International Journal For Advanced Research In Science & Technology
A peer reviewed international journal
ISSN: 2457-0362
www.ijarst.in

Convolutional Encoder-Based Error Detection and Correction Techniques for Enhanced Reliability in 5G Communication Systems

K. Shirisha[1],Muddasani Rajkumar[2],Boddu Teja[2],Aleti Himavarshini[2],Bathula Neeraja Yadav[2],Yella Sri Chaitanya[2]

[1]Assistant Professor,[2]UG Student,[1,2]Department of Electronics & Communication Engineering
[1,2]Malla Reddy Engineering College and Management Sciences,Kistapur,Medchal,501401, Telangana

Abstract

The advent of 5G communication systems has ushered in a new era of connectivity, promising unprecedented data speeds and low-latency communication. However, the increasing complexity and susceptibility to errors in these advanced systems necessitate robust error detection and correction mechanisms. This paper introduces a novel approach to enhance the reliability of 5G communication systems through Convolutional Encoder-Based Error Detection and Correction Techniques (MBE-DCC). In the conventional system, error correction relies on traditional methods that may not sufficiently address the unique challenges posed by 5G networks, such as the increased vulnerability to channel impairments and interference. The drawbacks of these conventional systems include suboptimal error correction rates and limited adaptability to the dynamic conditions of 5G environments. The proposed MBE-DCC system employs convolutional encoding to efficiently detect and correct errors, offering a more adaptive and reliable solution. Through extensive simulations and evaluations, the proposed system demonstrates superior error correction performance, making it a promising candidate for enhancing the reliability of 5G communication systems in real-world scenarios.

Keywords: Multi-Bit Error Detection and Correction Codes, encoding, decoding, syndrome decoding, error analysis, error correction modules.

1. Introduction

The space communications contain the channel and to store a value memory is used in channel. In which Random Access Memory (RAM) [1] has been the developing and most compatible device in particular for microprocessor and microcontroller. Main memory helps to keep the processor or controller busy and to perform any task in a minimum time [2]. To achieve this on-chip memory termed as cache memory is used. This type of main memory is developed and placed at different levels close to the processor on same die, additionally to reduce the access time of the processor for the data that to be processed from the main memory (RAM) [3]. Moreover, the data saved in the cache memory need to be the actual that has been stored during the write operation. Nevertheless, stored value need not be the same as that was stored. Probability of change in stored value is increasing linearly because of rapid change in semiconductor processing technology. To overcome this issue of reliability in cache memory CED is imperative [4].

Various works [5,6] proposed three models on memory with CED, of which first model is data communication between memory and cache, second model is the same that includes the additional copy. Last model is the cross-switch communication between the original and additional memory with CED. For all the model's memory designed is a simple single row address memory with CED as a separate block [7]. CED block for the memory is an additional hardware design which increases the delay in area, however additional overhead in area and latency enhances the performance degradation of the system. Performance degradation of a system that includes cache memory can be improved, provided the decoding time for detection [8], correction and the ability to correct or detect with some failure should be a fraction of total memory design size. In recent studies memory with CED is inherent for protecting the memory from errors in particular soft errors. Multiple Cell Upset (MCU) [9] denotes arbitrary number of cells which is repeatedly affected by soft error is random and most of the cases it is not adjacent. However due to scaling down of the gate width of transistors, density of transistors increases in this case occurrence of soft error has the tendency of dissemination. To mitigate this new and optimized coding algorithms are proposed and being proposed on adjacent error also called as burst errors. Most common adjacent errors are two, three termed as Double Adjacent [10], Triple Adjacent [11] respectively. However, the performance degradation of a system cannot be compromised for which optimized syndrome computation is considered. Therefore, the major contributions of this work are as follows:

- Implementation of MBE-DCC for multiple bits error-detection and correction in space communication applications.
- MBE-DCC encoder is developed with generator matrix, which generates the encoded code word.
- Implementation of MBE-DCC decoder with syndrome detection, error location detection, and error correction modules.

- Implementation of syndrome detection is introduced for detecting status of error, which results error presented or absented in encoded data.
- Implementation of error location detection module for identifying the number of error bits with their position.
- Implementation of error correction module for correcting all the errors in encoded data.

Rest of the article is organized as follows: section 2 deals with literature survey, section 3 deals with the proposed MBE-DCC implementation, section 4 deals with analysis of results with performance comparison, section 5 concludes the article with possible future directions.

2. Literature survey

Cosmic rays and IC packaging dye made from radioactive elements are few significant sources of soft error in cache memory. Furthermore, recent studies on cache memories have shown that reliability issues in cache memory for application that are used to store or during processing of data is significant concern. Especially graphical processing units [12] currently developed by NIVIDIA, AMD and INTEL are designed with high bandwidth cache memories and frequently subjected to reliability problems.

To substantive this, in [13] authors presented a flexible CED designed for 32-byte cache memory which also consumes low energy for data fetching. In [14] authors compared different error detection techniques and developed a method which has error guard coverage of 97.9 % using tag in cache. The tags are used for index identification. A limitation is that the adjacent location tag in cache memory may be having the same bits that leads to a faulty read or write operation in the cache memory. The change in bits of tag is due to alpha particles. This can be reduced by deploying CED, SEC – DED, In-Cache Replication (ICR), and in combination of SEC– DED [15], ICR and SEC – DED parity. However, the proposed ICR method has good fault coverage.

In [16] authors proposed counter, shifter, multiplexer and comparator were used and the additional peripherals added to the development of ICR contributed to additional overhead in area. Even after adding the addition peripherals, the delay and area are found to be less. This technique [17] lags because the disadvantage is if the size of the cache increases in terms of level 1, 2, the method proposed increases the complexity of peripherals in 11 proportion to the chance in the size of cache. Also, here only one error is possible to correct. Moreover, triple adjacent location is not addressed.

In [18] authors proposed a method to reduce energy overhead in DRAM (cache) achieved through CED. DRAM is not modified in design instead, for the usual DRAM a new CED that access or decodes only the error word is presented. Conventionally only encoded input data is stored, which is decoded to correct or detect the soft error using CED in a memory. In contrary only error word is corrected using hamming code and error in a word is detected by parity codes. In [19] authors also proposed a DRAM (cache) with CED, unlike other methods [20] in particular proposed a method that detects hard errors using a BIST and build in self repair circuits on a chip. CED that is capable to detect and correct physical fault and soft error is achieved through redundancy circuit or spare circuit and CED respectively. However, study outlined above of DRAM with CED [21] is limited to detect and correct two bits and single bit respectively and correcting soft error beyond one error is not possible. Most widely implemented memory circuit is made to store data in SRAM. There are few applications in which SRAM are neither used as cache nor main memory. For instance, SRAM is used as a configurable switch, programmed to connect the logic blocks, technically termed as routing this architecture [22] is implemented in commercial FPGA‘s such as Xilinx virtex 4 and Altera stratix, moreover SRAM‘s are used as configuration frames, which occupies more than 80% area in FPGA, such as Xilinx virtex 6 and FPGA of Altera family. However, both reconfigurable and configurable frames designed using SRAM are most probably sensitive to soft errors.

In [23] authors proposed a scheme to correct multiple bits upset in configurable frame of FPGA. To detect MBU by combining scrubbing and erasure code additionally to detect error Interleaving-n-Dimensional (InD) method is also proposed. This is also implemented in Virtex-6 XLV240T, which has less overhead in area. Since in proposed InD method [24], with reduced repeated parity bits in all dimensions at regular intervals helps for less area occupation in the SRAM. In [25] authors proposed also presented error correction for switch boxes that are built using SRAM, also mitigation in soft error is achieved through redundancy method. In which zero optimized SRAM are used for interconnection and one optimized SRAM are redundant interconnection termed short and open faults respectively. However major work presented is on optimized routing algorithm.

3. Proposed Method

This section gives the detailed analysis of proposed MBEC method. Figure 1 shows the flowchart of proposed MBEC method. The proposed MBE-DCC is implemented for multiple bits error-detection and correction in space communication applications. Initially, MBE-DCC encoder is developed with generator matrix, which generates the encoded code word. Here, the matrix multiplication is operation is performed that the generator matrix and data input, which generates the code word. Then, the code-word is transmitted in channel of space engineering, where data bits are corrupted by different types of errors and noises. Further, MBE-DCC decoder is developed with syndrome detection, error location detection, and error correction modules. Here, syndrome detection is implemented for detecting status of error, which results error presented or absented in encoded data. Then, error location detection module is introduced for identifying the number of error bits with their position. Then, error correction module is developed for correcting all the errors in encoded data.

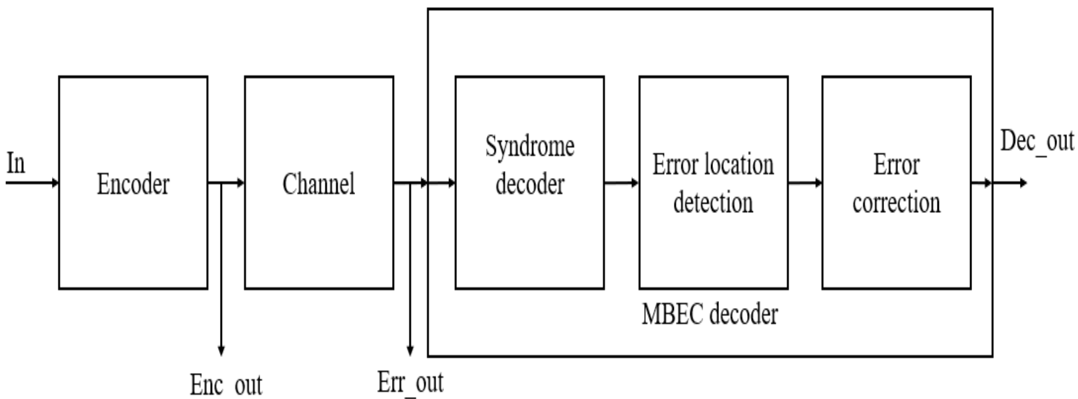


Figure 1.Proposed MBE-DCC flowchart

3.1. MBE-DCC Encoding

The operation of MBE-DCC encoding is achieved by performing the mathematical matrix multiplication between generator matrix and data input.

$$V = DG \tag{1}$$

Here, V is the encoded code word, G is the generator matrix, D is the input data. All of them are binary linear block codes. The process used to design these codes is based on some rules for linear block codes construction. In this paper, the proposed codes are also binary linear block codes and obey similar construction rules. Normally, the binary codes are described by the number of data-bits, k, redundancy bits, (n − k), and the block size of the encoded-word, n. An (n, k) code is defined by its generator matrix G or parity check matrix H in

$$G = [P_{k\times(n-k)} \cdot I_{k\times k}] \quad H = [P^T \cdot I_{(n-k)}] \tag{2}$$

International Journal For Advanced Research
In Science & Technology
A peer reviewed international journal
ISSN: 2457-0362
www.ijarst.in

where $I_{k\times k}$ is the identity matrix, P is the matrix with size k × (n − k), and P^T is the transpose of P. In the encoding process, the generator matrix G is used to encode the data bits through the process.

Table 1. Construction of generator matrix.

C 0	C 1	C 2	C 3	C 4	C 5	C 6	C 7	C 8	C 9	C 10	C 11	C 12	C 13	C 14	C 15	C 16	C 17	C 18	C 19	C 20	C 21	C 22
1	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	1	0	0	1	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	1	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
1	1	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
0	1	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	1	1	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
0	1	1	0	1	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
1	1	0	1	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
0	1	1	0	0	1	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
0	0	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
0	0	1	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Table 1 shows the final constructed generator Matrix, which contains identity matrix and parity bits. Here, parity bits contain size as 16x7, which are ranged from all rows with C0-C6 columns. Further, the size of identity bits contains size as 16x16 i.e., all rows with C7-C22 columns. The size of data is 16 bits as follows

$$D = [D1, D2, D3, D4, D5, D6, D7, D8, D9, D10, D11, D12, D13, D14, D15, D16] \quad (3)$$

Finally, encoding operation is achieved through calculation of check bits (C1 to C7) as follows:

$$C1 = D1 \oplus D4 \oplus D6 \oplus D8 \oplus D9 \oplus D10 \oplus D14 \quad (4)$$

$$C2 = D2 \oplus D4 \oplus D5 \oplus D7 \oplus D8 \oplus D11 \oplus D15 \quad (5)$$

$$C3 = D3 \oplus D7 \oplus D11 \oplus D13 \oplus D16 \oplus D10 \quad (6)$$

$$C4 = D1 \oplus D4 \oplus D8 \oplus D10 \oplus D12 \oplus D13 \quad (7)$$

International Journal For Advanced Research In Science & Technology
A peer reviewed international journal
www.ijarst.in
ISSN: 2457-0362

$$C5 = D2 \oplus D5 \oplus D6 \oplus D7 \oplus D8 \oplus D13 \oplus D14 \quad (9)$$

$$C6 = D2 \oplus D6 \oplus D7 \oplus D11 \oplus D13 \oplus D16 \quad (10)$$

$$C7 = D3 \oplus D6 \oplus D9 \oplus D11 \oplus D12 \oplus D13 \oplus D15 \oplus D16 \quad (11)$$

Finally, encoding operation is achieved as follows:

$$V = \begin{bmatrix} C1, C2, C3, C4, C5, C6, C7, D1, D2, D3, D4, D5, D6, D7, D8, D9, \\ D10, D11, D12, D13, D14, D15, D16 \end{bmatrix} \quad (12)$$

3.2 MBE-DCC Decoding

The MBE-DCC decoding is consisting of syndrome detection, error location detection and error correction stages. The encoded output V is transmitted into space communication channel, where different types of noises were added.

$$R = V + E$$

$$RC = [C1, C2, C3, C4, C5, C6, C7] + [E1, E2, E3, E4, E5, E6, E7]$$

$$RD = [D1, D2, D3, D4, D5, D6, D7, D8, D9, D10, D11, D12, D13, D14, D15, D16] + [E8, E9, E10, E11, E12, E13, E14, E15, E16, E17, E18, E19, E20, DE21, E22, E23]$$

$$R = [RC, RD]$$

$$R = \begin{bmatrix} RC1, RC2, RC3, RC4, RC5, RC6, RC7, RD1, RD2, RD3, RD4, RD5, RD6, RD7, RD8, RD9, \\ RD10, RD11, RD12, RD13, RD14, RD15, RD16 \end{bmatrix}$$

Here, V is the encoded code word, which stores into memory or transmitted into channel. Further, E represents the error and R represents the received vector with error.

$$S = R.H^T \quad (6)$$

Here, S represents the syndrome value, H represents the parity check matrix and it is constructed from generator matrix as shown in Table 2. The size of identity bits (I) are 7x7 i.e., all rows with H1-H7 columns and size of the parity bits (P) are 7x16 i.e., all rows with H8-H23 columns. Further, if the S is zero, which indicates no errors in received data. Further, if the S is not equal to zero, which indicates errors presented in received data.

Table 2. Construction of parity check matrix.

H1	H2	H3	H4	H5	H6	H7	H8	H9	H10	H11	H12	H13	H14	H15	H16	H17	H18	H19	H20	H21	H22	H23
1	0	0	0	0	0	0	1	0	0	1	0	1	0	1	1	1	0	0	0	1	0	0
0	1	0	0	0	0	0	0	1	0	1	1	0	1	1	0	0	1	0	0	0	1	0
0	0	1	0	0	0	0	0	0	1	0	0	0	1	0	0	1	1	0	1	0	0	1
0	0	0	1	0	0	0	1	0	0	1	0	0	0	1	0	1	0	1	1	0	0	0
0	0	0	0	1	0	0	0	1	0	0	1	1	1	1	0	0	0	0	1	1	0	0
0	0	0	0	0	1	0	0	1	0	0	0	1	1	0	0	0	1	0	1	0	0	1
0	0	0	0	0	0	1	0	0	1	0	0	1	0	0	1	0	1	1	1	0	1	1
RC1	RC2	RC3	RC4	RC5	RC6	RC7	RD1	RD2	RD3	RD4	RD5	RD6	RD7	RD8	RD9	RD10	RD11	RD12	RD13	RD14	RD15	RD16

Finally, the syndrome calculation is simplified as follows:

$$S1 = RC1 \oplus RD1 \oplus RD4 \oplus RD6 \oplus RD8 \oplus RD9 \oplus RD10 \oplus RD14$$

$$S2 = RC2 \oplus RD2 \oplus RD4 \oplus RD5 \oplus RD7 \oplus RD8 \oplus RD11 \oplus RD15$$

$$S3 = RC3 \oplus RD3 \oplus RD7 \oplus RD11 \oplus RD13 \oplus RD16 \oplus RD10$$

$$S4 = RC4 \oplus RD1 \oplus RD4 \oplus RD8 \oplus RD10 \oplus RD12 \oplus RD13$$

$$S5 = RC5 \oplus RD2 \oplus RD5 \oplus RD6 \oplus RD7 \oplus RD8 \oplus RD13 \oplus RD14$$

$$S6 = RC6 \oplus RD2 \oplus RD6 \oplus RD7 \oplus RD11 \oplus RD13 \oplus RD16$$

$$S7 = RC7 \oplus RD3 \oplus RD6 \oplus RD9 \oplus RD11 \oplus RD12 \oplus RD13 \oplus RD15 \oplus RD16$$

Then, based on syndrome values error locations were identified using bit pattern matching process. It is explained by following example.

$$D = [1,1,0,1,1,1,0,0,1,1,0,0,1,1,1,1]$$

Then, C values are calculated using Equations (4)-(11) and resulted as follows:

International Journal For Advanced Research In Science & Technology
A peer reviewed international journal
ISSN: 2457-0362
www.ijarst.in

$$C = [0,0,1,0,1,0,1]$$

Then, encoded codeword V becomes,

$$V = [0,0,1,0,1,0,1,1,1,0,1,1,1,0,0,1,1,0,0,1,1,1,1]$$

Consider 4 bits are corrupted and error occurred in [D2, D3, D4, D5] positions of V. Then, R becomes

$$R = [0,0,1,0,1,0,1,1,0,1,0,0,1,0,0,1,1,0,0,1,1,1,1]$$

The error locations are identified performing all combinations of XOR in H-matrix columns and the XOR outcome will be matched with syndrome for anyone of the combination. The process is repeated until the syndrome is matched. Table 3 illustrates the error location identification process.

Table 2. Error location identification.

H9	H10	H11	H12	XOR (H9, H10, H11, H12)	S
0	0	1	0	1	1
1	0	1	1	1	1
0	1	0	0	1	1
0	0	1	0	1	1
1	0	0	1	0	0
1	0	0	0	1	1
0	1	0	0	1	1

The syndrome is matched with the XOR (H9, H10, H11, H12) combination according to Table 3, so error locations become $[D2, RD3, RD4, RD5]$. The error occurred positions in R vector is $[RD2, RD3, RD4, RD5]$, which are equivalent to H vector positions as [H9, H10, H11, H12]. Finally, error correction operation is implemented by performing the complement of error corrected bits. The resultant error corrected outcome is obtained as follows

$$Out = [0,0,1,0,1,0,1,1,1,0,1,1,1,0,0,1,1,0,0,1,1,1,1]$$

4. Results and discussions

Xilinx ISE software was used to create all of the MBE-DCC designs. This software programmed gives two types of outputs: simulation and synthesis. The simulation results provide a thorough examination of the MBE-DCC architecture in terms of input and output byte level combinations. Decoding procedure approximated simply by applying numerous combinations of inputs and monitoring various outputs through simulated study of encoding correctness. The use of area in relation to the transistor count will be accomplished as a result of the synthesis findings. In addition, a time summary will be obtained with regard to various path delays, and a power summary will be prepared utilizing the static and dynamic power consumption.

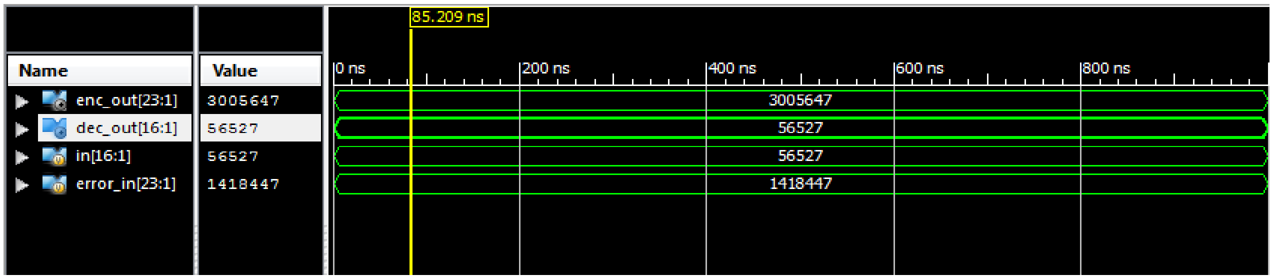


Figure 2. Simulation outcome of MBE-DCC.

Figure 5 represents the simulation outcome of MBE-DCC. Here, data denotes the initial input (in), error_in is the manual error input, and enc_out denotes the encoded operand as a whole. The dec_out is the decoded output data that was error-free and is identical to the input data.

Device Utilization Summary (estimated values)			[-]
Logic Utilization	**Used**	**Available**	**Utilization**
Number of Slice LUTs	55	17600	0%
Number of fully used LUT-FF pairs	0	55	0%
Number of bonded IOBs	77	100	77%

Figure 3. Design summary.

Figure 3 shows the design (area) summary of proposed method. Here, the proposed method utilizes the low area in terms of slice LUTs i.e., 55 out of available 17600.

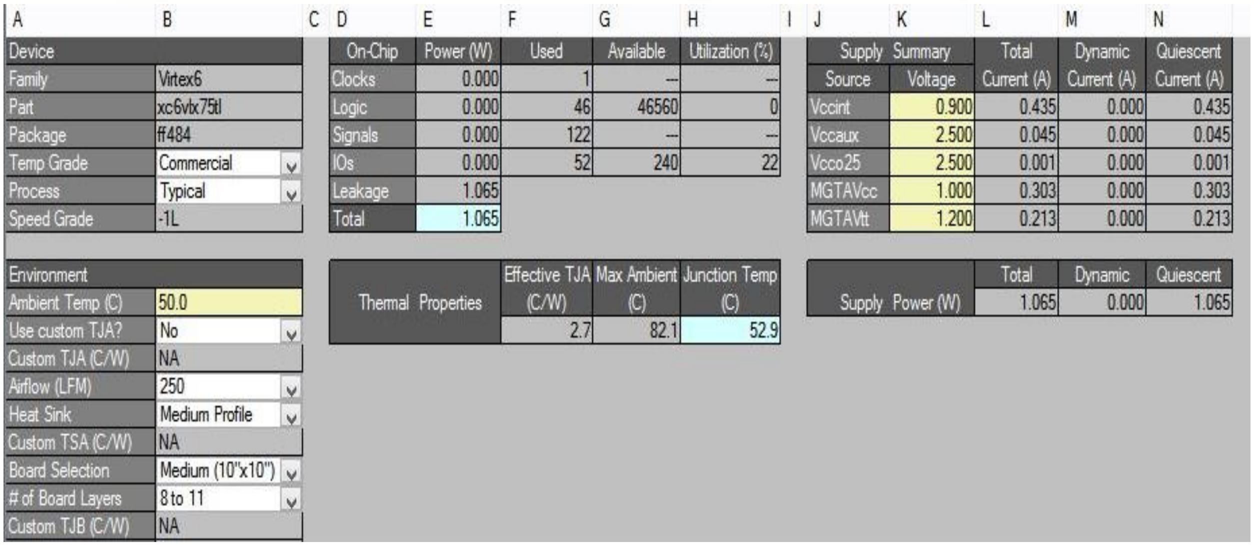
```
Data Path: in<11> to D1/dec_out_11
                              Gate     Net
  Cell:in->out      fanout   Delay   Delay  Logical Name (Net Name)
  ----------------------------------------  ------------
   IBUF:I->O            10   0.000   0.321  in_11_IBUF (in_11_IBUF)
   LD:D                     -0.035          D1/dec_out_11
  ----------------------------------------
  Total                      0.321ns (0.000ns logic, 0.321ns route)
                                     (0.0% logic, 100.0% route)

=========================================================================
```

Figure 4. Time summary

Figure 4 shows the time summary of proposed method. Here, the proposed method consumed total 0.321ns of time delay, which is entirely route delay.

International Journal For Advanced Research In Science & Technology
A peer reviewed international journal
www.ijarst.in
ISSN: 2457-0362



Device	
Family	Virtex6
Part	xc6vlx75tl
Package	ff484
Temp Grade	Commercial
Process	Typical
Speed Grade	-1L

On-Chip	Power (W)	Used	Available	Utilization (%)
Clocks	0.000	1	---	---
Logic	0.000	46	46560	0
Signals	0.000	122	---	---
IOs	0.000	52	240	22
Leakage	1.065			
Total	1.065			

Supply Source	Summary Voltage	Total Current (A)	Dynamic Current (A)	Quiescent Current (A)
Vccint	0.900	0.435	0.000	0.435
Vccaux	2.500	0.045	0.000	0.045
Vcco25	2.500	0.001	0.000	0.001
MGTAVcc	1.000	0.303	0.000	0.303
MGTAVtt	1.200	0.213	0.000	0.213

Environment	
Ambient Temp (C)	50.0
Use custom TJA?	No
Custom TJA (C/W)	NA
Airflow (LFM)	250
Heat Sink	Medium Profile
Custom TSA (C/W)	NA
Board Selection	Medium (10"x10")
# of Board Layers	8 to 11
Custom TJB (C/W)	NA

Thermal Properties	Effective TJA (C/W)	Max Ambient (C)	Junction Temp (C)
	2.7	82.1	52.9

	Total	Dynamic	Quiescent
Supply Power (W)	1.065	0.000	1.065

Figure 5. Power summary.

Figure 5 shows the power consumption report of proposed MBE-DCC. Here, the proposed method consumed power as 1.065 watts. Table 4 compares the performance evaluation of various MBE-DCC approches. Here, the propsoed MBE-DCC resulted in superior (reduced) performance in terms of LUTs, time-delay, and power consumption as compared to conventional approaches such as LBC [22], STBC [23], and Turbo [24].

Table 4. Performance evaluation.

Metric	**LBC [22]**	**STBC [23]**	**Turbo [24]**	**Proposed MBE-DCC**
LUTs	78	72	64	55
Time delay (ns)	3.28	2.284	1.453	0.321
Power consumption (w)	3.45	2.34	1.79	1.065

5. Conclusion

The MBE-DCC for multiple bits error detection and correction is implemented in this work. The initial implementation of MBE-DCC encoding employs a generator matrix that has both identity bits and parity bits. Then, encoded data that has been corrupted by various noises and errors is transmitted into the space communication channel. Thus, using error location detection, syndrome detection, and error correction modules, the MBE-DCC decoding operation was carried out at the receiver side of space communications. The simulations showed that the proposed MBE-DCC

International Journal For Advanced Research In Science & Technology
A peer reviewed international journal
ISSN: 2457-0362
www.ijarst.in

performed better than traditional CED techniques. This work can be extended with advanced multiple bit error detection and correction codes for real time applications.

References:

[1]. Somashekhar, Vikas Maheshwari, and R. P. Singh. "A Study of Fault Tolerance In High Speed VLSI Circuits." *International Journal Of Scientific & Technology Research* 8.08 (2019).

[2]. Palchaudhuri, Ayan, and Anindya Sundar Dhar. "Speed-area optimized VLSI architecture of multi-bit cellular automaton cell based random number generator on FPGA with testable logic support." *Journal of Parallel and Distributed Computing* 151 (2021): 13-23.

[3]. Sharma, Vishal, et al. "A reliable, multi-bit error tolerant 11T SRAM memory design for wireless sensor nodes." *Analog Integrated Circuits and Signal Processing* 107.2 (2021): 339-352.

[4]. Tavakkolai, Hamid, Gholamreza Ardeshir, and Yasser Baleghi. "Fast adder with the ability of multiple faults detection and correction." *International Journal of Nonlinear Analysis and Applications* 12.Special Issue (2021): 937-950.

[5]. Maiti, Swapan, Meghna Sengupta, and Dipanwita Roy Chowdhury. "Generating nonlinear codes for multi-bit symbol error correction using cellular automata." *Physica D: Nonlinear Phenomena* 415 (2021): 132758.

[6]. Abbas, S. M., Tonnellier, T., Ercan, F., & Gross, W. J. (2020, October). High-throughput VLSI architecture for GRAND. In *2020 IEEE Workshop on Signal Processing Systems (SiPS)* (pp. 1-6). IEEE.

[7]. Sai, G. Manoj, et al. "Diagonal hamming based multi-bit error detection and correction technique for memories." *2020 International Conference on Communication and Signal Processing (ICCSP)*. IEEE, 2020.

[8]. Satyanarayana, Telugu, Vaseen Ahmed Qureshi, and G. Divya. "Design and implementation of error detection and correction system for semiconductor memory applications." (2022): 325-330.

[9]. Flayyih, Wameedh Nazar. "Crosstalk aware multi-bit error detection with limited error correction coding for reliable on-chip communication." *International Journal of Computer Applications* 179.40 (2018).

[10]. Tripathi, S., Jana, J., Samanta, J., Raj, A., Ranjan, D., & Singh, M. P. (2020). Design and evaluation of neale-based multi-bit adjacent error-correcting codec for sram. In *Proceedings of the 2nd International Conference on Communication, Devices and Computing* (pp. 259-268). Springer, Singapore.

[11]. Sharma, V., Bisht, P., Dalal, A., Gopal, M., Vishvakarma, S. K., & Chouhan, S. S. (2019). Half-select free bit-line sharing 12T SRAM with double-adjacent bits soft error correction and a reconfigurable FPGA for low-power applications. *AEU-International Journal of Electronics and Communications*, *104*, 10-22.

[12]. Reviriego, Pedro, Salvatore Pontarelli, and Anees Ullah. "Error detection and correction in SRAM emulated TCAMs." *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* 27.2 (2018): 486-490.

[13]. Tajasob, Sarvenaz, Morteza Rezaalipour, and Masoud Dehyadegari. "Designing energy-efficient imprecise adders with multi-bit approximation." *Microelectronics Journal* 89 (2019): 41-55.

[14]. Mandal, Swagata, et al. "Criticality aware soft error mitigation in the configuration memory of SRAM based FPGA." *2019 32nd International Conference on VLSI Design and 2019 18th International Conference on Embedded Systems (VLSID)*. IEEE, 2019.

[15]. Mandal, Swagata, et al. "Criticality aware soft error mitigation in the configuration memory of SRAM based FPGA." *2019 32nd International Conference on VLSI Design and 2019 18th International Conference on Embedded Systems (VLSID)*. IEEE, 2019.

[16]. Sengupta, M., & Chowdhury, D. R. (2019). Generating Non-linear Codes for Multi-bit Symbol Error Correction using Cellular Automata. *Booklet of abstracts*, 42.

[17]. Sayed, Nour, Rajendra Bishnoi, and Mehdi B. Tahoori. "Fast and reliable STT-MRAM using nonuniform and adaptive error detecting and correcting scheme." *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* 27.6 (2019): 1329-1342.

[18]. Chabot, Alexandre, et al. "A memory reliability enhancement technique for multi bit upsets." *Journal of Signal Processing Systems* 93.4 (2021): 439-459.

[19]. Tripathi, S., Maity, R. K., Jana, J., Samanta, J., & Bhaumik, J. (2020). FPGA based low area multi-bit adjacent error correcting codec for SRAM application. *Radioelectronics and Communications Systems*, *63*(10), 543-552.

[20]. Jain, A., Veggetti, A. M., Crippa, D., Benfante, A., Gerardin, S., & Bagatin, M. (2022). Radiation Tolerant Multi-Bit Flip-Flop System With Embedded Timing Pre-Error Sensing. *IEEE Journal of Solid-State Circuits*.

[21]. Liu, L., Zhuang, Y., Zhang, L., Tang, H., & Dong, S. (2018). Proactive correction coset decoding scheme based on SEC-DED code for multibit asymmetric errors in STT-MRAM. *Microelectronics journal*, *82*, 92-100.

[22]. Li, T., Liu, H., & Yang, H. (2019). Design and characterization of SEU hardened circuits for SRAM-based FPGA. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, *27*(6), 1276-1283.

[23]. Kavitha, S., et al. "Energy Efficient, Hamming Code Technique for Error Detection/Correction Using In-Memory Computation." *2021 25th International Symposium on VLSI Design and Test (VDAT)*. IEEE, 2021.

[24]. Liu, Shanshan, Pedro Reviriego, and Fabrizio Lombardi. "Detection of limited magnitude errors in emerging multilevel cell memories by one-bit parity (OBP) or two-bit parity (TBP)." *IEEE Transactions on Emerging Topics in Computing* 9.4 (2019): 1792-1802.

[25]. Somashekhar, Vikas Maheshwari, and R. P. Singh. "A Study of Fault Tolerance In High Speed VLSI Circuits." *International Journal Of Scientific & Technology Research* 8.08 (2019).