

# FPGA Implementation of Error Detection and Correction using Decimal Matrix Code

**Lakshmeepathippappa K B**

*PG Student*

*Department of Digital Electronics and Communication System  
VTU, CPGS, Bengaluru region VAIT, Muddenahalli,  
Chikkaballapur, Bengaluru*

**Dr. Sarika Tale**

*Associate Professor*

*Department of Digital Electronics and Communication System  
VTU, CPGS, Bengaluru region VAIT, Muddenahalli,  
Chikkaballapur, Bengaluru*

## Abstract

Currently protection codes are essential to memory cells using to maintain a good reliability, various error detection and error correction methods can be used but to avoid corruption data Error correction code (ECC) are mostly widely used and introduced a delay penalty in accessing the encoding and decoding process can be performed but limitation of ECC are low speed performed in memory .it can be contributed to the utilization of simple code such as single error correction (SEC) and Double error correction (DEC) and detection. Matrix Codes (MC) constructed on hamming codes has been proposed for memory protection. The main dispute is that they are double error correction codes and the error correction capabilities are not enhanced in all cases. Decimal Matrix Code (DMC) structure on matrix and divide symbol is proposed to improve memory reliability with lesser delay. In this thesis, 64-bits and 128-bits Decimal Matrix Code is offered to assure the memory reliability. Here to detect and correct up to 9 and 17 errors respectively. The offered DMC based on decimal algorithm to extended the number of detection and correction capability of error bits stored in memory. More ever structure area of DMC is reduced by reusing its Encoder this is called Encoder Reuse Techniques (ERT). ERT can be used to minimize area without interrupting DMC encoder and decoder. ERT Uses DMC Encoder itself to be a part of decoder. Hence the entire structure area of DMC can be minimized. Using Xilinx Design Suite 14.2 simulation and implementation analysis can be done. Next A Verilog Description has been adopted to embed the low power design.

**Keywords:** ECC, DMC, ERT, MC, SEC

## I. INTRODUCTION

Now a day to keep up sensible level of reliability, it's terribly essential safeguarding the memory cells victimization protection codes for his or her drive, various errors detection and correction ways are being employed To avoid knowledge corruption, reminiscences are often protected with a slip correction code. It's given that almost all single bit errors are often detected extra kind of fault will upset its correct operation. This error, referred to as one event upset (SEU) or soft error.

Single event upset could be a bit-flip within the memory cell obtained of a transient gift pulse generated by ionization. Charged particles coming back from sun activity and neutrons that strike the fabric will cause this ionization. Additional than one Single event upset will occur at an equivalent time during a memory array. This can be referred to as multiple bits upset (MBU) .it may end up from a high-energy particle doubtless inflicting double bit upset or an occasional incident angle distinguished several cells during a row. Experiments victimization nucleon and heavy-ion fluxes calculate the chance of one particle agitating MBU. Error detection and correction code (EDAC) could be a recognized technique for safeguarding storage devices against transient errors. Associate example of Error detection and correction code relies on performing code is helpful for safeguarding recollections compared to Single Event Upset owing to its effective ability to express single upsets per coded word with reduced space and performance higher than. There square measure business recollections that use performing code for high-responsibleness applications to extend turn out or error tolerance. so performing code isn't able to make sure the responsibleness of once producing defects and MBUs from the atmosphere square measure gift. There square measure alternatives to EDAC, like and RS (Reed-Solomon) and (Bose-Chaudhuri- Hocquenghem) codes, supported finite-field (also called mathematician field) arithmetic, which may address multiple faults. Given variety of bits at any position will correct victimization BCH codes. Whereas RS codes cluster the bits in blocks to correct them. Their disadvantages square measure that they need complicated and repetitious cryptography algorithms, and use tables of constants within the formula. However, studies have proved that eliminating the tables of constants will alter RS codes which the cryptography formula is easier within the case of single-block correction.

RS rather than BCH and similar code reduce this impact Error correction codes that secret writing is straightforward square measure employed in most cases. Single design of codes that secret writing is finished low interruption is Single Error Correction (SEC) codes. Single error correction codes square measure smallest distance of 3 so a double error is incorrect for one error and speciously corrected. to avoid this concern Single Error Correction Double Error Detection codes should be a minimum distance.

There square measure totally different choices for SEC-DED codes which will reduce fault error probability for triple errors. We acquire lower delay we've to use overacting codes easy to conception.

In this thesis, 64-bits and 128-bits Decimal Matrix Code was planned to declare the reliableness of memory .The maximum detect and correct up to 9 and 17 errors respectively. Decimal matrix code (DMC) conception on divide-symbol is obtainable to boost memory reliableness with the employment of lower delay the planned DMC uses decimal whole number addition and decimal whole number subtraction is employed to get the quantity of error detection proficiency. To boot, the circuit space of Decimal Matrix Code is reduced by reprocessing encoder half. This is often referred to as the encoder-reuse technique (ERT). ERT Technique is employed in decoder half. The Encoder employ Techniques will reduction space of DMC while not displeasing the total encryption and secret writing processes. The Encoder employ Technique uses Decimal Matrix Code encoder this one to be a part of the decoder. Therefore, the entire circuit space of Decimal Matrix Code is decreased of encoder part. Next A Verilog Description has been adopted to embed the low power design.

### A. Problem Definition

Transient multiple cell upsets (MCUs) are getting significant issue within the reliableness of reminiscences visible to emission surroundings. To avoid MCUs from moving knowledge corruption, extra multifaceted error correction codes (ECCs) square measure sometimes accustomed safeguard memory however the most disadvantages it need larger delay overhead. Lately new matrix codes (MCs) established on acting codes relies on acting code. However main downside double error correction codes and also the error correction experiences aren't increased all told cases. Decimal matrix code (DMC) created on divide-symbol is obtainable to extend memory reliableness with lower delay. The planned Decimal Matrix Code uses decimal rule to achieve the determined error detection capability. The encoder-reuse technique (ERT) is obtainable to decrease the realm while not displeasing the entire cryptography and decryption ways. Encoder employ Technique uses DMC encoder himself to be portion of the DMC decoder.

### B. Existing Method

To avoid Multiple Cell Upsets from causing data corruption more problematic error correction codes (ECCs) are generally used to protect memory. The main difficult only double error correction codes and the error correction proficiencies are not superior. To overcome these defects we propose the new method explained below.

### C. New Method

In this thesis, Decimal Matrix Code (DMC) arranged divide-symbol is offered to improve memory dependability with lower delay. The offered DMC uses decimal algorithm uses decimal number addition and decimal number subtraction to obtain the maximum error detection capability. The encoder-reuse technique (ERT) is offered to reduce the area without upsetting the complete encoding and decoding processes.

## II. PROPOSED SYSTEM

### A. DMC Encoder and Decoder

In this proposal theory, Different Decimal Matrix Code (DMC) made on matrix ideas is presented to offer enhanced memory dependability. Using Decimal algorithm to obtain decimal addition and decimal subtraction to identify the error. Using Decimal algorithm we can use number of error detection capability is increased thus the reliability of memory is improved. Also, with the use of Encoder-Reuse Technique (ERT) to decrease the area upstairs of additional circuits without upsetting the complete encoding and decoding methods, since ERT uses DMC encoder this one to be part of the decoder. In this thesis, Decimal Matrix Code is presented to declare dependability of the memory in the presence of Multiple Cells Upset's with reduced routine overheads and 64-bits and 128 bits word is encoded and decoded based on the offered techniques.

#### 1) Proposed Schematic of Fault-Tolerant Memory

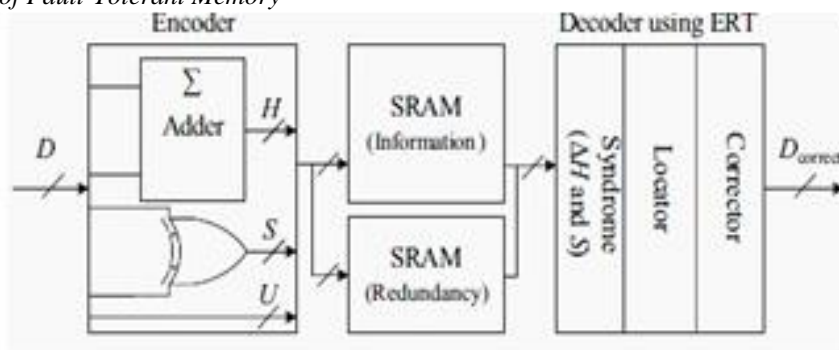


Fig. 1: Proposed schematic of fault-tolerant memory protected with DMC

The presented schematic of fault-tolerant memory is represented in Fig. 1. In encoding procedure, the information bits D are applied to the encoder, and then from the encoder part the vertical redundant bits V and the Horizontal redundant bits H are obtained. After Completion of encoder part then the DMC code word is stored in memory. In case Multiple Cells Upsets occur in the memory, these faults can be modified in the decoding process. The offered Decimal Matrix Code has developed fault-tolerant proficiency with lower performance. The main advantage of decimal algorithm using ERT technique. We can minimize the area without disturbing the encoder and decoder circuit in the fault-tolerant memory.

**B. Proposed DMC Encoder using 64 bits**

In the offered DMC, the principal DMC is constructed based on arrange matrix types and divide-symbol are done and the number of N-bit word is divided into k symbols of m bits and these symbols are arranged in a k1 × k2 where k1 and k2 are symbols for the number of rows and columns in matrix respectively. By performing the decimal number addition are obtained horizontal redundant bits H selected symbols per row. And the binary operation is obtained vertical redundant bits V selected symbols per column. i.e can be operated both arranged matrix ideas type and divide symbols are executed in logical memory instead of in physical. So, the offered DMC does not need altering the physical structure of the memory.

The offered DMC structure, the cells from D0 to D63 are occupied as information bits. This 64-bit word has been divided into eight symbols of 4 bit. k1 = 2 and k2 = 4 have been chosen simultaneously. H0–H35 are horizontal check bits. V0 through V31 are vertical check bits. It should be mentioned that the maximum correction capability and the number of redundant bits are different when the different values for k and m are chosen. In case, to take k (symbols) = 2\*2 and m (bits) = 8, only number of error correction bits is 1-bits and 40 is the total number of redundant bits. Similarly when k (symbols) = 4\*4 and m (bits) = 2, only number of error correction bits is 3-bit and 32 is the total number of redundant bits reduced Yet, when k (symbols) = 2\*4 and m (bits) = 4, the total number of maximum correction proficiency is 9 bits and 68 is the total number of redundant bits can be performed.

The decimal and binary addition operations performs the encoding. Multi-bit adders are used to compute redundant bits by encoder and XOR gates is shown in Fig. 3. In this figure the horizontal redundant bits are H35 – H0, the vertical redundant bits are V31 – V0 and the remaining information bits are U63 – U0 which are directly copied from D63 to D0. In this thesis, In order to increase the dependability of memory, the error correction capability is first careful, so k = 2 × 4 and m = 8 are utilized to construct DMC. The Horizontal redundant bits H can be obtained by decimal addition as follows:

$$H_8H_7H_6H_5H_4H_3H_2H_1H_0 = D_7D_6D_5D_4D_3D_2D_1D_0 + D_{23}D_{22}D_{21}D_{20}D_{19}D_{18}D_{17}D_{16} \dots \dots \dots (1)$$

$$H_{17}H_{16}H_{15}H_{14}H_{13}H_{12}H_{11}H_{10}H_9 = D_{15}D_{14}D_{13}D_{12}D_{11}D_{10}D_9D_8 + D_{31}D_{30}D_{29}D_{28}D_{27}D_{26}D_{25}D_{24} \dots \dots (2)$$

And likewise Horizontal Redundant bits H<sub>14</sub>H<sub>13</sub>H<sub>12</sub>H<sub>11</sub>H<sub>10</sub> and H<sub>19</sub>H<sub>18</sub>H<sub>17</sub>H<sub>16</sub>H<sub>15</sub>, where “+” represents decimal number addition. For the vertical redundant bits V, we have

$$V_0 = D_0 \wedge D_{31} \dots \dots \dots (3)$$

$$V_1 = D_2 \wedge D_{33} \dots \dots \dots (4)$$

And similarly for the rest vertical redundant bits.

**C. Proposed DMC Decoder using 64 bits**

The offered DMC decoder is containing the following sub modules, and each performs certain exact tasks in the decoding process: syndrome calculating, error locating, and error correcting.

The decoding process is required to obtain a word being corrected. The obtained redundant bits H<sub>8</sub>H<sub>7</sub>H<sub>6</sub>H<sub>5</sub>H<sub>4</sub>H<sub>3</sub>H<sub>2</sub>H<sub>1</sub>H<sub>0</sub> and V<sub>0</sub>'-V<sub>7</sub>' are formed by the received information bits D'. Second, the horizontal syndrome bits ΔH<sub>4</sub>H<sub>3</sub>H<sub>2</sub>H<sub>1</sub>H<sub>0</sub> and the vertical syndrome bits S<sub>7</sub> – S<sub>0</sub> can be calculated as follows:

$$\Delta H_8H_7H_6H_5H_4H_3H_2H_1H_0 = H_8H_7H_6H_5H_4H_3H_2H_1H_0' - H_8H_7H_6H_5H_4H_3H_2H_1H_0 \dots \dots (5)$$

$$S_0 = V_0' \wedge V_0 \dots \dots (6)$$

And similarly for the rest vertical syndrome bits, where “-” represents decimal integer subtraction. When ΔH<sub>8</sub>H<sub>7</sub>H<sub>6</sub>H<sub>5</sub>H<sub>4</sub>H<sub>3</sub>H<sub>2</sub>H<sub>1</sub>H<sub>0</sub> and S<sub>7</sub> – S<sub>0</sub> are equal to zero, the stored code word has original information bits in symbol 0 where no errors occur. When ΔH<sub>8</sub>H<sub>7</sub>H<sub>6</sub>H<sub>5</sub>H<sub>4</sub>H<sub>3</sub>H<sub>2</sub>H<sub>1</sub>H<sub>0</sub> and S<sub>7</sub> – S<sub>0</sub> are nonzero, the made 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 Formula

$$D_{0correct} = D_0 \wedge S_0 \dots \dots (7)$$

**D. Proposed DMC Encoder using 128 bits**

The offered DMC structure, we take a 128-bit word. D0 to D127 are information bits cells.128-bit word can be selected into eight symbols of 8 bits cells. Where k1 =2 and K2 = 4 must selected simultaneously. H0–H67 is horizontal check bits. V0 through V31 are vertical check bits. When k = 2 × 4 and m = 16, the maximum correction capability is up to 17 bits and the number of redundant bits is 132.

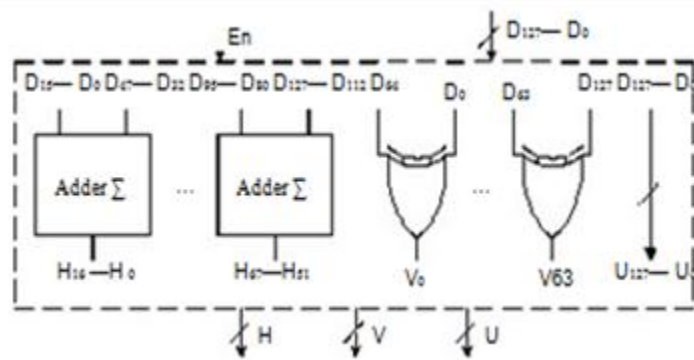


Fig. 5: 128-bit DMC encoder structure using multi-bit adders and XOR gates

From decimal number addition the horizontal redundant bits H can be obtained as follows:

$$H_{16}H_{15}H_{14}H_{13}H_{12}H_{11}H_{10}H_9H_8H_7H_6H_5H_4H_3H_2H_1H_0 = D_{15}D_{14}D_{13}D_{12}D_{11}D_{10}D_9D_8D_7D_6D_5D_4D_3D_2D_1D_0 + D_{47}D_{46}D_{45}D_{44}D_{43}D_{42}D_{41}D_{40}D_{39}D_{38}D_{37}D_{36}D_{35}D_{34}D_{33}D_{32} \dots (8)$$

$$H_{67}H_{66}H_{65}H_{64}H_{63}H_{62}H_{61}H_{60}H_{59}H_{58}H_{57}H_{56}H_{55}H_{54}H_{53}H_{52}H_{51} = D_{95}D_{94}D_{93}D_{92}D_{91}D_{90}D_{89}D_{88}D_{87}D_{86}D_{85}D_{84}D_{83}D_{82}D_{81}D_{80} + D_{127}D_{126}D_{125}D_{124}D_{123}D_{122}D_{121}D_{120}D_{119}D_{118}D_{117}D_{116}D_{115}D_{114}D_{113}D_{112} \dots (9)$$

And similarly Horizontal Redundant bits  $H_{33}H_{32}H_{31}H_{30}H_{29}H_{28}H_{27}H_{26}H_{25}H_{24}H_{23}H_{22}H_{21}H_{20}H_{19}H_{18}H_{17}$  and  $H_{50}H_{49}H_{48}H_{47}H_{46}H_{45}H_{44}H_{43}H_{42}H_{41}H_{40}H_{39}H_{38}H_{37}H_{36}H_{35}H_{34}$ , where “+” represents decimal integer addition.

For the vertical redundant bits V, we have

$$V_0 = D_0 \wedge D_{63} \dots (10)$$

$$V_1 = D_2 \wedge D_{32} \dots (11)$$

And similarly like this....

**E. Proposed DMC Decoder using 128 bits**

The proposed 128 bit DMC decoder using ERT is denoted in Fig. 6.

The decoding process is required to obtain a word being corrected. For example,  $H_{16}H_{15}H_{14} H_{13}H_{12}H_{11}H_{10}H_9 H_8H_7H_6H_5H_4H_3H_2H_1H_0$  and  $V_0'-V_7'$  are the output redundant bit and information bits. And then the  $\Delta H_{16}H_{15}H_{14}H_{13}H_{12}H_{11}H_{10}H_9 H_8H_7H_6H_5H_4H_3H_2H_1H_0$  are Horizontal syndrome and  $S_{15} - S_0$  are Vertical syndrome bits can be calculated as follows:

$$\Delta H_{16}H_{15}H_{14} H_{13}H_{12}H_{11}H_{10}H_9 H_8H_7H_6H_5H_4H_3H_2H_1H_0 = H_{16}H_{15}H_{14} H_{13}H_{12}H_{11}H_{10}H_9 H_8H_7H_6H_5H_4H_3H_2H_1H_0' - H_{16}H_{15}H_{14} H_{13}H_{12}H_{11}H_{10}H_9 H_8H_7H_6H_5H_4H_3H_2H_1H_0 \dots (12)$$

$$S_0 = V_0' \wedge V_0 \dots (13)$$

$\Delta H_{16}H_{15}H_{14} H_{13}H_{12}H_{11}H_{10}H_9 H_8H_7H_6H_5H_4H_3H_2H_1H_0$  and  $S_{15} - S_0 = 0$ , Hence No Errors Occurs.

$\Delta H_{16}H_{15}H_{14} H_{13}H_{12}H_{11}H_{10}H_9 H_8H_7H_6H_5H_4H_3H_2H_1H_0$  and  $S_{15} - S_0 \neq 0$ , Hence Induced errors, this Error can be detected and located and then these errors can be corrected by Formula

$$D_{0correct} = D_0 \wedge S_0 \dots (14)$$

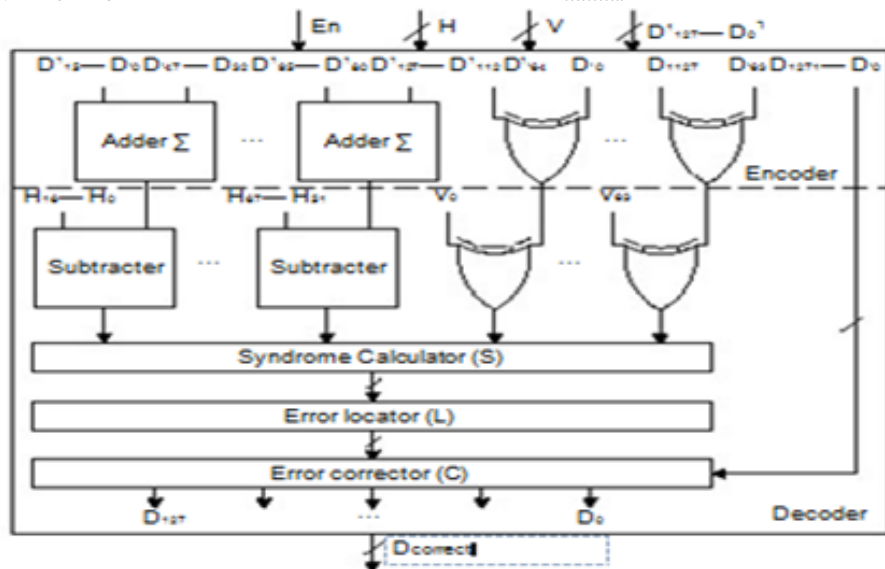


Fig. 6: 128-bit DMC decoder structure using ERT

### III. HARDWARE AND SOFTWARE IMPLEMENTATION

#### A. Hardware Implementation

FPGAs have recognized to be very effective and efficient procedures on which to implement procedures. They can perform a much quicker data-rates and to offer improved safety than corresponding software executions. They also offer more flexibility than ASIC executions.

The Field Programmable Gate Array (FPGA) is a controlling tool to recognize the functionality to the design. The main following benefits to select FPGA.

- It can support Reprogrammable hardware.
- Shorter progress agrees for fast time to market place.
- We have to Design and debug alterations can be through suddenly.
- It allows designing reuse, corresponding design, and SOC (system on a chip) design.
- To Designs migrated to ASICs using FPGA.
- To Decreases processor obsolescence.

#### B. Xilinx ISE

Xilinx ISE (Integrated Software Environment) is a software device which produced by synthesis and analysis of HDL designs, it can perform timing analysis which can be perform enabling the developer to synthesize designs, to inspect RTL diagrams, simulate a proposal's reaction to altered stimuli, and configure the target device with the programmer.

The Xilinx ISE is a complete FPGA/CPLD programmable logic design suite providing:

- Programmable specification of logic through representation of capture or Verilog/VHDL
- To Synthesis and Place & Route of identified logic for various Xilinx FPGAs and CPLDs
- To provide the timing simulation and Functional simulation
- To Copy of structure data into goal device through communications cable

The Xilinx ISE Design Suite offers an integrated flow with the ISE Simulator (ISim) which can allows simulations to launch directly from the Project Navigator (ISE). All simulation commands that prepare the ISim simulation are generated by ISE Project.

The Xilinx ISE Simulator is a Hardware Description Language (HDL) simulator that allows the computer operator to execute behavioral, functional and timing simulations for VHDL, Verilog and mixed-language designs. The ISim Graphical User Interface comprises the toolbars, wave window, status bar and the panels. From the main window we can vision the simulation-visible shares of the design add and vision signals in the wave window, employ ISim instructions to run the simulation program and to observe the design, and debug necessary.

Xilinx offers power approximation tools like Xilinx Power Estimator (XPE) and Xilinx Power Analyzer (XPA). XPE spreadsheet is a rule approximation device normally used pre-implementation phases of a project and pre-design of the project. It can helps with construction estimate and device collection and it helps select proper power supply and the current controlling apparatuses which may be essential for numerous applications.

For the application of pre-implementation device, in early stages XPE can be used in to design cycle after the RTL explanation of the design is partial. After completion of implement part the XPA device can be used to more accurate approximations and power enquiry. XPE shows tabs for each type of module in device architecture. Depending upon The number of tabs showed in XPE will differ provisional on the device architecture.

#### C. Software Requirements

##### 1) XILINX 14.3 ISE overview

The Integrated Software Environment (ISE) is the Xilinx proposal software suite that precedes the proposal from design entry complete Xilinx device software design. The ISE Project Navigator can be used to manage and processes design

### IV. RESULT ANALYSIS AND SIMULATION OUTPUT

This chapter deals with Simulation and Synthesis results of each individual module and finally, results of the integrated module are depicted. The different modules are designed using Verilog HDL simulated using ISIM simulator.

#### A. RTL Schematics of 128 bit DMC Encoder

The RTL schematic 128 bits DMC Encoder is shown in fig. 7. It has 2 inputs and 2 outputs. The inputs are enable (en) and information bit (D) and outputs are horizontal bits (H) and vertical bits (V). The below RTL schematic shows the sub module of encoder it consists of horizontal redundant bits (H) and vertical redundant bits (V).





### E. Power analysis Report

Total power utilization of signal processing hardware for software defined radio receiver project is 0.036 watt, provided the clock frequency of operation is 100 MHz and the Maximum ambient temperature is 84.2 degree centigrade. The total power utilization of signal processing hardware for software defined radio receiver project is 0.037 watt, provided the clock frequency of operation is 100 MHz and the Maximum ambient temperature is 84.2 degree centigrade

### F. Timing summary

#### 1) 128 Bit DMC encoder Timing Summary

- Speed grade: -3
- Minimum poeriod: No path found
- Minimum input arrival time before clock: No path found
- Maximum output required time after clock: No path found
- Maximum combinational path delay: 14.224ns

#### 2) 128 bit DMC Decoder Timing Summary

- Speed grade: -3
- Minimum poeriod: No path found
- Minimum input arrival time before clock: 16.441ns
- Maximum output required time after clock: 4.118ns
- Maximum combinational path delay: No path found

## V. COMPARATIVE RESULTS

### A. Comparison of Information Bits, correction Proficiency and Redundant Bits of DMC Encoder and Decoder using ERT

Information Bits 64, while  $m=8$ ,  $k=2$  the entire number of correction proficiency is up to 9 bits and and the total number of 68 redundant bits.

Information Bits 128, while  $m=16$ ,  $k=2$  the entire number of correction proficiency is up to 17 bits and the total number of 132 redundant bits.

## VI. CONCLUSION

The Decimal Matrix Code Algorithm is to obtain the maximum error detection and correction bit capability. Decimal Matrix Code is suggested that declare the reliability of memory. Using Decimal algorithm, the protection code word to Detect Errors so that bit reliability is enhanced. The found results showed that the proposed scheme has a superior protection level against large MCUs in memory.

In this thesis, we have successfully design DMC encoder and ERT decoder , which takes 64 bits inputs and maximum error correction capability is upto 9 bits and 128 bits inputs and maximum error correction capability is upto17 bits .we also done comparisons between 64 bits and 128 bits of DMC encoder and ERT Decoder .

The different Modules are designed using Verilog HDL and synthesized using Xilinx integrated software environment (ISE). The design simulated using ISIM Simulator .The design implementation is done on Xilinx Spartan 6 xc6slx45-cgs324-3

### A. Future Work

Future work will be lead for the number of redundant bits can be reduced. Other method will increase number of Detectable and Correctable errors and will decrease the total number of additional bits essential to store to detect the errors.

## 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 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 difference-set 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 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.