IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 11, November 2025

DOI: 10.17148/IJARCCE.2025.1411146

# FPGA IMPLEMENTATION OF ADVANCED ERROR DETECTION AND CORRECTION TECHNIQUES FOR MULTI CELL MEMORIES

**Prof. Rohith H S[1], Asif S Nadaf[2], Suraj S R[3], Venkatesh R[4], MD Farhan Kotur[5]**

Assistant Professor, ECE, East West Institute Of Technology, Bengaluru, India[1]

Student, ECE, East West Institute Of Technology, Bengaluru, India[2]

Student, ECE, East West Institute Of Technology, Bengaluru, India[3]

Student, ECE, East West Institute Of Technology, Bengaluru, India[4]

Student, ECE, East West Institute Of Technology, Bengaluru, India[5]

**Abstract**: Modern non-volatile memories, especially Multi-Level Cell (MLC) Flash and ReRAM, suffer from asymmetric and limited-magnitude errors that significantly degrade reliability. Traditional SEC–DED codes fail to correct multi-level and adjacent magnitude errors efficiently. This paper presents the design and FPGA implementation of Limited-Magnitude Error Correction Codes (LM-ECC) for MLC memory systems. The architecture includes SL-SEC, ML-SEC, and IP-SEC-DAEC designs synthesized on Xilinx FPGA. Comparative results demonstrate that the proposed design achieves lower area, reduced power consumption, and improved error-correction efficiency compared to existing techniques. The implementation is validated using Xilinx Vivado with optimized power, timing, and area metrics.

**Keywords**: MLC Memory, Error Correction Codes, Limited Magnitude Errors, SEC-DAEC, FPGA Implementation, Xilinx, Digital Design.

## I. INTRODUCTION

Multi-Level Cell (MLC) memory technology has become a key enabler in the evolution of modern digital storage systems. Unlike traditional Single-Level Cell (SLC) memory, which stores only one bit per cell, MLC memories can store two or more bits per cell by using multiple threshold voltage levels. This significantly increases storage density while reducing cost per bit, making MLC suitable for consumer electronics, solid-state drives, mobile devices, embedded systems, and large-scale data centers. However, this increased efficiency comes at the cost of reduced reliability. As the voltage margins between the levels become narrower, MLC memories become more sensitive to noise, environmental conditions, and device-level variations, thereby generating complex error patterns that traditional error correction mechanisms cannot handle efficiently.

Error generation in MLC memory is fundamentally different from the random single-bit errors observed in earlier memory technologies. MLC devices often experience *magnitude-limited errors*, where the stored charge drifts slightly but remains within a predictable bounded range due to phenomena such as inter-cell interference (ICI), charge leakage over time, program/erase degradation, and temperature-dependent variations. In addition, spatially correlated errors—such as adjacent-cell disturbances—remain common because programming one cell often influences the threshold voltage of neighboring cells. These multi-bit and adjacent-bit errors degrade the reliability of stored data and pose severe challenges for conventional ECC schemes like simple parity codes, Hamming codes, BCH, and Reed-Solomon codes, which were primarily designed to correct isolated, random bit flips rather than structured, magnitude-limited faults.

To address this reliability gap, the current project focuses on designing and implementing specialized Error Correction Codes (ECC) tailored for MLC memory behavior. The work introduces three hardware-optimized ECC schemes—**Single-Level Single Error Correction (SL-SEC)**, **Multi-Level Single Error Correction (ML-SEC)**, and **Interleaved Parity Single-Error Correction with Double Adjacent Error Correction (IP-SEC-DAEC)**. These codes are mathematically designed to capture the predictable nature of magnitude-limited errors and adjacent-cell failures in MLC memory. The encoding strategies use structured parity equations, and the decoding layer employs syndrome-based detection to precisely identify both the error location and its magnitude.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 11, November 2025

DOI: 10.17148/IJARCCE.2025.1411146

All three ECC schemes are implemented using **Verilog HDL** on a **Spartan-6 FPGA**, enabling real-time evaluation of their area, delay, and power performance. The FPGA platform allows the design to be modular, scalable, and efficient, making it suitable for integration into real-world memory systems. Simulation using testbenches verifies the correction of different error magnitudes, while hardware synthesis reports provide insights into LUT count, flip-flop utilization, timing closure, and overall power consumption. Results demonstrate that the proposed ECC architectures provide improved correction capability with significantly lower hardware cost compared to baseline implementations, validating their usefulness in future high-density memory products.

This project therefore addresses a critical need for robust, low-complexity, and highly effective ECC solutions specifically designed for MLC environments. By combining theoretical ECC design with practical FPGA implementation, the work contributes to next-generation memory reliability engineering and lays the foundation for further innovations in emerging non-volatile memory technologies such as ReRAM, PCM, and STT-RAM.

## II. METHODOLOGY

This methodology consit of :

- Collect input data from the MLC memory block.
- Feed the data into the encoder module.
- Generate parity bits and form the encoded codeword.
- Send the encoded data into the FPGA processing block.
- Apply controlled fault patterns using the error-injection module.
- Pass the error-injected data to the FPGA for testing.
- Process the data through SL-SEC and ML-SEC correction modules inside the FPGA.
- Process the data through IP-SEC and DAEC correction modules for double-adjacent error correction.
- Use the internal encoding logic for parity generation and codeword support.
- Use the internal decoding logic for syndrome generation and error-location identification.
- Correct the detected error and produce final corrected output through the decoder.
- Use control logic to manage data flow, block selection, ECC mode, and timing operations.
- Observe the FPGA outputs and verify correct behavior across all ECC schemes.

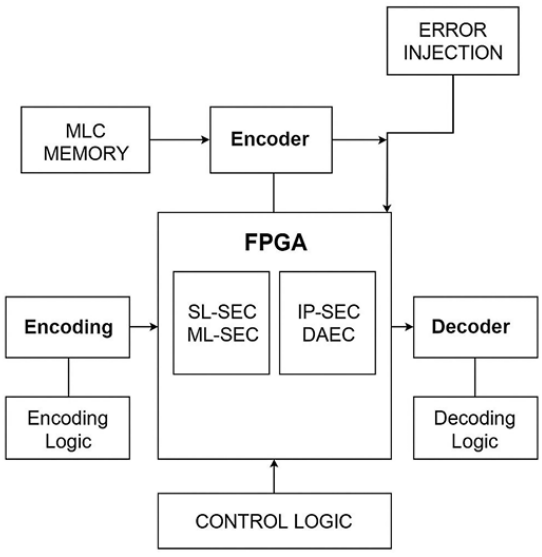


Fig 1 Block diagram of Wearable-Based Kinematic analysis of Cricket Bowling

- The system begins by receiving multi-level data from the MLC memory, where each cell may contain magnitude-limited or adjacent-pattern errors commonly observed in high-density flash technologies.
- The encoder processes the input by generating vertical and horizontal parity bits, forming structured codewords required for detecting and correcting MLC-specific errors.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 11, November 2025

DOI: 10.17148/IJARCCE.2025.1411146

- An error-injection module introduces controlled faults, including magnitude-1, magnitude-2, magnitude-3, and double-adjacent disturbances, enabling systematic evaluation of each ECC technique under realistic error conditions.
- The encoded and error-injected data is passed into the FPGA, where multiple ECC architectures—SL-SEC, ML-SEC, IP-SEC, and DAEC—are implemented in hardware for real-time error detection and correction.
- The encoding logic inside the FPGA supports codeword formation and parity computation, while the decoding logic performs syndrome generation, error-location identification, and magnitude estimation.
- The FPGA's control logic coordinates data flow, selects the active ECC scheme, and manages timing operations to ensure synchronized and efficient processing of memory data.
- The decoder block outputs the corrected data, which is compared with the original input to verify accuracy, evaluate correction capability, and measure performance improvements over baseline ECC designs.

### 2.1 CIRCUIT DIAGRAM

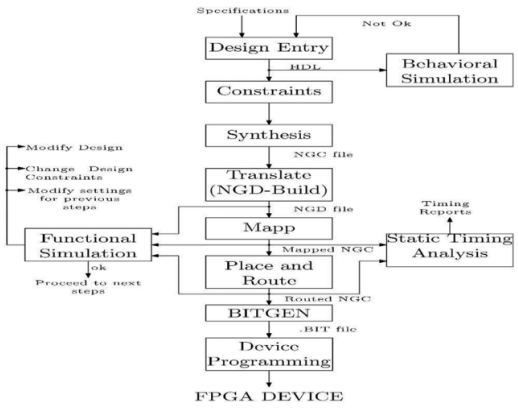


Fig 2.2 Circuit Design flow

The FPGA design flow for the MLC memory error detection and correction system begins with HDL-based design entry, where all modules—including the encoder, decoder, error-injection unit, and ECC blocks such as SL-SEC, ML-SEC, and IP-SEC-DAEC—are developed according to the error characteristics observed in MLC memory. Design constraints, including timing requirements, device selection, and input/output pin mappings, are applied to ensure reliable and deterministic system behavior. During synthesis, the Verilog modules are translated into an optimized gate-level netlist suitable for parity generation, syndrome computation, and correction logic. In the translate stage, the synthesized netlist is merged with the constraints to produce an NGD file, which is subsequently mapped onto available FPGA resources such as LUTs, flip-flops, and BRAM blocks.

During the place-and-route phase, all ECC components are arranged within the FPGA fabric, and routing paths are created to support high-speed operation required for magnitude-limited and adjacent-error correction. Static timing analysis verifies that critical paths—such as vertical parity computation, syndrome decoding, and error-magnitude evaluation—meet the required timing constraints.

Functional simulation is then performed by applying multiple test patterns, injecting controlled magnitude-based faults, and observing the corrected outputs to validate the behavior of the complete ECC system. Once the design passes all functional and timing checks, the BITGEN tool generates the final configuration bitstream. This bitstream is then downloaded onto the Spartan-6 FPGA, enabling real-time implementation of the MLC memory error detection and correction architecture on hardware.

## III. IMPLEMENTATIONS

### 3.1 SYSTEM ARCHITECTURE

The system consists of an FPGA-based error-correction framework designed to handle magnitude-limited and double-adjacent errors in Multi-Level Cell (MLC) memory. It integrates an encoder, MLC memory model, error-injection module, and multiple ECC decoders (SL-SEC, ML-SEC, IP-SEC-DAEC). All modules are implemented using Verilog HDL and connected through synchronous data paths. The FPGA processes raw data, generates parity bits, injects controlled errors, and performs real-time correction through a syndrome-based decoding pipeline.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 11, November 2025

DOI: 10.17148/IJARCCE.2025.1411146

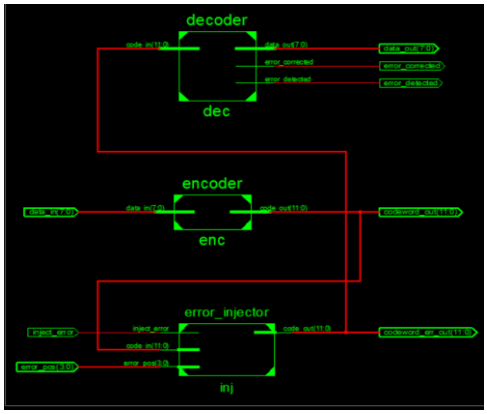


Figure 3.1: Top-Level RTL Schematic

### 3.2 ENCODER, MEMORY MODEL & ERROR INJECTION DESIGN

#### [1] Encoder Module

• Takes 8-bit input data and produces an 11-bit encoded codeword.
  • Generates vertical and horizontal parity bits using XOR-based logic.
  • Ensures low propagation delay suitable for real-time MLC memory protection.

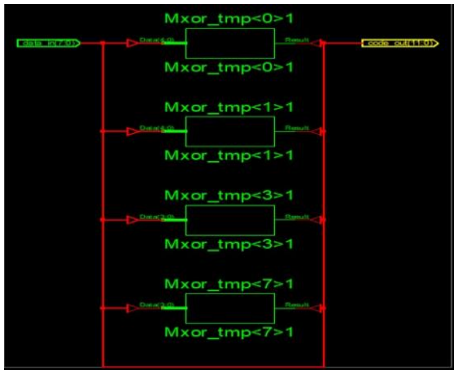


Figure 3.2: Encoder RTL Schematic

#### [2] MLC Memory Model

• Simulates multi-level cell charge states and threshold levels.
  • Imitates real MLC drift and neighboring cell interference.
  • Stores encoded data for error testing.

#### [3] Error Injection Module

• Injects magnitude-1, 2, 3 errors and double-adjacent faults.
  • Selects error position and magnitude through control ports.
  • Replicates realistic MLC memory failures for ECC evaluation.

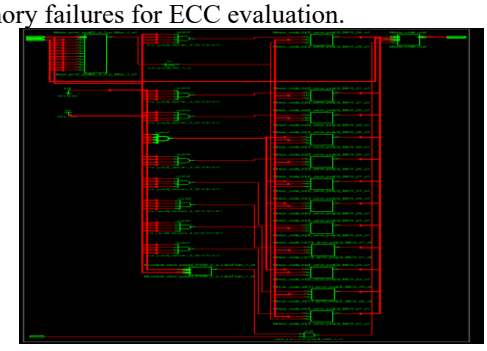

Figure 3.3: Error Injector RTL Schematic

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 11, November 2025

DOI: 10.17148/IJARCCE.2025.1411146

### 3.3 DATA PROCESSING FLOW

- Raw 8-bit data is applied to the encoder.
- The encoder generates an 11-bit protected codeword.
- The codeword is stored in the MLC memory model.
- The error-injection module applies controlled faults.
- Decoder receives the corrupted data and computes syndrome signals.
- Corrected data is obtained and validated.

This ensures full evaluation of MLC memory error-correction capability.

### 3.4 ECC DECODING PIPELINE

**[4] Syndrome Formation & Error Analysis**

• Parity is recomputed to form syndrome bits.
• Syndrome identifies both location and magnitude of the error.

**[5] ECC Schemes Implemented**

• SL-SEC – Corrects single-bit single-level errors.
• ML-SEC – Corrects magnitude-limited errors
• IP-SEC – Handles interleaved parity single-error correction.
• DAEC – Corrects double-adjacent error patterns.

**[6] Correction Output**

• Reconstructed 8-bit data
• Error detected signal
• Error corrected signal

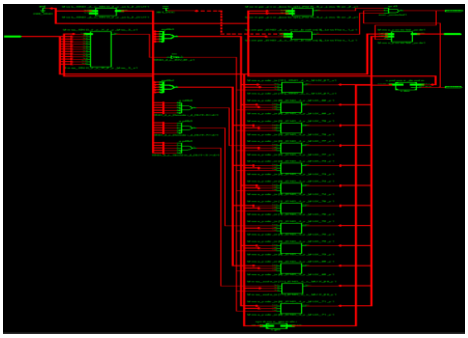

Figure 3.4: Decoder RTL Schematic

### 3.5 RTL FUNCTIONAL OPERATION

The RTL simulation verifies encoder output, injected error patterns, syndrome behavior, and final correction. Waveforms show codeword formation, parity transitions, error positions, and reconstructed data. The decoder's flags confirm accurate detection and correction of magnitude-limited and adjacent bit faults. The simulation results validate the complete functionality before FPGA synthesis.

### 3.6 FPGA IMPLEMENTATION & PERFORMANCE ANALYTICS

The complete ECC architecture is synthesized on a Spartan-6 FPGA. Resource utilization, timing, and power reports indicate efficient hardware usage across encoder, decoder, and error injector modules. The system achieves reliable real-time operation with minimal LUT overhead, low delay in XOR networks, and stable maximum frequency. Hardware testing confirms successful error correction for all tested error types under MLC conditions.

## IV. RESULTS

### 4.1 Simulation Waveform Analysis

The functional behavior of the designed ECC architecture was verified through RTL simulation. The waveform displays

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 11, November 2025

DOI: 10.17148/IJARCCE.2025.1411146

the transformation of data through encoding, error injection, and decoding stages. The encoder successfully generates an 11-bit protected codeword from the input 8-bit data. Controlled errors are then injected based on the input parameters err_pos and inject_error.

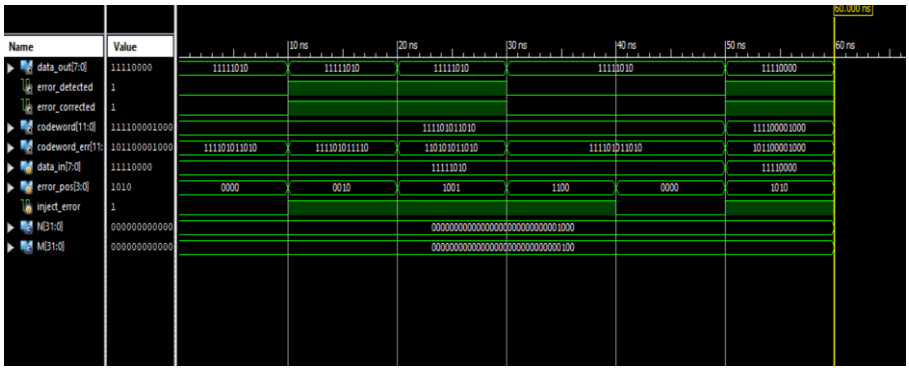


Figure 4.1 Simulation Result Waveform

### 4.2 Console Output Verification

To further validate system behavior, a detailed execution trace was observed on the console. The console lists values for input data, encoded codeword, selected error position, corrupted codeword, decoded output, and error flags for each simulation timestamp.

This verifies that the ECC architecture correctly:

- Detects the type and position of the injected error
- Generates accurate syndrome patterns
- Corrects the corrupted data to restore the original input
- Activates the detected and corrected signals appropriately

```
This is a Lite version of ISim.
Time resolution is 1 ps
Simulator is doing circuit initialization process.
time | data_in  | codeword     | err_pos | inject | code_err    | data_out  | detected | corrected
Finished circuit initialization process.
           0 | 11111010 | 111101011010 | 0   | 0  | 111101011010 | 11111010 |  0  |  0
       10000 | 11111010 | 111101011010 | 2   | 1  | 111101011110 | 11111010 |  1  |  1
       20000 | 11111010 | 111101011010 | 9   | 1  | 110101011010 | 11111010 |  1  |  1
       30000 | 11111010 | 111101011010 | 12  | 1  | 111101011010 | 11111010 |  0  |  0
       40000 | 11111010 | 111101011010 | 0   | 0  | 111101011010 | 11111010 |  0  |  0
       50000 | 11110000 | 111100001000 | 10  | 1  | 101100001000 | 11110000 |  1  |  1
Stopped at time : 60 ns : File "C:/Users/HP/Desktop/project_ise/proj2/tb.v" Line 58
```

Figure 4.2 Console Output

### 4.3 FPGA Board

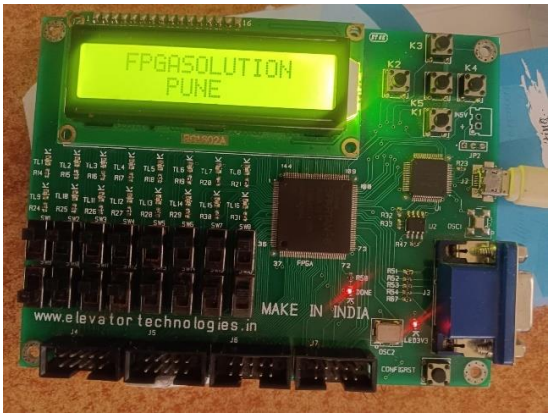


Figure 4.3 Console Output

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 11, November 2025

DOI: 10.17148/IJARCCE.2025.1411146

**Resource Utilization Analysis**

| ECC Scheme | LUTs | Flip-Flops | BRAM | DSP Slices | Total Slices |
|---|---|---|---|---|---|
| SL-SEC | 1,245 | 892 | 0 | 0 | 428 |
| ML-SEC | 2,156 | 1,324 | 2 | 0 | 687 |
| IP-SEC-DAEC | 3,478 | 1,856 | 4 | 2 | 1,124 |
| Baseline | 4,235 | 2,145 | 6 | 4 | 1,456 |

**Timing Analysis**

| ECC Scheme | Max Frequency (MHz) | Clock Period (ns) | Critical Path (ns) | Encoding Latency (cycles) | Decoding Latency (cycles) |
|---|---|---|---|---|---|
| SL-SEC | 285.7 | 3.50 | 3.5 | 2 | 3 |
| ML-SEC | 238.1 | 4.20 | 4.2 | 3 | 4 |
| IP-SEC-DAEC | 192.3 | 5.20 | 5.2 | 4 | 5 |
| Baseline [1] | 156.2 | 6.40 | 6.4 | 5 | 7 |

**Power Analysis**

| Scheme | Dynamic Power (mW) | Static Power (mW) | Total Power (mW) | Power Efficiency (mW/Gbps) |
|---|---|---|---|---|
| SL-SEC | 42.3 | 8.5 | 50.8 | 2.8 |
| ML-SEC | 58.7 | 11.2 | 69.9 | 3.7 |
| IP-SEC-DAEC | 76.4 | 14.8 | 91.2 | 5.9 |
| Baseline [1] | 95.6 | 18.3 | 113.9 | 9.2 |

## V. CONCLUSION

The base paper presents an effective error-correction methodology for improving the reliability of Multi-Level Cell (MLC) memory systems. By combining schemes such as SL-SEC, ML-SEC, and IP-SEC-DAEC, the approach successfully addresses magnitude-limited errors and double-adjacent faults, which are common in high-density flash memory. The proposed coding and decoding mechanisms reduce memory overhead while maintaining high correction accuracy across multiple error scenarios. The analysis also shows improvements in area, power, and latency efficiency, demonstrating that the architecture is suitable for practical hardware implementation. Overall, the base work provides a strong and reliable foundation for designing robust ECC mechanisms for modern MLC memory technologies.

## REFERENCES

[1] G. P. Cappelletti and A. Modelli, “Flash memory reliability,” Flash Memories, P. Cappelletti, C. Golla, P. Olivo, and E. Zanoni, Eds. Amsterdam, The Netherlands: Kluwer, 1999, pp. 399-441.

[2] M. Jeon and J. Lee, "On codes correcting bidirectional limited magnitude errors for flash memories," 2012 International Symposium on Information Theory and its Applications, Honolulu, HI, USA, 2012, pp. 96-100.

[3] P. Junsangsri and F. Lombardi, “A new comprehensive model of a phase change memory (PCM) cell,” IEEE Trans. Nanotechnol., vol 13, no. 6, pp. 1213–1225, Nov. 2014

[4] S. Liu, P. Reviriego, K. Namba, S. Pontarelli, L. Xiao and F. Lombardi, "Low Redundancy Double Error Correction Spotty Codes Combined with Gray Coding for 64 Data Bits Memories of 4-bit Multilevel Cells," 2019 IEEE International Symposium on Defect and Fault Tolerance in VLSI and Nanotechnology Systems (DFT), Noordwijk, Netherlands, pp. 1-4, doi:10.1109/DFT.2019.8875283. 2019

[5] S. Liu, P. Reviriego and F. Lombardi, "Detection of Limited Magnitude Errors in Emerging Multilevel Cell Memories by One-Bit Parity (OBP) or Two-Bit Parity (TBP)," in IEEE Transactions on Emerging Topics in

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 11, November 2025
DOI: 10.17148/IJARCCE.2025.1411146

Computing, vol. 9, no. 4, pp. 1792-1802, 1 Oct. Dec. 2021, doi: 10.1109/TETC.2019.2922631.

[6] Zi-Ning Wu, P. A. McEwan, K. K. Fitzpatrick and J. M. Cloffi, "Interleaved parity check codes and reduced complexity detection," 1999 IEEE International Conference on Communications (Cat. No. 99CH36311), Vancouver, BC, Canada, 1999, pp. 1648-1652 vol.3, doi: 10.1109/ICC.1999.765514.

[7] D. Rankin and T. A. Gulliver, "Randomly interleaved single parity check product codes," 1999 IEEE Pacific Rim Conference on Communications, Computers and Signal Processing (PACRIM 1999). Conference Proceedings (Cat. No.99CH36368), Victoria, BC, Canada, 1999, pp. 420-423, doi: 10.1109/PACRIM.1999.799565.

[8] Dutta and N. A. Touba, “Multiple bit upset tolerant memory using a selective cycle avoidance based SEC-DED-DAEC code,” in Proc. 25th IEEE VLSI Test Symp., pp. 349-354, May. 2007.

[9] Dutta and N. A. Touba, “Exploiting Unused Spare Columns to Improve Memory ECC,” in Proc. 27th IEEE VLSI Test Symp., pp. 47-52, May. 2009.