	93 XOR	1 XOR

It can be seen from the TABLE 2 that the DMC requires lesser number of computations when compared to other techniques hamming, two dimensional and SEC-DED codes and hence when applied to faster memories it can offer better performance at the same time improving the fault tolerance.

The correction coverage of various techniques can be compared from the graphs given below.

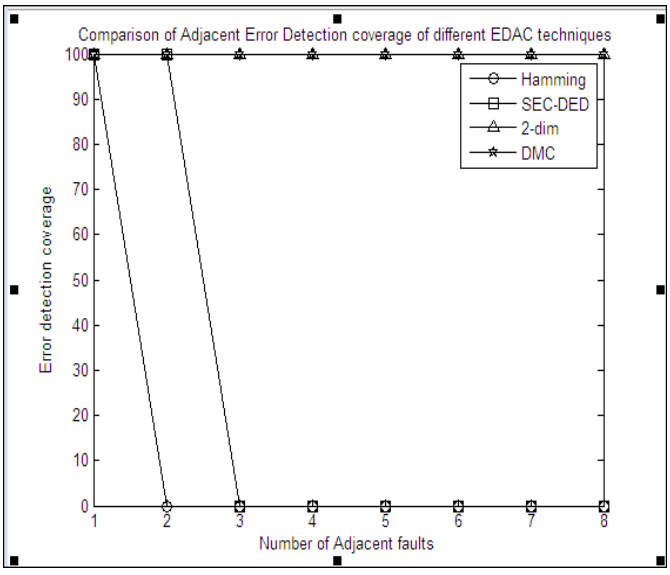

Fig. 2: Comparison of Adjacent Error detection coverage

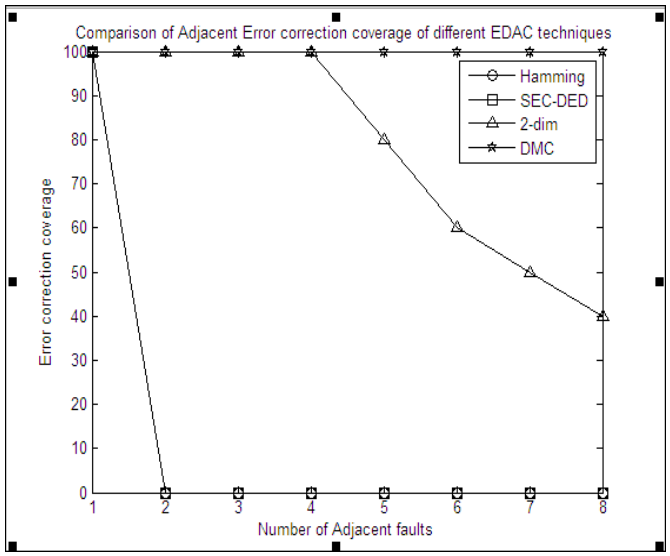

Fig. 3: Comparison of Adjacent Error correction coverage

From Fig. 2 and Fig. 3 it can be justified DMC has relatively same detection coverage as 2-dimensional code provided L=8 is chosen. DMC is excellent in case of adjacent error detection and can provide excellent correction coverage when compared to that of other techniques.

It can be justified that in case of random error detection and correction DMC offers excellent coverage compared to any other techniques. This can be observed from the graphs below.

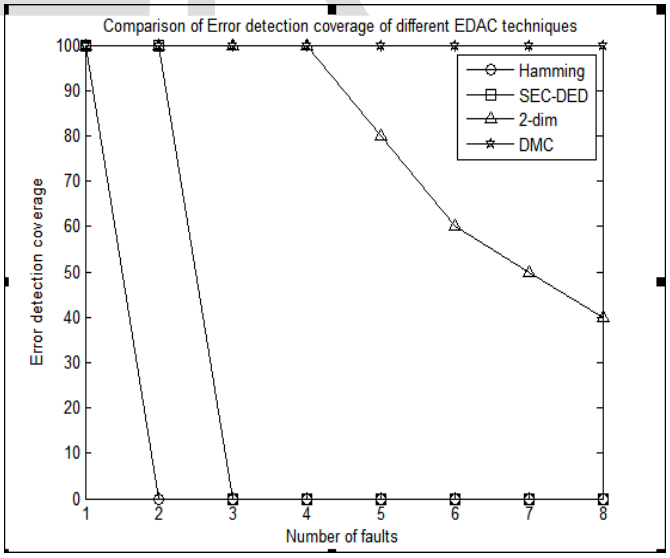

Fig. 4: Comparison of Random Error detection coverage

IJSER

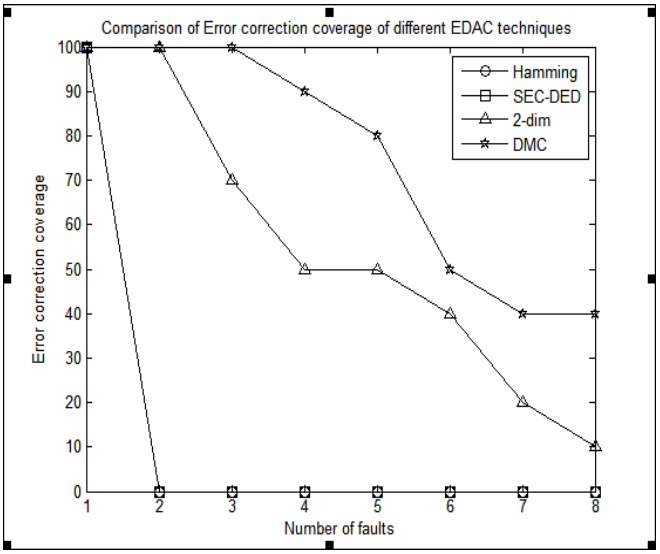

Fig. 5: Comparison of Random Error Correction coverage

It can be observed that DMC code is superior to all other techniques as it can detect all the 8 errors introduced and can correct efficiently up to 5 errors in case of random errors.This is analyzed by fault injection using the Pseudo SEU and MBU introduction blocks and then verifying by simulation using ISIM simulator of Xilinx ISE design suite 14.5 version and the graphs are obtained using MATLAB R2013a.

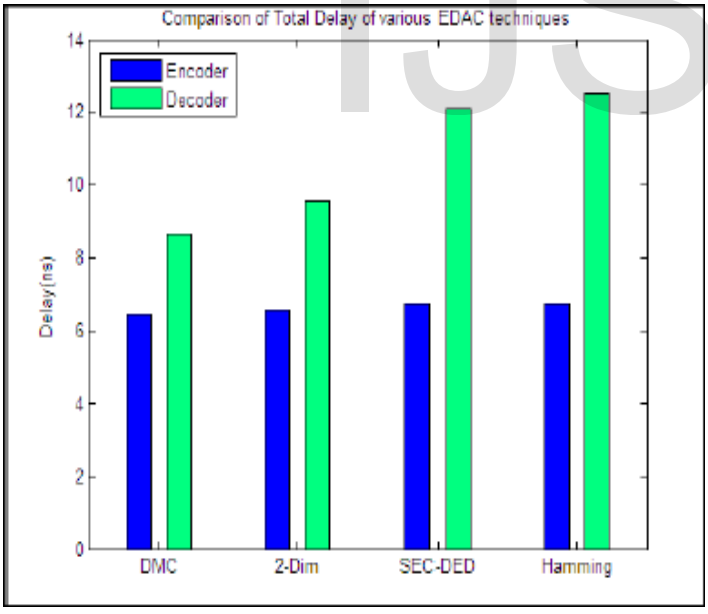

Fig.6: Total Delay of Encoder and Decoder (in nanoseconds)

The total delays and area overhead due to redundant bits used is calculated and the results are shown in Fig.6. and Fig.7.

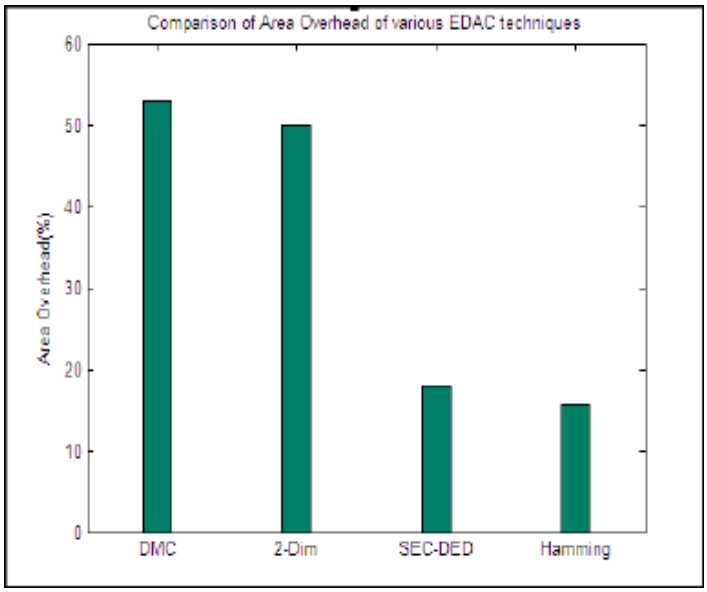

Fig.7: Area overhead due to redundant bits (%)

So it can be observed that the total delay involved in DMC code is lesser when compared to Hamming and SEC-DED and 2-dimensional code however it can be achieved through slight modification of the logic as it has already been shown that computational complexity of DMC is very less compared to other codes.

The area overhead of DMC is more than the other EDAC techniques due to more number of redundant bits which is the only disadvantage of using this technique.

The Design summary of all the implemented blocks can be shown from the TABLE 3. It shows the percentage utilization various blocks on XC6SLX45 device of Spartan-6 FPGA family. TABLE 4 and TABLE 5 shows the timing analysis of the blocks implemented in which various delays in each of the blocks are clearly shown.

TABLE 3

Device Utilization Summary (% utilization)

Implemented Block	#(slice reg)	#(slice LUTs)	#(occupied slices)	#(LUT FF's)	#(slice LUTs)
DMC Encoder	0	1	1	100	45
DMC Decoder	0	1	1	100	100

SEU Block	0	1	1	100	31
MBU Block	0	1	2	100	49
Main Memory	0	0	0	0	35
Rad-Hard Main Memory	1	19	21	98	86
Cache Memory	1	1	1	62	37
Rad-Hard Cache Memory	1	8	11	94	88
Address Seperator	0	0	0	0	11
Rad-Hard Cache Controller	1	9	10	94	88

TABLE 4

Timing Analysis of implemented blocks

Implemented Block	Maximum combinational path delay (ns)
DMC Encoder	6.445
DMC Decoder	8.670
SEU Block	6.211
MBU Block	16.169
Address separator	4.372

TABLE 5

Timing Analysis of implemented blocks

Implemented Block	Min period	Min time before clock (ns)	Max time after clock (ns)
Main Memory	No path	2.813	5.000
Rad-Hard Main Memory	3.750	18.642	6.971
2-Way Set Associative Cache	4.364	6.855	4.453
Rad-Hard Cache Memory	5.742	10.758	7.794
Rad-Hard Cache with Controller	5.711	10.758	7.794

IJSER

6 CONCLUSION

On the basis of experimental results it can be concluded that Decimal Matrix Code is superior to all the other existing techniques in terms of correction coverage and computational complexity. It is analysed for 32 bit data and is observed that it can detect 32 errors and can correct up to 8 adjacent and 5 random errors efficiently, and can also correct more number of errors depending on the type of errors introduced. Number of computations required for detection and correction process are also very less and hence the delay involved in computation is decreased. Though the correction capability of this technique in case of random errors is observed to be decreasing after 5 errors, it is observed from experimental studies that adjacent errors are of much concern, occurrence of random errors is at most 3 to 4 and hence this technique is best in terms of fault tolerance. Its application to A LEON 3-FT Processor prototype 2-way Set Associative cache memory provides superior protection with optimized performance. After correcting, data is written back to memory which helps to avoid accumulation of errors i.e. memory contain the corrected data but not the erroneous data. The only disadvantage over here is the area overhead due to use of more number of redundant bits so to decrease the area overhead without affecting fault tolerance and reducing the delay involved in

writing back the corrected data into memory will be the future scope of the project.

REFERENCES

[1] S. Hareland, J. Maiz, M. Alavi, K. Mistry, S. Walsta, and C. Dai, "Impact of CMOS process scaling and SO1 on the soft error rates of logic processes", Symposium on VLSI Technology Digest of Technical Papers, June, 2001, pp. 73-74.

[2] P. E. Dodd, M. R. Shaneyfelt, J. R. Schwank, and J. A. Felix, "Current and Future Challenges in Radiation Effects on CMOS Electronics", IEEE Trans. Nucl. Sci., vol. 57, no. 4, pp.1747-1763, Aug., 2010.

[3] A.Y. Ozyildirim, and O. Ergun, "Comparison Of Radiation Analyses", in Proc. 6th Int. Conf. Recent Advances in Space Technologies, June, 2013, pp. 705-710.

[4] P.Reviriego, and J. A.Maestro, "Efficient error detection codes for multiple-bit upset correction in SRAMs with BICS", ACM Trans. Design Automation of Electronic Systems, vol. 14, no. 1, article 18, Jan., 2009.

[5] C. Argyrides, P. Reviriego, D. K. Pradhan, and J.A. Maestro, "Matrix-Based Codes for Adjacent Error Correction", IEEE Trans. Nucl. Sci., vol. 57, no. 4, pp. 2106-2111, Aug. 2010.

[6] Ming Zhu, Liyi Xiao, Shuhao Li, Yanjing Zhang, "Efficient Two-Dimensional Error Codes for Multiple Bit Upsets Mitigation in Memory",in 2010 IEEE 25th International Symposium on Defect and Fault Tolerance in VLSI Systems., Oct., 2010, pp.129-135.

[7] Jing Guo, Liyi Xiao, Zhigang Mao, and Qiang Zhao, "Enhanced Memory Reliability Against Multiple Cell Upsets Using Decimal Matrix Code", IEEE Trans. VLSI Systems, vol. 22, no. 1, pp.127-135, Jan. 2014.

[8] C.Argyrides, H. R.Zarandi, and D. K.Pradhan, "Multiple Upsets Tolerance in SRAM Memory",in IEEE International Symposium on Circuits and Systems, May, 2007, pp.365-368.

[9] C.Argirides, D. K. Pradhan, and T.Kocak, "Matrix Codes for Reliable and Cost Efficient Memory Chips", IEEE Trans. VLSI Systems, vol.19, no. 3, Mar. 2011.

[10] Samir Palnitkar, Verilog HDL: A Guide to Digital Design and Synthesis, 2nd ed, California, USA, Prentice Hall PTR, 2003.

[11] Mostafa Abd-El-Barr, Hesham El-Rewini, Fundamentals Of Computer Organisation and Architecture, John Wiley & Sons, Inc. publication, Hoboken, New Jersey,2005.

[12] LEON3-FT SPARC V8 Processor LEON3FT-RTAX Data Sheet and User's Manual, Aeroflex Glasier,v 1.9, Jan., 2013.

[13] Fa-Xin Yu.et.al., "Overview of Radiation Hardening Techniques for IC Design". Information Technology Journal, pp:1068-1080, June 10, 2010.

[14] P.J.Denning,"The Locality Principle", Communications Of The ACM ,vol.48, no. 7,pp.19-24,July., 2005.

[15] J.Bhasker, Verilog Hdl Synthesis:A Practical Primer, Star Galaxy Publishing, Allentown, PA, 1998.

[16] ISE Design Suite 14: Release notes, Installation, and Licensing, v.14.5, March 20, 2013.

[17] Fazal Noorbasha.et.al,"VLSI Implementation of Encryption and Decryption System Using Hamming Code Algorithm", Int.Journal of Engineering Research and Applications, vol. 4, no. 4, pp.52-55, April., 2014.

[18] P.Reverigo, J.A.Maestro, "Study of the Effects of Multibit Error Correction Codes on the Reliability of Memories in the Presence of MBU", IEEE Trans. Device And Materials Reliability, vol. 9, no. 1, Mar., 2009.

AUTHORS

First Author – L. Prathyusha Reddy, Pursuing M -Tech in Computational Engineering from Rajiv Gandhi University Of Knowledge Technologies -BASAR.

prathyusha.lakkadi@gmail.com

Second Author – Hemanjaly Ojha, Lecturer in Department of ECE, Rajiv Gandhi University Of Knowledge Technologies - BASAR.

ohemanjali@gmail.com

Third Author – Rohit Kumar Singh, Pursuing M -Tech in Computational Engineering from Rajiv Gandhi University Of Knowledge Technologies -BASAR.

Thakursingh1991@gmail.com

IJSER