ISSN 2319-5991 www.ijerst.com
© 2026 IJERST. All Rights Reserved
Peer Reviewed, Referred & Indexed Journal

Research Paper

SYNDROME BASE DECODING ALGORITHM FOR DOUBLE ERROR CORRECTION CODES

[1]A. Baby
vanisriamboji@gmail.com

[2]Dr. N. Suresh
Ecehod@avniet.ac.in

[1]Scholar, Master of Technology, Dept of Electronics and Communication Engineering, AVN Institute of Engineering and Technology Koheda Road M.P.Patel Guda Post Ibrahimpatnam, Dist., Hyderabad, Telangana 501510

[2]Professor& Hod, AVN Institute of Engineering and Technology Koheda Road M.P.Patel Guda Post Ibrahimpatnam, Dist., Hyderabad, Telangana 501510

ABSTRACT

The project titled "Syndrome-Based Decoding Algorithm for Double Error Correction Codes" focuses on improving the reliability of memory systems by detecting and correcting errors in stored data as well as in the supporting encoder and decoder circuits. Traditional memory systems use Error-Correcting Codes (ECCs), such as Hamming codes, to correct single-bit errors and detect double-bit errors. However, these systems generally do not adequately address transient faults occurring within the encoder and decoder circuitry, which can lead to incorrect data processing and system failures.

The proposed system employs a fault-tolerant memory architecture using syndrome-based error detection, fault-secure detectors, and error-correction techniques. The information vector is encoded into a codeword using a generator matrix and stored in memory. During data retrieval, a syndrome is calculated using the received codeword and the parity-check matrix to identify potential errors. The system uses a detect-and-repeat mechanism to verify encoder and corrector outputs, repeating operations when errors are detected. Periodic memory scrubbing is also performed to identify, correct, and rewrite erroneous data, preventing errors from accumulating beyond the correction capability of the code.

The design also incorporates Euclidean Geometry Low-Density Parity-Check (EG-LDPC) codes and Projective Geometry codes to support fault-secure error detection and reliable data processing. The proposed architecture aims to improve data accuracy, reduce the probability of undetected errors, and increase the reliability of memory and communication systems. The design can be implemented and verified using hardware description languages such as Verilog or VHDL, with simulation tools such as ModelSim and Xilinx ISE. Overall, the project provides a fault-tolerant approach to double-error correction and protection against transient faults in memory storage and supporting logic circuits.

Keywords:

Syndrome-Based Decoding, Double Error Correction Codes, Error-Correcting Codes (ECC), Fault-Tolerant Memory Systems, Fault-Secure Detectors, Euclidean Geometry LDPC Codes, Transient Faults, Memory Scrubbing, Verilog, VLSI Design.

I. INTRODUCTION

Memory systems play a vital role in modern digital devices, computers, communication systems, and VLSI-based applications. As technology advances, memory density and operating speeds continue to increase, making electronic circuits more susceptible to transient faults and soft errors. Error-

Correcting Codes (ECCs) are widely used to improve data reliability by adding redundant bits to the original information. Traditional Hamming codes can correct single-bit errors and detect double-bit errors in a memory word. However, conventional memory protection techniques mainly focus on errors occurring in stored data and do not adequately protect the supporting encoder and decoder circuitry against transient faults.

The Syndrome-Based Decoding Algorithm for Double Error Correction Codes aims to improve the reliability of memory systems by detecting and correcting errors in both stored data and supporting logic circuits. The proposed approach uses fault-secure detectors to verify the correctness of encoder and decoder outputs. The information vector is converted into a codeword using a generator matrix and stored in memory. When the data is retrieved, a syndrome is calculated using the received codeword and the parity-check matrix. A zero syndrome indicates a valid codeword, whereas a non-zero syndrome indicates a potential error. Based on the detection results, the system identifies errors and performs the required correction operations.The proposed architecture incorporates Euclidean Geometry Low-Density Parity-Check (EG-LDPC) codes and Projective Geometry codes to support reliable error detection and correction. A detect-and-repeat mechanism is used to repeat encoding or correction operations whenever the fault-secure detector identifies an error. Additionally, memory scrubbing periodically reads stored codewords, corrects detected errors, and writes the corrected data back into memory to prevent error accumulation. The system can be designed using Verilog or VHDL and evaluated through simulation tools such as ModelSim and Xilinx ISE. This approach aims to improve data integrity, reduce the risk of undetected errors, and enhance the fault tolerance of memory systems used in digital electronics and communication applications.

II. LITERATURE REVIEW

Hamming [1] introduced error-detecting and error-correcting codes to identify and correct errors in digital information, establishing the foundation for modern Error-Correcting Codes (ECCs). Gallager [2] proposed Low-Density Parity-Check (LDPC) codes using sparse parity-check matrices for efficient error correction. MacWilliams and Sloane [3] presented the mathematical foundations of coding theory, including generator matrices, parity-check matrices, and minimum distance. Peterson and Weldon [4] discussed error-correcting codes and their applications in digital communication systems. These studies provide the theoretical basis for designing reliable encoding and decoding mechanisms that improve data integrity and reduce transmission errors.

Naeimi and DeHon [5] investigated fault-secure encoders and decoders for memory applications, focusing on detecting errors in both memory data and supporting logic circuits. Their approach used fault-secure detectors and geometry-based codes to improve system reliability. Shannon [6] established the theoretical relationship between information transmission, redundancy, and communication reliability. Erozan et al. [7] investigated adjacent error detection techniques for nanoscale devices using Euclidean Geometry LDPC codes. These studies highlight the importance of reliable error detection, efficient coding structures, and protection against transient faults. However, designing memory systems that can detect errors in encoder and decoder circuitry while maintaining efficient hardware operation remains an important challenge.

Raveendran et al. [8] studied soft-syndrome iterative decoding of quantum LDPC codes and associated hardware architectures, emphasizing reliable syndrome processing and efficient implementation. Shibata [9] reviewed synchronization error-correction coding based on probabilistic inference and sparse matrices, addressing reliability challenges in communication and recording systems. Maan et al. [10] investigated graph-based decoding techniques for correlated errors in quantum LDPC codes, contributing to developments in syndrome-based error correction. Based on these research directions, the proposed project, Syndrome-Based Decoding Algorithm for Double Error Correction Codes, focuses on using syndrome calculation, fault-secure detectors, detect-and-repeat operations, and memory scrubbing to improve the reliability of memory systems. The architecture aims to detect and correct errors in stored data and supporting circuits, thereby improving data accuracy and fault tolerance in digital systems.

III. EXISTING SYSTEM

The existing system uses conventional Error-Correcting Codes (ECCs) to detect and correct errors that occur in stored or transmitted digital data. Techniques such as Hamming codes and parity-check methods use redundant bits to identify errors and, depending on the code design, correct certain numbers of bit errors. Syndrome calculation compares the received data with the parity-check matrix to determine whether an error has occurred. However, conventional ECC-based systems mainly focus on errors in the stored data and may not adequately protect the encoder, decoder, and other supporting logic against transient faults. Their error-correction capability is also limited by the specific code used.

IV. PROPOSED SYSTEM

The proposed system introduces a syndrome-based decoding algorithm for double error correction codes to improve the reliability of memory systems. It uses encoding and decoding circuits, parity-check matrices, syndrome calculation, and fault-secure detectors to identify errors in stored data and supporting logic circuits. During encoding, redundant parity bits are added to the original data. During decoding, the syndrome is calculated using the received codeword and the parity-check matrix. Based on the syndrome, the system identifies error conditions and applies the appropriate correction procedure when the error pattern is within the selected code's correction capability. methodology

The proposed architecture also incorporates detect-and-repeat operations to handle detected faults in the encoder or decoder, along with memory scrubbing to periodically check and correct eligible errors in stored data. Geometry-based codes, such as Euclidean Geometry LDPC codes and Projective Geometry codes, can be used to improve error-detection and correction capabilities. The design can be described using Verilog or VHDL and tested with simulation tools such as ModelSim and Xilinx ISE.

V. METHODOLOGY

The proposed system follows a syndrome-based decoding methodology to detect and correct errors in memory data while improving the reliability of encoder and decoder circuits. The methodology consists of the following steps.

1. Data Encoding: The input information bits are supplied to the encoder. Using a generator matrix \(G\), the encoder generates a codeword by adding redundant bits for error detection and correction.

\[C = I \times G \]

where \(I\) represents the input information vector, \(G\) is the generator matrix, and \(C\) is the encoded codeword.

2. Memory Storage: The encoded codeword is stored in the memory block. During storage, transient faults may change one or more bits, potentially producing an incorrect codeword.

3. Syndrome Calculation: When the data is retrieved, the decoder uses the parity-check matrix \(H\) to calculate the syndrome.

\[S = R \times H^T \]

where \(R\) is the received codeword, \(H^T\) is the transpose of the parity-check matrix, and \(S\) is the syndrome vector. A zero syndrome indicates that the received word satisfies the parity checks, while a nonzero syndrome indicates a possible error.

4. Error Detection and Correction: The decoder analyzes the syndrome to identify the error pattern. If the selected code supports correction of the detected double-bit error pattern, the decoder corrects the affected bits. If the error is outside the code's correction capability, the system should report an uncorrectable error rather than assume successful correction.

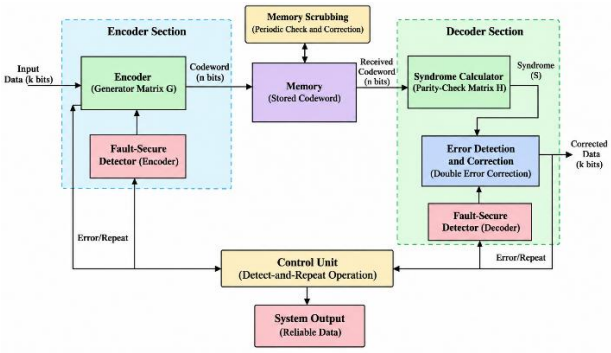
5. Fault-Secure Detection: Fault-secure detectors monitor the outputs of the encoder and corrector. If a fault is detected, the system initiates a detect-and-repeat operation to repeat the affected operation and attempt to produce a valid result.

6. Memory Scrubbing: The system periodically reads stored codewords, checks their syndromes, corrects errors that are within the code's capability, and writes the corrected codewords back into memory. This helps prevent errors from accumulating over time.

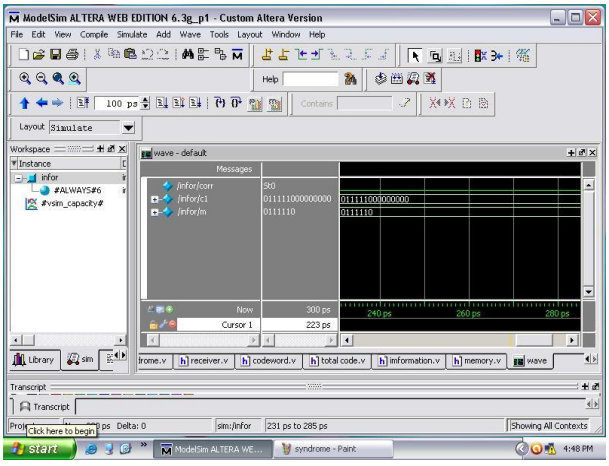
7. Hardware Implementation and Testing: The architecture can be described using Verilog or VHDL and simulated using ModelSim or Xilinx ISE. Testing should evaluate correct encoding and decoding, single- and double-bit error cases, fault detection, and recovery behavior. The results can be used to assess error-correction performance and hardware reliability.

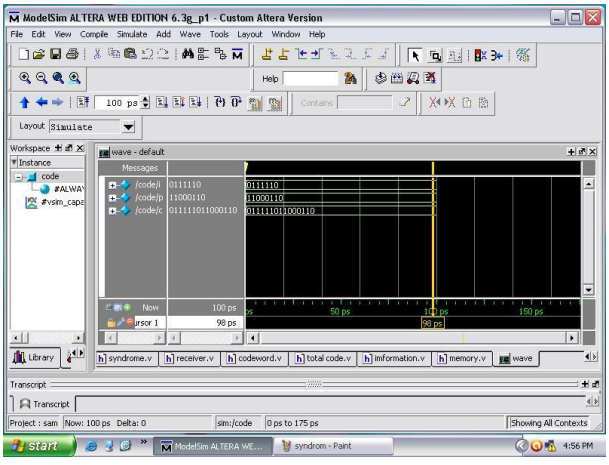
VI. SYSTEM MODEL

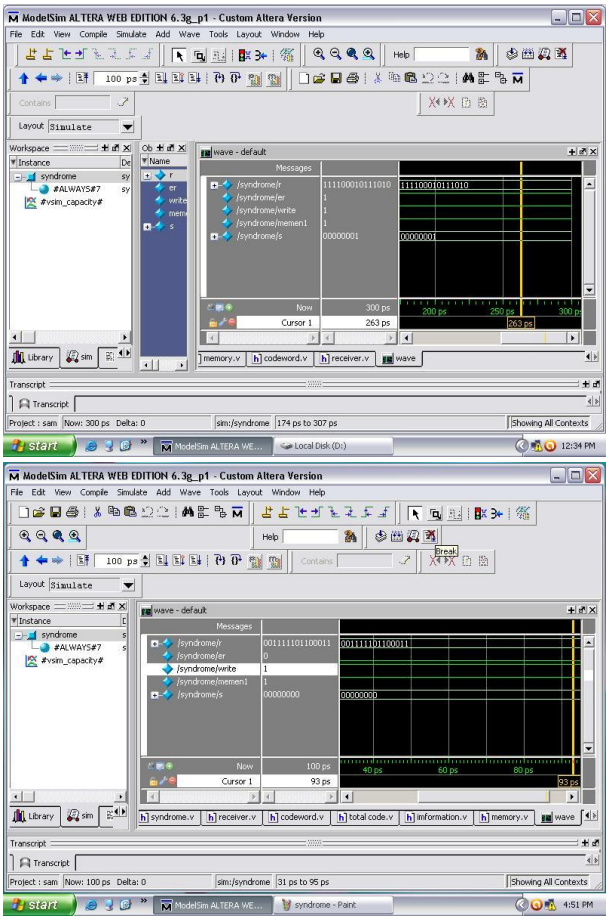
System Architecture



VII. RESULTS AND DISCUSSIONS







VIII. CONCLUSION

The project "Syndrome-Based Decoding Algorithm for Double Error Correction Codes" focuses on improving memory reliability through syndrome-based error detection and correction techniques. The proposed system uses generator and parity-check matrices, syndrome calculation, fault-secure detectors, detect-and-repeat operations, and memory scrubbing to identify errors and improve data integrity. Geometry-based codes, such as Euclidean Geometry LDPC and Projective Geometry codes, can provide additional error-correction capabilities when appropriately designed for the required error patterns.

The proposed architecture aims to overcome limitations of conventional error-correcting methods by considering faults in both stored data and supporting encoder and decoder circuits. Hardware description languages such as Verilog or VHDL can be used to implement the design, while ModelSim and Xilinx ISE can support functional simulation and verification. Overall, the project provides an approach for developing more reliable and fault-tolerant memory systems. Its effectiveness should be validated through simulation and hardware evaluation, including error-correction capability, detection accuracy, processing delay, and hardware overhead.

IX. FUTURE WORK

In the future, the proposed syndrome-based decoding algorithm can be extended to support more complex error patterns and improve the correction of multiple-bit errors. Advanced error-correcting codes, such as optimized LDPC and BCH codes, can be explored to enhance error-detection capability and reduce decoding complexity. The decoding algorithm can also be optimized to reduce processing delay, power consumption, and hardware resource utilization while maintaining reliable memory operation. Recent research on LDPC decoding highlights the importance of efficient decoding architectures and handling unreliable syndrome information.

Furthermore, the proposed architecture can be implemented on FPGA platforms and evaluated under different fault conditions. Future versions may integrate improved fault-secure detectors, adaptive memory-scrubbing intervals, and enhanced recovery mechanisms to handle repeated transient faults. The system can also be tested with larger memory configurations and compared with conventional error-correcting methods using metrics such as error-correction rate, detection accuracy, latency, power consumption, and hardware overhead. These improvements could help develop more efficient and reliable fault-tolerant memory systems for embedded devices, communication systems, and high-performance computing applications.

X. REFERENCES

[1] R. W. Hamming, "Error Detecting and Error Correcting Codes," Bell System Technical Journal, vol. 29, no. 2, pp. 147–160, 1950. https://doi.org/10.1002/j.1538-7305.1950.tb00463.x.

[2] R. G. Gallager, "Low-Density Parity-Check Codes," IRE Transactions on Information Theory, vol. 8, no. 1, pp. 21–28, 1962. https://doi.org/10.1109/TIT.1962.1057683.

[3] F. J. MacWilliams and N. J. A. Sloane, The Theory of Error-Correcting Codes. North-Holland, 1977.

[4] W. W. Peterson and E. J. Weldon, Error-Correcting Codes, 2nd ed. MIT Press, 1972.

[5] H. Naeimi and A. DeHon, "Fault Secure Encoder and Decoder for Memory Applications," in Proceedings of the 22nd IEEE International Symposium on Defect and Fault-Tolerance in VLSI Systems, pp. 409–417, 2007. https://doi.org/10.1109/DFT.2007.54.

[6] C. E. Shannon, "A Mathematical Theory of Communication," Bell System Technical Journal, vol. 27, pp. 379–423, 623–656, 1948.

[7] S. Lin and D. J. Costello, Error Control Coding, 2nd ed. Pearson Prentice Hall, 2004.

[8] H. Tang, J. Xu, S. Lin, and K. A. S. Abdel-Ghaffar, "Codes on Finite Geometries," IEEE Transactions on Information Theory, vol. 51, no. 2, pp. 572–596, 2005.

[9] S. J. Piestrak, A. Dandache, and F. Monteiro, "Designing Fault-Secure Parallel Encoders for Systematic Linear Error Correcting Codes," IEEE Transactions on Reliability, vol. 52, no. 4, pp. 492–500, 2003.

[10] H. Naeimi and A. DeHon, "Fault Secure Encoder and Decoder for NanoMemory Applications," IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 17, no. 4, pp. 473–486, 2009.