INTERNATIONAL JOURNAL OF TECHNO-ENGINEERING
IJTE

ISSN: 2057-5688

# VLSI Implementation of Error Detection and Correction Codes for Space Engineering

**K.RAJASEKHAR[1], R.L.B.R.PRASAD REDDY[2], P.HARI OBULESU[3]**
[1]PG Student, Dept of ECE, SITS, Kadapa, AP, India.
[2]Associate Professor, Dept of ECE, SITS, Kadapa, AP, India.
[3]Assistant Professor, Dept of ECE, SV College of Engineering, AP, India.

***Abstract—*** In this paper, On behalf of technology scaling, on-chip memories in a die undergoes bit errors because of single events or multiple cell upsets by the ecological factors such as cosmic radiation, alpha, neutron particles or due to maximum temperature in space, leads to data corruption. Error detection and correction techniques (ECC) recognize and rectify the corrupted data over communication channel. In this paper, an advanced error correction 2-dimensional code based on divide-symbol is proposed to weaken radiation-induced MCUs in memory for space applications. For encoding data bits, diagonal bits, parity bits and check bits were analyzed by XOR operation. To recover the data, again XOR operation was performed between the encoded bits and the recalculated encoded bits. After analyzing, verification, selection and correction process takes place. The proposed scheme is simulated and synthesized using Xilinx implemented in Verilog HDL.

## I. INTRODUCTION

BINARY information is stored in a storage space called memory. This binary data is stored within metal-oxide semiconductor memory cells on a silicon integrated circuit memory chip. Memory cell is a combination of transistors and capacitors where capacitor charging is considered as 1 and discharging considered as 0 and this can store only one bit. Errors which are temporary or permanent are created in the memory cells and need to be eliminated. Single bit error correction is most commonly used technique which is capable of correcting upto one bit. Since technology is increasing rapidly, there are more probabilities of getting multiple errors. Use of Diagonal Hamming method leads to efficient correction of errors in the memories. Memory was divided as SRAM, DRAM, ROM, PROM, EPROM, EEPROM and flash memory. Main advantages of semiconductor memory are easy to use, less in cost, and have high bits per square micrometers. Temporary errors are called transient errors which are caused because of fluctuations in potential level. Permanent errors are caused because of defects during manufacturing process or large amount of radiations.

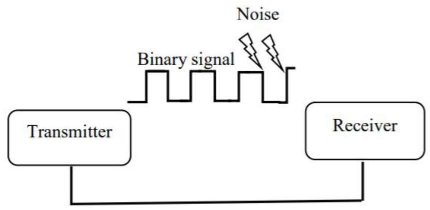



**Figure:** Data Transmission

## II. RELATED WORK

1) Punctured Difference Set (PDS) code is a process to identify the multiple cell upsets in memories
2) The bits by the same logical word into different physical words which has been used in restrain multiple cell upsets in interleaving technique.
3) Built-in current sensors (BICS) are scheduled for reinforcement on correction single error detection and double error detection for granted fortification against multiple cell upsets.
4) 2-Dimentational matrix codes are prospective to conducive correct multiple cell upsets per word with a low decoding delay in which one word has been divided into multiple rows and columb.

**Drawbacks:** 1) PDS codes require more area, power, and delay overheads since the encoding and decoding circuits are more complex in these complicated codes.
2) Interleaving technique may not be practically used in content-addressable memory (CAM), because of the tight coupling of hardware structures from both cells and comparison circuit structures
3) BICS technique can only correct two errors in a word.
4)2D MC is capable of correcting only two errors in all cases. In the recent technique names as FUEC–triple adjacent error correction (TAEC), is able to correct an error in a single bit, or an error in two adjacent bits (2- bit burst errors) or a 3-bit burst error, or it can detect a 4-bit burst error. This is possible by adding one more code bit. In this case, for a 16-bit data word, the FUEC– TAEC code needs eight code bits. The parity-check matrix H for this code is presented. As in the case of the FUEC–DAEC, Ci are the code bits and Xi are the primary data bits. Similarly, from H it is very easy

ISSN: 2057-5688

to design the encoder/decoder circuitry. But this technique will be considered as less precision which could not correct the large number of data's.

## III. HAMMING CODE

Hamming code are the linear block code which are invented by Richard.W. Hamming. They are an improvement over simple parity code method. The simple parity code cannot correct errors, and can only detect an odd number of errors. They are the type of binary codes. The idea of hamming distance is the central concept in coding the error Control. The hamming distance between the two words (of the same size) is the number of differences between the corresponding bits. The hamming distance can easily be found if we apply the Xor operation on the two words and count the number of 1s in the result. The hamming distance is a value always greater than zero. If we find the hamming distance between any two words it will be the result of the Xoring of the two bits. Like the hamming distance between d (000,011) is 2 because 000 xor 011 is 011 (two 1s) and the hamming distance between d (10101, 11110) is 3 because 10101 xor 11110 is 01011(three 1s). [3][7], if we calculate the number of redundancy bits for a 7 bit of information then it comes to be 4 redundancy bit. Redundancy bits are those extra bits which are required to detect and correct errors. The redundancy bits are added to the information bit at the transmitter and removed at the receiver. The receiver is able to detect the error and correct it because of the redundancy bits. Hamming codes are used as forward error correcting codes in the Bluetooth standard, and to protect data stored in semiconductor memories. Hamming codes are generally used in computing, telecommunication, and other applications including data compression, and turbo codes. They are also used for low cost and low power applications.

## IV. METHODOLOGY

The proposed system Design of Diagonal Hamming method for memory Proposed design of Diagonal Hamming based multi-bit error detection and correction technique to the memory is shown in Fig. Using this approach of diagonal Hamming bits, the errors in the message can be recognized and can be rectified which are saved in the memory. In encoding technique message bits are given as input to the Diagonal Hamming encoder and the Hamming bits are calculated. Message and Hamming bits are saved in the memory after the encoding technique.

**a) Encoding Process**

Here input is Data of 16 bits, this 16bits are divided into 4 groups: a,b,c,d

Its Representation

**Data**

| | | | |
|---|---|---|---|
| X[3] | X[2] | X[1] | X[0] |
| X[7] | X[6] | X[5] | X[4] |
| X[11] | X[10] | X[9] | X[8] |
| X[15] | X[14] | X[13] | X[12] |

**Dividing into Groups**

| | | | |
|---|---|---|---|
| a[1] | a[2] | a[3] | a[4] |
| b[1] | b[2] | b[3] | b[4] |
| c[1] | c[2] | c[3] | c[4] |
| d[1] | d[2] | d[3] | c[4] |

**Diagonal Equation**

D[1] = a[1]^b[2]^a[3]^b[4];
D[2] = b[1]^a[2]^b[3]^a[4];
D[3] = c[1]^d[2]^c[3]^d[4];
D[4] = d[1]^c[2]^d[3]^c[4];

**Parity Equation**

P[1] = a[1]^b[1]^c[1]^d[1];
P[2] = a[2]^b[2]^c[2]^d[2];
P[3] = a[3]^b[3]^c[3]^d[3];
P[4] = a[4]^b[4]^c[4]^d[4];

**Check bits Equation**

Ca = {a[1]^a[3],a[2]^a[4]};
Cb = {b[1]^b[3],b[2]^b[4]};
Cc = {c[1]^c[3],c[2]^c[4]};
Cd = {d[1]^d[3],d[2]^d[4]};

The output of encoding is 32bits i.e., original data with Diagonal bits, parity bits, Check bits arranged in matrix order

**Output**

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| O[19] | O[18] | O[17] | O[16] | O[3] | O[2] | O[1] | O[0] |
| O[23] | O[22] | O[21] | O[20] | O[7] | O[6] | O[5] | O[4] |
| O[27] | O[26] | O[25] | O[24] | O[11] | O[10] | O[9] | O[8] |
| O[31] | O[30] | O[29] | O[28] | O[15] | O[14] | O[13] | O[12] |

**Placing the bits**

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| D[1] | P[1] | Ca[1] | ca[2] | X[3] | X[2] | X[1] | X[0] |
| D[2] | P[2] | Cb[1] | Cb[2] | X[7] | X[6] | X[5] | X[4] |
| D[3] | P[3] | Cc[1] | Cc[2] | X[11] | X[10] | X[9] | X[8] |
| D[4] | P[4] | Cd[1] | Cd[2] | X[15] | X[14] | X[13] | X[12] |

**b) Decoding Process**

The encoded data ie., 32bits will be the input From that first 16bits will be having some error That's called Redundancy Bits.

ISSN: 2057-5688

**Input for Decoding**

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| X[19] | X[18] | X[17] | X[16] | | X[3] | X[2] | X[1] | X[0] |
| X[23] | X[22] | X[21] | X[20] | | X[7] | X[6] | X[5] | X[4] |
| X[27] | X[26] | X[25] | X[24] | | X[11] | X[10] | X[9] | X[8] |
| X[31] | X[30] | X[29] | X[28] | | X[15] | X[14] | X[13] | X[12] |

And its pacing of bits Rbits are noting but It may have any Error D,P,C bit are the input What we got from Encoded output.

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| D[1] | P[1] | Ca[1] | ca[2] | | R[3] | R[2] | R[1] | R[0] |
| D[2] | P[2] | Cb[1] | Cb[2] | | R[7] | R[6] | R[5] | R[4] |
| D[3] | P[3] | Cc[1] | Cc[2] | | R[11] | R[10] | R[9] | R[8] |
| D[4] | P[4] | Cd[1] | Cd[2] | | R[15] | R[14] | R[13] | R[12] |

**Data**

| | | | |
|---|---|---|---|
| R[3] | R[2] | R[1] | R[0] |
| R[7] | R[6] | R[5] | R[4] |
| R[11] | R[10] | R[9] | R[8] |
| R[15] | R[14] | R[13] | R[12] |

**Dividing into Groups**

| | | | |
|---|---|---|---|
| a[1] | a[2] | a[3] | a[4] |
| b[1] | b[2] | b[3] | b[4] |
| c[1] | c[2] | c[3] | c[4] |
| d[1] | d[2] | d[3] | c[4] |

**Diagonal Equation**

RD[1] = a[1]^b[2]^a[3]^b[4];
RD[2] = b[1]^a[2]^b[3]^a[4];
RD[3] = c[1]^d[2]^c[3]^d[4];
RD[4] = d[1]^c[2]^d[3]^c[4];

**Parity Equation**

RP[1] = a[1]^b[1]^c[1]^d[1];
RP[2] = a[2]^b[2]^c[2]^d[2];
RP[3] = a[3]^b[3]^c[3]^d[3];
RP[4] = a[4]^b[4]^c[4]^d[4];

**Check bits Equation**

RCa = {a[1]^a[3],a[2]^a[4]};
RCb = {b[1]^b[3],b[2]^b[4]};
RCc = {c[1]^c[3],c[2]^c[4]};
RCd = {d[1]^d[3],d[2]^d[4]};

**Calculate Syndrome bits**

Sd = D xor RD
Sp = P xor RP
Sc = C xor RC

Select the Region From Equation**Sd** ,**Sp**as given above

| Region | Selection criteria |
|---|---|
| Region 1 | SD1+SD2+SP1+SP2 > SD3+SD4+SP3+SP4 |
| Region 2 | SD1+SD2+SP1+SP2 < SD3+SD4+SP3+SP4 |
| Region 3 | SD1+SD2+SP1+SP2 = SD3+SD4+SP3+SP4 |

C orrection Procedure: Correction will be done According with Region

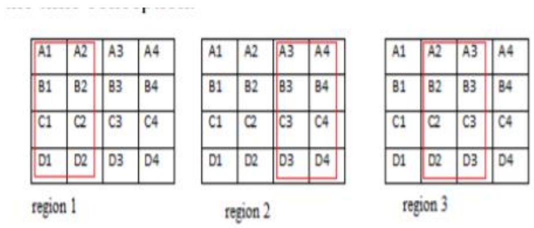



**Figure:** Regions of bits
Once Region is Selected
To get the final Output
The Purticular Region is Xored with SC matrix.

## V. RESULTS

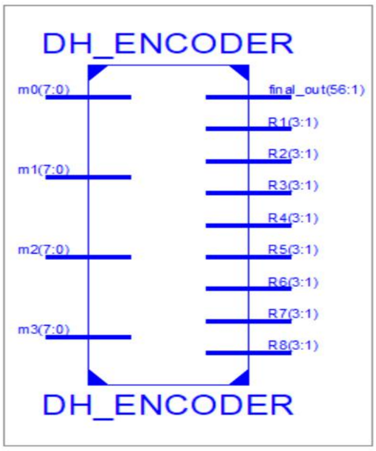
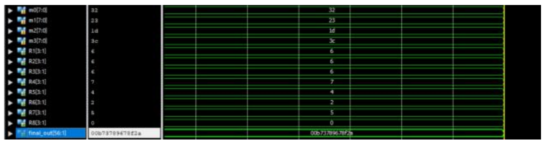



**Figure:** Block diagram for diagonal Hamming Encoder



The input given for this circuit is mo,m1,m2,m3 and output we get parity bit separate order and final_out .

R1,R2,R3,R4,R5,R6,R7,R8 is the parity bits

Final _out = Parity bits + msg_input
= 8 x 3 ( 8 parities each having 3 bits) + 32 bit
= 56 bits

Error Corrector At Reciever side also called Decoder

INTERNATIONAL JOURNAL OF TECHNO-ENGINEERING
IJTE
ISSN: 2057-5688

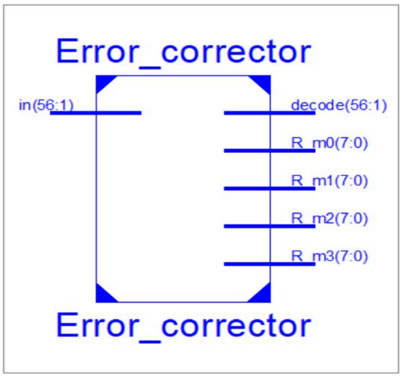



**Figure:** Block Diagram For Decoder

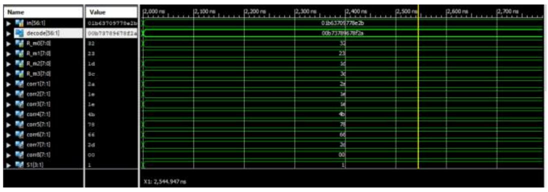


**Figure:** output Waveform for Error corrector

The output we got from encoding block is final_out 00b73789678f2a. Now, this is sent at the receiver side. But at the receiver side suppose the data getting with some Error Consider that as in = 01b63709778e2b input for Error corrector block or Decoding Block and Here at Output Error will be getting corrected and Getting Exact data whatever sent at Transmitter side .
out = 00b73789678f2a.

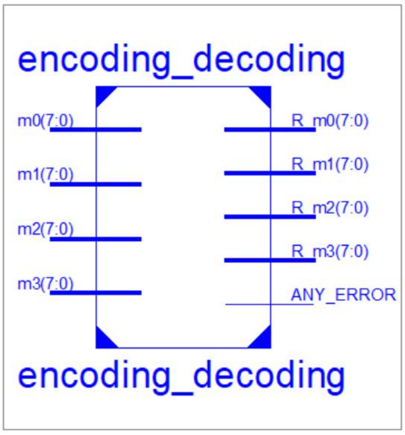
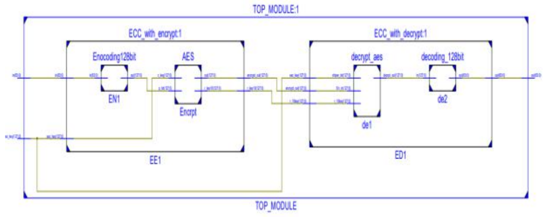



**Figure:** Encoding And Decoding Block

In this the 32bit input is grouped into 8bit msg_data ie.., m0,m1,m2,m3;
Output is R_m0, R_m1, R_m2, R_m3 (Recived_msg)
Output is also indicating Any_error Occured in Matching encoding and Decoding Data.
DIAGONAL HAMMING CODE 32 BIT with Cryptography.



With the help of advance encryption standard ,here error correction codes are designed.
In this Encoded Ouput will be the Input for AES Encryption By using Secured Key.
The Encrypted Output will forwarded to DecryptBlock with same Secured Key and the Decrypt Output will be given for Decoding Which is Decoding the input without any Error and The Required Data is Retrived At Reciever Side.

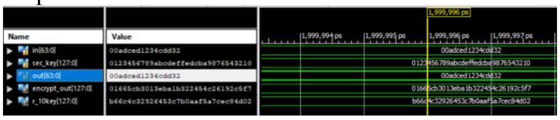


**Figure:** ECC with Cryptography output waveforms

## REFERENCES

[1] Farahat, I.S., Tolba, A.S., Elhoseny, M. and Eladrosy, W., 2019. Data Security and Challenges in Smart Cities. In Security in Smart Cities: Models, Applications, and Challenges. Springer, Cham, pp. 117-142.
[2] Atzori, L., Iera, A. and Morabito, G., 2010. The internet of things: A survey. Computer networks, 54(15), pp.2787-2805.
[3] Zanella, A., Bui, N., Castellani, A., Vangelista, L. and Zorzi, M., 2014. Internet of things for smart cities. IEEE Internet of Things journal, 1(1), pp.22-32.
[4] Kuo, Y.W., Li, C.L., Jhang, J.H. and Lin, S., 2018. Design of a Wireless Sensor Network-Based IoT Platform for Wide Area and Heterogeneous Applications. IEEE Sensors Journal, 18(12), pp.5187- 5197.
[5] Sheng, Z., Yang, S., Yu, Y., Vasilakos, A., Mccann, J. and Leung, K., 2013. A survey on the ietf protocol suite for the internet of things: Standards, challenges, and opportunities. IEEE Wireless Communications, 20(6), pp.91-98.
[6] Gubbi, J., Buyya, R., Marusic, S. and Palaniswami, M., 2013. Internet of Things (IoT): A vision, architectural elements, and future directions. Future generation computer systems, 29(7), pp.1645- 1660.
[7] Hamming, R.W., 1950. Error detecting and error correcting codes. Bell System technical journal, 29(2), pp.147-160.
[8] Hsiao, M.Y., 1970. A class of optimal minimum odd-weight-column SEC-DED codes.

ISSN: 2057-5688

IBM Journal of Research and Development, 14(4), pp.395-401.
[9] Reviriego, P., Liu, S.S., Snchez-Macin, A., Xiao, L. and Maestro, J.A., 2016. Unequal error protection codes derived from SEC-DED codes. Electronics Letters, 52(8), pp.619-620.
[10] Chowdhury, D.R., Gupta, I.S. and Chaudhuri, P.P., 1995. CA-based byte error-correcting code. IEEE Transactions on Computers, 44(3), pp.371-382.
[11] Chaudhuri, P.P., Chowdhury, D.R., Nandi, S. and Chattopadhyay, S., 1997. Additive cellular automata: theory and applications (Vol. 1). John Wiley and Sons. IEEE Computer Society Press, 1997.
[12] Bhaumik, J. and Chowdhury, D.R., 2010. New architectural design of CA-based codec. IEEE transactions on very large scale integration (VLSI) systems, 18(7), pp.1139-1144.
[13] Samanta, J., Bhaumik, J. and Barman, S., 2015. CA-Based Area Optimized Three Bytes Error Detecting Codes. J. Cellular Automata, 10(5-6), pp.409-423.
[14] Samanta, J., Bhaumik, J. and Barman, S., 2018. Compact CABased Single Byte Error Correcting Codec. IEEE Transactions on Computers, (2), pp.291-298.
[15] Cha, S. and Yoon, H., 2012. Efficient implementation of single error correction and double error detection code with check bit pre-computation for memories. JSTS: Journal of Semiconductor Technology and Science, 12(4), pp.418-425.