Design and Implementation of DMC for Memory Reliability Enhancement

Hema [1], Rekha S [2], Dr. A.M Bhavikatti[3]

[1] *Fourth SEM M. Tech Appa Institute of Engineering and Technology Kalaburagi, Karnataka, India*
[2] *Assoc. Professor Appa Institutes of Engineering and Technology Kalaburagi, Karnataka, India*
[3] *HOD & Professor BKIT Bhalki, Karnataka, India*

Abstract - *Demand for high accurate detection and correction of data in memory brings motivation for this work. In this modern era flow of data has increased to a higher extent which is creating the corruption of required data. Errors created by self-generator machines have less control from the human beings and also it's very difficult to monitor such a huge data. If one could achieve high accurate data correction then a capacity of memory integration on chip can be increased to fill the gap between the technology and semiconductor industry. The scope for DMC exists because of its less area and more accurate error detection. Along with that scope exists for ERT because it determines the area without actually encoding and decoding*

***Key Words*: DMC (decimal matrix code), ERT (error reuse technique), Multiple Bit Upsets (MCUs), Error Correction Code (ECC), Punctured Difference Set (PDS).**

1. INTRODUCTION

The CMOS technology has scaled down to nano scale which is making more transistors on a chip. As the number of electronic components increased and error rate in memories also increased due to ionizing effects from atmosphere. In memory single bit error is a major problem and Multiple Bit Upsets (MCUs) has turned into a bigger problem than the single bit error.Was used for over the year to correct the single bit errors in the memories. To manage MCUs Reed – Solomon codes, Punctured Difference Set (PDS) and Bose-Chaudri-Hocquenghem codes have been used till now. In these converted codes encoding and decoding are more complex and requires more power, more area and large delay. The technique of interleaving has been used in MCUs to improve memory cells in physical arrangement on the boards. In Content Addressable Memory (CAM) interleaving cannot be used because of tight coupling in between the cells.

To assure the MCUs, with respect to single-error rectification and two fold locations the Built-In Current Sensors are used. To effectively rectify the MCUs error the new method, Two Dimensional Matrix Code (MCs) is used, which divides the long data into the various sub data's. In Hamming codes a bit to every line is secured and in equality code every segment is secured. After identifying these two errors in Hamming codes the vertical syndrome in MC have capacity of correcting errors up to 2bits. In a methodology that gives number of calculation with respect to Hamming code has considered to be in programming level and it also utilizes expansion of number qualities to distinguish and right delicate errors. Comparatively recent study as given a hint that these demonstrated methodology has less delay comparing to other methodologies.

1.1 PROBLEM DEFINITION

A modern literature survey conveys that scope for high reliability and high accurate error correction and detection exists using Decimal Matrix Code [1], [2], [3] and [4]. The technique of making less number of the redundant bits and the maintaining the reliability of the planned task is a challenging work as taking into account of fast accessing over a large data.

An attempt has been made to detect and correct more errors by completely understanding the fundamental concepts of DMC. The main objective is to:

- Maintain higher reliability of memory.
- Fault-tolerant memory protected with DMC can be simulated by using Xilinx.
- Comparing the area, delay, power with different papers.

Finding power and area using cadence.

1.2 OVERVIEW OF PREVIOUS WORKS

In memories errors occurs and those errors are detected and corrected by various methods like Hamming codes and self-checking techniques but these are not competence to the present generation need of accuracy and high speed. It is easy to detect error but it's difficult to correct them because correcting error into again a fault error will create a false positive error which cannot be identified and hence it becomes a big problem. The system created for identifying the faults should give the assurance that output codes given are not the fault codes. The decision of the output data code is an exceptionally discriminating errand. They chose code that has mistake identification capacities can less demanding accomplish the fault secure property however includes an extensive number of outputs, along these lines, expanding equipment cost. Then again, picking a code of less error recognition capacities would include fewer additional outputs at the same time, for accomplishing shortcoming sacredness, so then it may be required to change the circuit structure (and along these lines additional expense). Now that it's out in the open, the determination of the generated code may be examined by the specific circuit, to get the finest result.

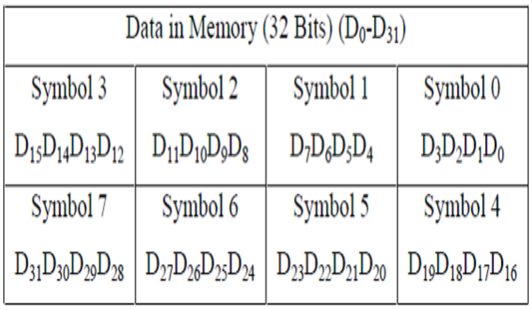
2. DESIGNS AND IMPLEMENTATION

DMC is planned to guarantee dependability of Multi bit upsets with lessened execution overheads. To recognize the right mistakes in the circuit a 32-bit of encoding and decoding is proposed in my project work. Various system architecture and methodologies are discussed.

2.1 System Design

Proposed work incorporates gap of image and organization of basic thoughts are achieved, that N-bit data is shared out to K image length of multiple-bits. The expression can be written as (N=K × m) where (K=K1 × K2, lines and segments). As shown in Table 4.1, a 32- bit code is divided into 8 sub symbols and again in that 4-bit are chosen simultaneously. Here while detecting and correcting maximum number of errors the check bits are different for all the K-bits generated. By properly selecting the values of K and m, the maximum bits can be corrected.

Table -1 Logical Organization of 32-bit data



Data in Memory (32 Bits) (D_0-D_{31})			
Symbol 3 $D_{15}D_{14}D_{13}D_{12}$	Symbol 2 $D_{11}D_{10}D_9D_8$	Symbol 1 $D_7D_6D_5D_4$	Symbol 0 $D_3D_2D_1D_0$
Symbol 7 $D_{31}D_{30}D_{29}D_{28}$	Symbol 6 $D_{27}D_{26}D_{25}D_{24}$	Symbol 5 $D_{23}D_{22}D_{21}D_{20}$	Symbol 4 $D_{19}D_{18}D_{17}D_{16}$

2.2 Proposed DMC

The planned task is of error controlling scheme is portrayed in Figure 1 the data bits (D) are first given to the encoder. Output of the DMC encoder is Horizontal (H) and Vertical bits (V). These values are stored in the memory temporarily. The SRAM extra added redundancy tallied to that of information data during read operation. Then decoder gives the data bits which are corrected.

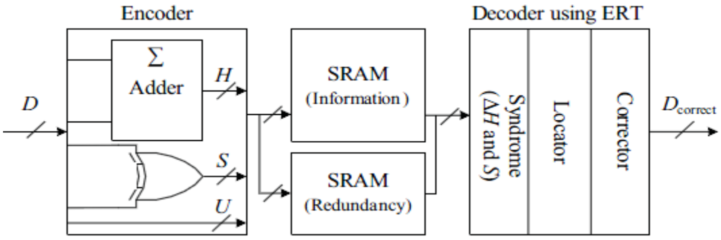


Fig-1 Proposed design schematic for fault-tolerant DMC

2.3 Planned Diagram of Fault-Tolerant Memory

As shown in the Figure 2, encoder which takes data bit d as input. Output of the DMC encoder is Horizontal (H) and Vertical bits (V). These values are stored in the memory temporarily. The SRAM redundant bits are compared with the information bits while read operation. Then decoder gives the data bits which are corrected that mean while read operations the errors can be corrected. This is an advantage of DMC with high performance and high fault-tolerant capacity, ERT technique is used to optimize the area.

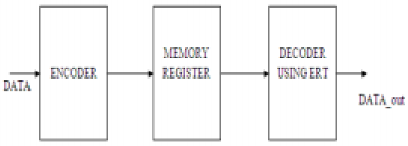


Fig-2 Block Diagram of DMC for fault-tolerant

2.4Planned Decimal Matrix Code Encoder

This depicts the, gap image and organize matrix thoughts are achieved, that is the data N-bit is divided into K image of length m-bits. The expression can be written as (N=K × m) where (K=K1 × K2, lines and segments). It's not necessary to change design to implementing the idea of project main schematic. Let us take 32-bit data and explain the DMC scheme. Total 32 bits are divided into 8 parts of size 4 bits. From D0 to D31 are 32 information bits, check bits are 10 Horizontal (Ho-H9) and 16 vertical bits. The detection and correction ability depends on the how we choose the values of k and m. trade-off occurs in choosing these values for maximum performance.

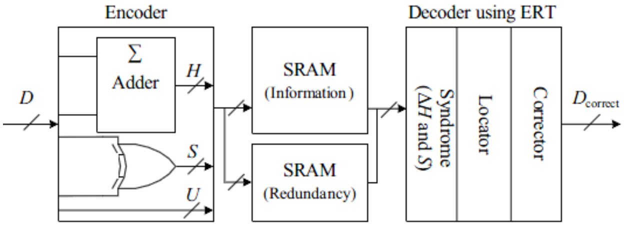


Fig-3 Applied schematic fault tolerant

Subsequently, m and k ought to be painstakingly changed in accordance with augment the redress ability and also diminish the quantity of excess bits. In this instance, for this situation, when m = 8 and k = 2×2, then only 1-bit fault is remedied, in this quantity of repetitive is 40bits.When m = 2 and k = 4×4, then 2-bit fault is remedied, in this quantity of repetitive bits is 32.When m = 4 and k = 2×4, then 5-bit fault is remedied, in this quantity of repetitive bits is 36. In this thesis, with a specific end goal to upgrade the dependability of memory, the mistake remedy capacity is initially considered, so k = 2 × 4 and m = 4 are used to develop
Then can be obtained by:

$$H4H3H2H1H0 = D3D2D1D0 + D11D10D9D8 \qquad (1)$$

$$H9H8H7H6H5 = D7D6D5D4 + D15D14D13D12 \qquad (2)$$

Equally for the horizontal dismissed$H14H13H12H11H10$ and $H19H18H17H16H15$,

For the we have by XOR of data bits.

$$V0 = D0 \oplus D16 \qquad (3)$$

$$V1 = D1 \oplus D17 \qquad (4)$$

Using the above equations (1) to (4), encoding is formed with the help of Decimal and Binary operations.

2.5 Planned Decoder for DMC

To acquire a data is remedied the translating procedure has to needed. Case in point, to begin with, they got repetitive bits H4H3H2H1H0 and V'0−V'3 are obtained from available data. Even disorder horizontal bits ΔH4H3H2H1H0 and straight down disorder bits S3 − S0 are ascertained as takes.

$$\Delta H4H3H2H1H0 = H4H3H2H1H0 - H4H3H2H1H0 \qquad (5)$$

$$S0 = V'0 - V0 \qquad (6)$$

Furthermore, correspondingly to other rest vertical disorder bits, here "−"signifies decimal whole number deduction. At the point when these values _H4H3H2H1H0 and S3 − S0 are equivalent to the value zero. At the point when these values _H4H3H2H1H0, S3 − S0 becomes nonzero, the incited mistakes can be distinguished and situated for image 0, afterward those errors can be removed easily.

$$D0correct= D0 \oplus S0. \ (7)$$

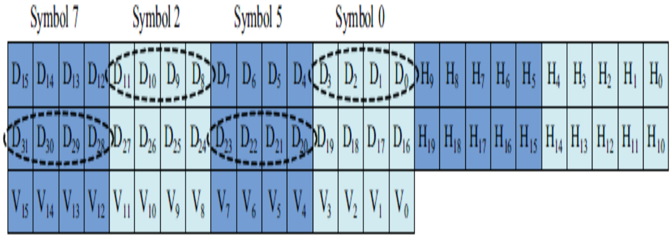

Symbol 7
Symbol 2
Symbol 5
Symbol 0
D15 D14 D13 D12 D11 D10 D9 D8 D7 D6 D5 D4 D3 D2 D1 D0 H9 H8 H7 H6 H5 H4 H3 H2 H1 H0
D31 D30 D29 D28 D27 D26 D25 D24 D23 D22 D21 D20 D19 D18 D17 D16 H19 H18 H17 H16 H15 H14 H13 H12 H11 H10
V15 V14 V13 V12 V11 V10 V9 V8 V7 V6 V5 V4 V3 V2 V1 V0

Fig-4 32-bits DMC.

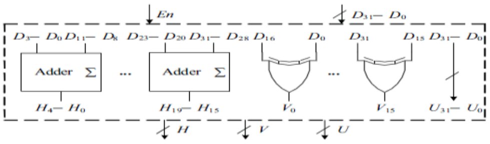
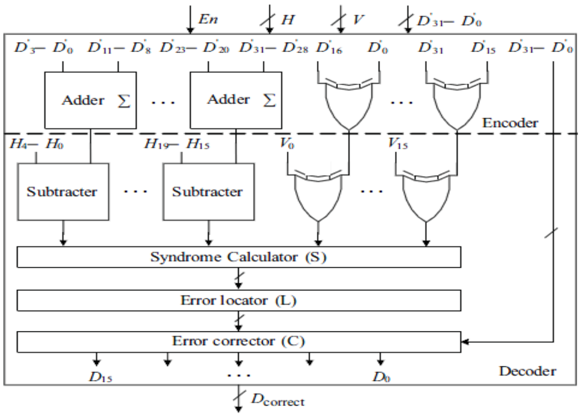


Fig-5 Design of 32-bit encoder for DMC using adders and XOR gates



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

Fig-6 A 32-bit decoder for DMC using ERT

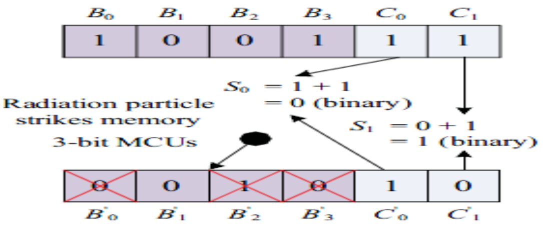


Fig.-7 Restrictions of error in Binary operations

2.6 Bound of Binary Error Detection

In this planned twofold fault discovery method, in spite of the fact that it obliges low repetitive bits, its slip recognition capacity is constrained. The principle purpose behind this is that its error discovery component is in view of binary. We show the points of confinement of this basic paired slip recognition utilizing a straightforward illustration. Figure 7 shows Binary bits (B3 to B0) which are unique and data bits (C0 and C1) are redundant bits.

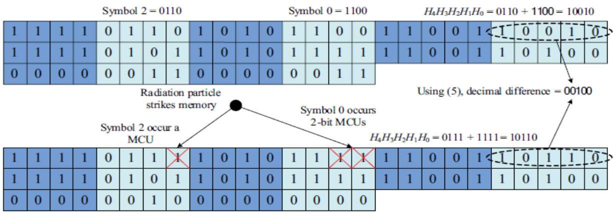


Fig.-8 Benefit locating glitches

By solving, $H4H3H2H1H0$ does not to be zero. It is characterizes that MCUs is noticed and altered for decoding errors.

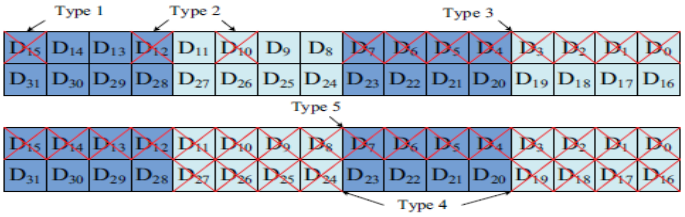

Type 1
Type 2
Type 3
Type 5
Type 4
D15 D14 D13 D12 D11 D10 D9 D8 D7 D6 D5 D4 D3 D2 D1 D0
D31 D30 D29 D28 D27 D26 D25 D24 D23 D22 D21 D20 D19 D18 D17 D16

Fig.-9 Adjustment of MCUs with DMC.

Sort 1 is a solitary mistake, sort 2 is not following an continuo's sequence slip in dual successive images, sort 3 is a sequential mistake in dual continuous images, sort 4 is similar to sort 2 with two discontinuous images, back to back slip in four successive images seen in 5.

$$C_0 = B_0 \oplus B_2 = 1 \oplus 0 = 1 \qquad (8)$$
$$C_1 = B_1 \oplus B_3 = 0 \oplus 1 = 1. \qquad (9)$$

$$C_0^{'} = B_0^{'} \oplus B_2^{'} = 0 \oplus 1 = 1 \qquad (10)$$
$$C_1^{'} = B_1^{'} \oplus B_3^{'} = 0 \oplus 0 = 0. \qquad (11)$$

$$S_0 = C_0^{'} \oplus C_0 = 1 \oplus 1 = 0 \qquad (12)$$
$$S_1 = C_1^{'} \oplus C_1 = 0 \oplus 1 = 1. \qquad (13)$$

These outcomes imply that mistake bit values B2 and B0 were incorrectly viewed as the first bits because of that errors bits are not revised. From the above example it's clear that an even order error cannot be identified in binary operation.

2.7 Decimal Error Detection Benefits

Previously it has been demonstrated as error location taking into account parallel calculation can just distinguish a limited number of errors. In any case, when these decimal calculations are utilized for distinguish errors, then these errors can be identified such that unraveling error can be stayed away from. The reason for this operation component of calculation is not the same as that of paired. The location method of decimal error discovery utilizing the proposed structure indicated as a part of Fig. 2 is completely depicted in Fig. 6. As a matter of first importance, the flat excess H4H3H2H1H0 are of first data bits in images 0 and 2 as per 1

$$\begin{aligned} H_4H_3H_2H_1H_0 &= D_3D_2D_1D_0 + D_{11}D_{10}D_9D_8 \\ &= 1100 + 0110 \\ &= 10010. \end{aligned} \qquad (14)$$

At the point in MCUs happen to be image 0, image 2, then the bits in image 0 surprise to "1111" from the "1100"(D3D2D1D0= 1111) then bits in image 2 may be miracle to "0111" from "0110" (D11D10D9D8= 0111). Amid the disentangling migration, got flat repetitive H4H3H2H1H0 are initially processed.

$$\begin{aligned} H_4H_3H_2H_1H_0^{'} &= D_{11}D_{10}D_9D_8^{'} + D_3D_2D_1D_0^{'} \\ &= 0111 + 1111 \\ &= 10110. \end{aligned} \qquad (15)$$

Decimal integer subtraction gives *H4H3H2H1H0* syndrome bits .

$$\begin{aligned} \Delta H_4H_3H_2H_1H_0 &= H_4H_3H_2H_1H_0^{'} - H_4H_3H_2H_1H_0 \\ &= 10110 - 10010 \\ &= 00100. \end{aligned} \qquad (16)$$

The decimal estimation of_H4H3H2H1H0 are not zero which speaks to the errors are recognized and situated to image 0 or image 2. Therefore, the exact area these flopped bits are situated to utilizing the vertical disorder bits S3 – S0 or S11 – S8. At long last, every one of these errors can be amended by (7). Thusly, in light of decimal calculation, the proposed system has higher resilience ability for ensuring memory against M

Accordingly this is conceivable to single and decimal slips and also any other sorts of numerous errors per line could be revised to proposed strategy regardless of whether errors are back to back or inconsecutive. As shown in the Figure 9.

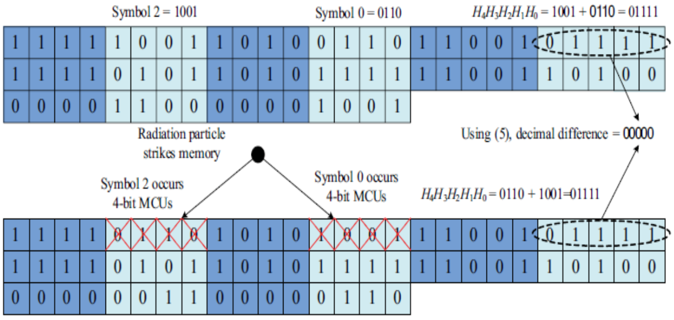


Fig.-10 Mistake Find cannot be modified by DMC.

One of the fundamental for these bits H4H3H2H1H0 is zero in decimal. Note that despite the fact that 7-bit slips happen in images 0 and 2 at the same time, the disentangling error can be declined can undoubtedly right surprises of sort 1, 2, and 3, in light of the fact that they can be crucial properties of DMC: all single-error and numerous mistake remedies in two sequential images. Miracles of sorts 4 and 5 presented are additionally redressed in light of the fact that the numerous faults per column could be recognized by the even disorder bits. These demonstrate as a system is an appealing choice to memory recollections from substantial MCUs. Then again, bombshells of sort 4 and 5, is vital to perceive this bring about interpreting slip when the accompanying essential variables are accomplished.

1) Information bits in symbols with decimal integer.

0 and 2 are equal to 2*m* −1 and

2) Every bits in symbols 0 and 2 are troubled.

Accepting that these 2 components have been accomplished, by encoding and disentangling procedures of DMC, H4H3H2H1H0, and are registered, as takes after:

$$\begin{aligned} H_4H_3H_2H_1H_0 &= D_3D_2D_1D_0 + D_{11}D_{10}D_9D_8 \\ &= 0110 + 1001 \\ &= 01111 \end{aligned} \qquad (17)$$

$$\begin{aligned} H_4H_3H_2H_1H_0' &= D_3D_2D_1D_0' + D_{11}D_{10}D_9D_8' \\ &= 1001 + 0110 \\ &= 01111. \end{aligned} \qquad (18)$$

*H*4*H*3*H*2*H*1*H*0 are obtained

$$\begin{aligned} \Delta H_4H_3H_2H_1H_0 &= H_4H_3H_2H_1H_0' - H_4H_3H_2H_1H_0 \\ &= 01111 - 01111 \\ &= 00000. \end{aligned} \qquad (19)$$

The above results shows that in symbol 0 and 2 no errors occurred and memory would grieve a disappointment though, type is unusual. E.g. when *m* = 4, then likelihood of the decoding errors is:

$$P_{\Delta H=0} = 4 \times \left({}^{1}\!/_{2^4}\right)^2 \times P_{\text{MCU8}} \approx 0.001. \qquad (20)$$

If $m = 8$

$$P_{\Delta H=0} = 4 \times \left({}^{1}\!/_{2^8}\right)^2 \times P_{\text{MCU16}} \approx 0.0000011. \qquad (21)$$

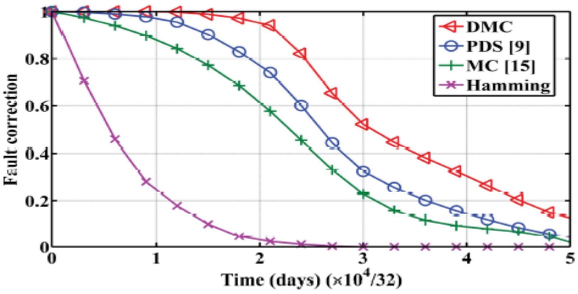
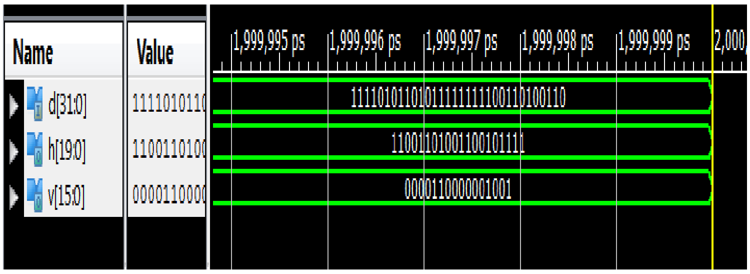


Fig-11 Time Difference security code with J(S)

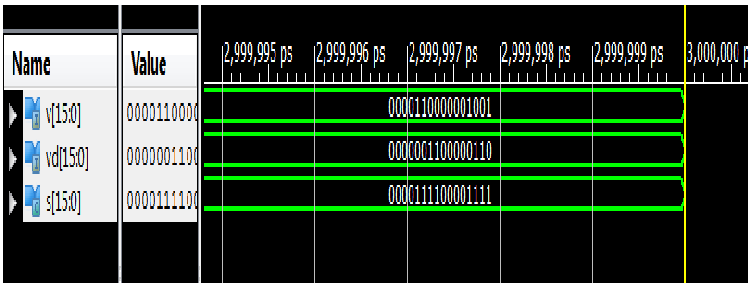
The Figure 11 shows the plot of 8 upsets in data, from this it can be concluded that memory usually contains very small amount of regular order of errors and into a space of these errors length is not more than 3 bits, it's not big dispute.

2.8 Results and Discussion



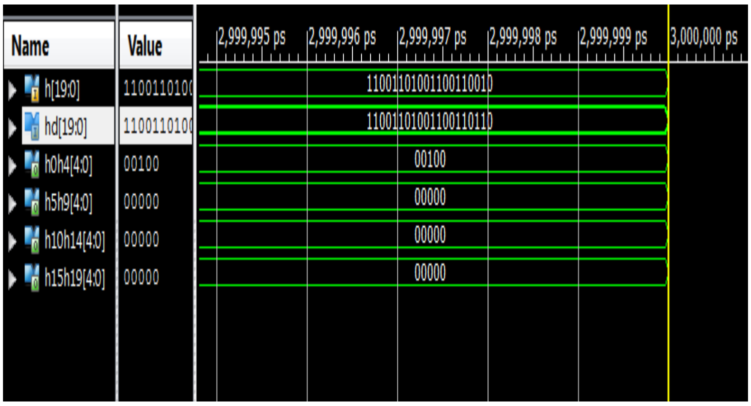
Name	Value	
d[31:0]	1111010110	11110101101011111111100110100110
h[19:0]	1100110100	11001101001100101111
v[15:0]	0000110000	0000110000001001

Fig.-11 DCM Encoding Output



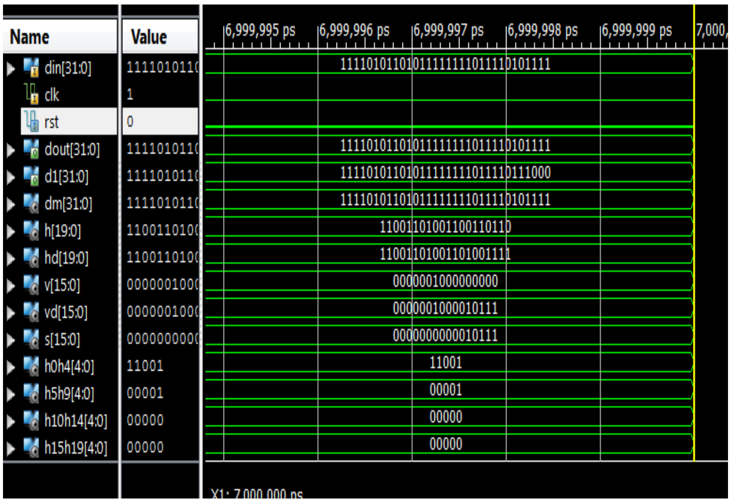
Name	Value	
v[15:0]	0000110000	0000110000001001
vd[15:0]	0000001100	0000001100000110
s[15:0]	0000111100	0000111100001111

Fig.-12 DCM Decoding output



Name	Value	
h[19:0]	1100110100	11001101001100110010
hd[19:0]	1100110100	11001101001100110110
h0h4[4:0]	00100	00100
h5h9[4:0]	00000	00000
h10h14[4:0]	00000	00000
h15h19[4:0]	00000	00000

Fig.-13DCM Detecting output



Name	Value	
din[31:0]	1111010110	11110101101011111111011110101111
clk	1	
rst	0	
dout[31:0]	1111010110	11110101101011111111011110101111
d1[31:0]	1111010110	11110101101011111111011110111000
dm[31:0]	1111010110	11110101101011111111011110101111
h[19:0]	1100110100	11001101001100110110
hd[19:0]	1100110100	11001101001101001111
v[15:0]	0000001000	0000001000000000
vd[15:0]	0000001000	0000001000010111
s[15:0]	0000000000	0000000000010111
h0h4[4:0]	11001	11001
h5h9[4:0]	00001	00001
h10h14[4:0]	00000	00000
h15h19[4:0]	00000	00000

Fig.-14 DCM Top Decimal Output

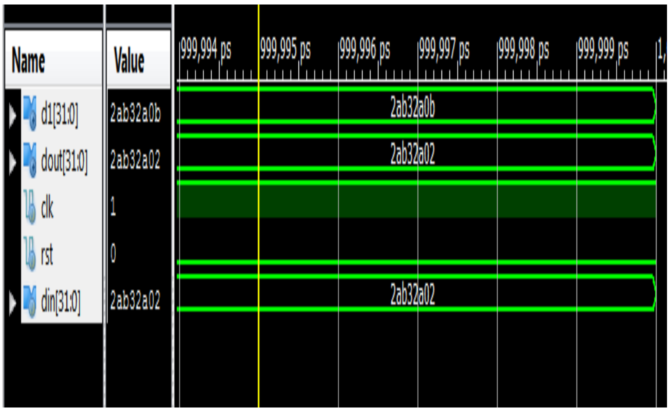


Fig.-15 DCM Final Output

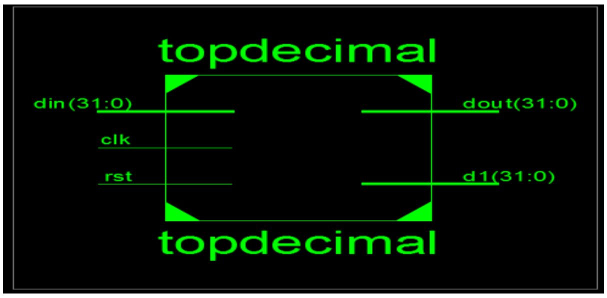
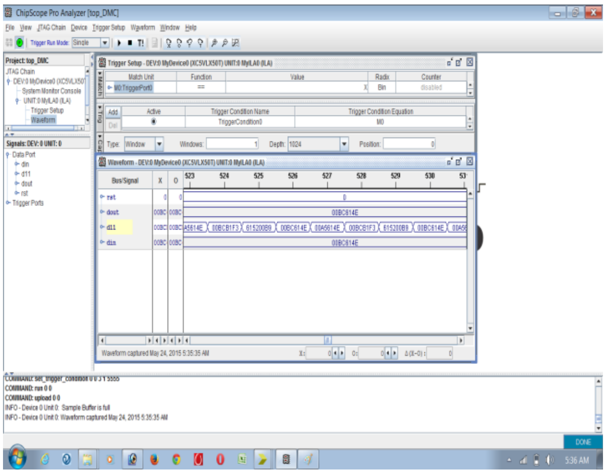


Fig.-16 RTL Schematic



Waveform captured May 24, 2015 5:35:35 AM

```
COMMAND: set_trigger_condition 0 0 3 1 5555
COMMAND: run 0 0
COMMAND: upload 0 0
INFO - Device 0 Unit 0:  Sample Buffer is full
INFO - Device 0 Unit 0: Waveform captured May 24, 2015 5:35:35 AM
```

Fig.-17 Vertex5 Hardware output

Table -2: Error Correction capability

Types of applied ECC	no. of Information bits	no. of redundant bits	error correction capability
proposed DMC	32	36	5
matrix code	32	28	2
hamming code	32	7	1

2.9 Applications

- This can easily incorporated in VLSI design system.
- This is highly preferable for the applications of security basis.
- Flip flops and registers used are basically deals in lowering energy.
- This will preferably detects data signals having glitches and correcting the signals.
- Gives more reliable result, this idea is more efficient in defense to maintain encryption of data.

3. CONCLUSIONS

Error rectification codes are utilized to enhance the memory insurance and make the memory error free. It utilized to distinguish a event of error furthermore corrected. The proposed DMC system achieves in detecting and correcting the errors very efficiently using the decimal algorithm method. In comparing to previous works the Enhanced DMC have the capability to correct errors up to 5 bits. The proposed system shows that it has superior protection level comparing to the large multiple cell upsets in the memory cells.

ACKNOWLEDGEMENT

I am graceful to my guide for supporting me in my project and my presentation, suggesting in every mistake done by me

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 (VLSI) SYSTEMS, VOL. 22, NO. 1, pp. 127-135, JANUARY 2014

[2]. Riaz Naseer and Jeff Draper, "Parallel Double Error Correcting Code Design to Mitigate Multi-Bit Upsets in SRAMs," 978-1-4244-2361-3/08/$25.00 ©2008 IEEE. Abbreviation

[3]. T.Maheswari and Mr.P.Sukumar, "Error Detection and Correction in SRAM Cell Using Decimal Matrix Code," IOSR Journal of VLSI and Signal Processing (IOSR-JVSP) Volume 5, Issue 1, Ver.II (Jan – Feb. 2015), PP 09-14

[4]. Juan Antonio Maestro, Pedro Reviriego, Sanghyeon Baeg, Shijie Wen, Richard Wong, "Soft error tolerant Content Addressable Memories (CAMs) using error

detection code and duplication," @2013 Elsevier B.V.

[5]. Gustavo Neuberger, Fernanda Gusmao de lima, kastensmidt and Ricardo Reis, "An Automatic Technique for Optimizing Reed – Solomon Codes to Improve Fault Tolerance in Memories," IEEE Design & Test of Computers, Copublished by the IEEE CS and the IEEE CASS @2005 IEEE

[6]. K.Madhuri, V.Thrimurthulu, "Implementation of Decimal Matrix Code for Correcting Cell Upsets in Static Random Access Memories," International Journal of Electrical and Data Communication, ISSN: 2320 – 2084, Volume-2, Issue-10, Oct-2014

[7]. Sanghyeon Baeg, Pedro Reviriego, Juan Antonio Maestro, Shijie and Richard Wong, "Analysis of a Multiple Cell Upset Failure Model for Memories"

[8]. 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," Proc. Of Int. Conf. on Recent Trends in Signal Processing, Image Processing and VLSI, ICrtSIV, ACEEE,2014

[9]. Costas Argyrides, Stephania Loizidou and Dhiraj K. Pradhan, "Area Reliability Trade – Off in Improved Reed Muller Coding," SAMOS 2008, LNCS 5114, pp. 116-125, 2008 Springer – Verlag Berlin Heidelberg 2008

[10].X Susan Christiana, K Javid and C Mahalakshmi, "Comparative Analysis of Decimal Matrix Code with parity matrix code," National Conference on "Recent Trends in Communication & Information Technologies" NCRTCIT, Special Issue, Vol.1, No. 1, March 2015