Journal of Science Engineering Technology and Management Science
Volume 02, Issue 07, June 2025 *ISSN: 3049-0952*
www.jsetms.com

Design of hybrid memory using ECC, EDC, & BIST using Verilog and verification using system Verilog.

[1]Medikondu Abhinav, [2]Abhiram, [3]P.Enosh, [4]M. Lokesh,[5]M.Bhanu, [6]DR. J.Kavitha

[1,2,3,4,5]U. G Student, Dept ELECTRONICS AND COMMUNICATION ENGINEERING, St.Ann's College of Engineering and Technology (Autonomous), Chirala, Bapatla Dist, Andhra Pradesh – 523187, India

[6]Associated professor Dept. of ECE,St. Ann's College of Engineering & Technology,Chirala, India

Abstract:

Random Access Memory (RAM) system has been designed and functionally verified using Verilog HDL and SystemVerilog-based verification environment. The RAM module incorporates Error Correction Code (ECC) and Built-In Self-Test (BIST) mechanisms to enhance data integrity and reliability against transient and permanent faults, which are critical in safety-critical and high-reliability systems.The ECC encoder implements a Hamming SEC-DED (Single Error Correction) scheme by generating 5 parity bits for every 16-bit data, resulting in a 21-bit encoded output. During read operations, the ECC decoder computes the syndrome to detect and correct single-bit errors and flag double-bit errors. The BIST controller automates the test process by writing known patterns into memory, reading them back, and comparing the values to detect permanent faults without external test equipment.

The top-level module integrates the RAM, ECC encoder/decoder, and BIST controller. It supports normal operation mode and self-test mode, making the system capable of runtime error detection and correction. The entire design is validated using a SystemVerilog testbench to ensure functional correctness under various scenarios, including fault injection and random stimuli. Simulation results confirm the system's ability to correct single-bit errors, detect double-bit errors, and identify memory faults, demonstrating its effectiveness and reliability in fault-prone environments

Introduction:

The fundamental element of System-On-Chip (SOC) systems is entrenched memories. Generally, these memories are implemented using SRAMs as they are robust, have high speed and easily incorporated in logic circuits. But they suffer from disadvantage that they occupy more area which affects power and the yield. Dynamic Random Access Memory is a kind of RAM in which data bits are stored in capacitors in ICs. Even though DRAM'S can be easily implemented and require lower area than SRAM, they suffer from the drawback that they have to be refreshed in order to retain data stored in them. Static RAM (SRAM) is advanced type of memory

and doesn't need refresh like DRAM'S. They are realized using Flip Flops and they retain data. They are faster than DRAM'S but need more area for a given memory compared to DRAM. Single port RAM is simple to implement and need less area. But its drawbacks are that it is slow and data access at an address is one at time of each clock pulse. Dual port RAM overcomes these shortcomings. It enables quicker data access at various addresses at the same time. The aim is to improve parameters like area, speed, power. In modern digital systems, memory subsystems play a crucial role in storing and retrieving data for computational operations. With the increasing demand for high-performance and safety-critical applications in fields such as aerospace, automotive systems, medical devices, and mission-critical embedded systems, ensuring the reliability and integrity of memory operations has become paramount. Transient faults caused by radiation (soft errors) and permanent faults due to aging or manufacturing defects can significantly degrade the performance and safety of such systems. As a result, designing fault-tolerant memory architectures has become a key area of research and development in digital system design.This work presents the design and verification of a fault-tolerant 64×16 Random Access Memory (RAM) system using Verilog Hardware Description Language (HDL), along with a SystemVerilog-based verification framework. The proposed memory system is equipped with Error Correction Code (ECC) and Built-In Self-Test (BIST) mechanisms to ensure robustness against both transient and permanent faults. The ECC mechanism employed is based on the Hamming SEC-DED (Single Error Correction – Double Error Detection) code, which enhances reliability by generating 5 parity bits for every 16-bit data word, forming a 21-bit encoded data word. During read operations, the ECC decoder analyzes the syndrome vector to detect and correct single-bit errors and to flag double-bit errors, thereby maintaining data integrity.

To further enhance reliability and ease of testing, a BIST controller is incorporated. The BIST module enables autonomous memory testing without the need for external equipment. It operates by writing predefined patterns into memory, reading them back, and comparing the output to detect permanent faults such as stuck-at faults or address decoding issues. This makes the RAM system suitable for in-field diagnostics and self-maintenance, especially in environments where external test access is limited or impractical.

The top-level architecture of the system integrates the memory core, ECC encoder/decoder, and the BIST controller. It is designed to operate in two distinct modes: normal operation mode for typical read/write access, and self-test mode to invoke the BIST process. This dual-mode operation allows the system to provide continuous runtime fault detection and correction, which is a crucial requirement for fault-resilient computing platforms.

To ensure the functional correctness and robustness of the proposed system, a comprehensive verification environment is developed using SystemVerilog. The verification strategy includes randomized test stimuli, corner-case scenarios, and fault injection techniques to rigorously evaluate the system under diverse operational conditions. Simulation results confirm the system's ability to correct single-bit errors, detect uncorrectable double-bit errors, and identify memory failures, validating the effectiveness of the fault-tolerant mechanisms.

This work demonstrates a scalable and efficient solution for fault-tolerant memory design using industry-standard HDL and verification techniques. The proposed architecture is well-suited for deployment in environments where high reliability, fault resilience, and autonomous testing are critical.

LITERATURE SURVEY

A new P-MBIST with the aim of merging the FSM and Microcode architecture using macro-commands is proposed. The hybrid P-MBIST utilizes the same macro-commands for selecting the test algorithm and same encoding technique for the MARCH elements but instead of using state machines, it is designed by implementing clusters of microcode to control the read/write operation and test data injection. EDA industry is see-king maintenance methodologies to support its software, and to improve the overall quality of tools as they are affecting customer satisfaction. Monitoring activities of tools and detecting post development software errors cannot be overestimated. The experiments show the ability of the TMB Validator to verify various controller features and demonstrate its versatility to determine reliably when working with a variety of memory fault models. The Current March Algorithm with 22 N is inefficient in certain cases to make a full diagnosis of SRAM.

The proposed scheme is more efficient in terms of circuit size and test data to be applied, and it requires less time to test

SRAM chip. The area occupied by embedded recollections in System-on-Chip (SoC) is over 90%, and expected to elevate up to 94% by 2014. Thus, the performance and yield of embedded recollections will dominate that of SoCs. SRAM is more expensive and less dense than DRAM and is therefore not used for high capacity, low cost applications such as the main memory in personal computers. Programmable BIST approaches, allowing selecting after fabrication a large variety of memory tests, are therefore desirable, but may lead on unacceptable area cost. BIST approaches enabling test algorithm program ability and data background program ability at low area cost have been presented in the past. However, no proposals exist for programming the address sequence used by the test.

EXISTING METHOD

Single-port memories will be having only one port for reading and writing data. The data connections can be separated into output and input connections, and it will have only one address input. A read or write operation is only possible with reference to single-port memory which can provide only one access at a time.To overcome the disadvantage associated with single-port memories, we have designed multi-port memories. Multi-port memories will have many address inputs with corresponding data inputs and outputs. Concurrent operation depends on the number of input address line. Dual-port memory is the most common type of multi-port memory.

Area consumption of multi-port memory is more compared to single-port memory with same number of bits of storage, the reason for this is multi-port memories consists of separate address decoder and data multiplexers for each access port. Additional wiring will be required to connect the cells to the access the port, and the internal storage is shared with the memories. Additional cost can be justified for many applications like high-speed network connections and high performance graphics processing. Let us consider a scenario in which we have a subsystem which produces the data for storing the data in memory and another subsystem to access the data to process it. If we implement the above system using single-port memory we have an additional task of multiplexing the data and address of the subsystem with the memory.

The arrangements of the control section has to be made in such a way that it has to take turns to access the memory. Now let us discuss the problems associated with the mentioned technique, the memory becomes

a bottle neck when the total rate of moving the data in and out of the subsystem exceed the overall rate. If have more than one subsystems and two systems have a requirement of having access to memory at the same time then there will be loss of data. The solution to the mentioned problem is to incorporate separate access ports for subsystems.

In asynchronous dual-port memory simultaneous access results in delayed response. If the operation is write and if the two subsystems try's to perform it simultaneously then it will result in unpredictable results getting stored in memories. Multi-port memories manufactured in the form of packaging components can avoid this issues by using additional circuits to indicate the time at which contention take places.

Proposed method

The proposed system introduces a **fault-tolerant 64×16 Random Access Memory (RAM)** architecture that integrates Error Correction Code (ECC) and Built-In Self-Test (BIST) techniques to ensure high reliability and data integrity in modern digital systems. The design primarily targets environments where memory faults can significantly impact system performance, such as safety-critical and mission-critical applications.

To address transient faults, the system employs a **Hamming SEC-DED (Single Error Correction and Double Error Detection)** based ECC mechanism. The ECC encoder converts 16-bit input data into a 21-bit codeword by appending parity bits, while the ECC decoder performs syndrome analysis during read operations to detect and correct single-bit errors and flag double-bit errors. This ensures that data corruption is minimized and reliability is maintained during runtime.

In addition to ECC, a **Built-In Self-Test (BIST) controller** is incorporated to detect permanent faults within the memory. The BIST operates using a finite state machine that performs sequential write and read operations with predefined test patterns across all memory locations. The read data is compared with expected values to identify faults such as stuck-at and addressing errors, enabling autonomous testing without external hardware support.

The overall architecture is designed with a **top-level integration module** that combines the RAM core, ECC encoder/decoder, and BIST controller. The system supports dual modes of operation: a normal mode for standard memory access and a self-test mode for diagnostic operations. During normal operation, data is encoded, stored, and decoded with real-

time error handling, whereas in test mode, the BIST controller takes control to validate memory integrity.

The design is implemented using Verilog HDL and verified using a SystemVerilog-based testbench with fault injection and randomized testing techniques. Simulation results demonstrate that the system effectively corrects single-bit errors, detects double-bit errors, and identifies permanent memory faults. Thus, the proposed system provides a robust, self-reliable, and efficient solution for fault-tolerant memory design.

ARCHITECTURE

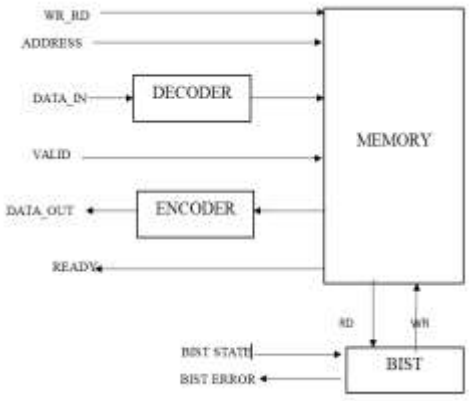

Figure 1: proposed method architecture

METHODOLOGY DESCRIPTION

Input Data (16-bit)

This block represents the input data provided to the system. It consists of 16-bit user data that needs to be stored in memory. The data is given along with control signals such as clock, write enable, and valid signal.

ECC Encoder

The ECC (Error Correction Code) Encoder generates redundant parity bits using the Hamming SEC-DED scheme. It converts the 16-bit input data into a 21-bit encoded data by adding check bits. This helps in detecting and correcting errors during read operations.

RAM (64×16 Memory)

The RAM module is the main storage unit of the system, consisting of 64 memory locations with 16-bit word size. It stores the encoded data received from the ECC encoder and performs read/write operations based on control signals.

ECC Decoder

The ECC Decoder checks the integrity of the stored data during read operations. It calculates the syndrome to identify errors. Single-bit errors are corrected automatically, while double-bit errors are detected and flagged without correction.

BIST Controller

The Built-In Self-Test (BIST) Controller is responsible for testing the memory without external hardware. It generates test patterns, writes them into memory, reads them back,

and compares the results to detect faults such as stuck-at errors and address faults.

Output Data (16-bit)

This block provides the final corrected data after ECC decoding. It ensures that the output is reliable and free from single-bit errors, maintaining data integrity.

Control Logic (Top Module)

This block manages the overall operation of the system. It controls mode selection (normal mode or test mode), coordinates between ECC, RAM, and BIST modules, and ensures proper data flow across the system.

SOFTWARE USED

QuestaSim:

Questa-Sim is an advanced, high-performance digital logic simulator used for verifying the functionality of complex electronic designs. QuestaSim software, a high-performance simulator from Siemens used for electronic design automation (EDA) to verify complex digital circuits like ASICs, SoCs, and FPGAs. It is a part of Siemens' Questa Advanced Functional Verification Platform and is used to simulate hardware designs written in languages like Verilog, VHDL, and System Verilog. It provides advanced features for verification and is considered a modern, 64-bit successor to the older ModelSim simulator.

RESULT AND DISCUSSION

The proposed fault-tolerant 64×16 RAM system was successfully designed and verified using Verilog and SystemVerilog simulation. The simulation results confirm that the ECC encoder and decoder effectively perform **single-bit error correction and double-bit error detection** using the Hamming SEC-DED scheme. During fault injection testing, single-bit errors were automatically corrected, while double-bit errors were accurately detected and flagged.

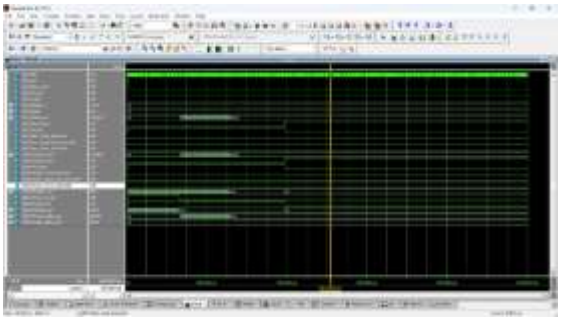

Figure 3.1: - waveform

The testbench is developed using SystemVerilog to verify the functionality of the proposed RAM system. It generates clock and reset signals and provides input data to the design.

Different test cases are applied to check both **write and read operations** of the

memory. To test the error correction capability, faults are introduced manually. The system successfully corrects single-bit errors and detects double-bit errors using the ECC technique.The BIST controller is also tested by enabling test mode, where it automatically writes and reads data to identify faults in memory.The simulation results show that the system works correctly under all conditions. The coverage report indicates **100% coverage**, proving that all parts of the design are properly tested.

Figure 3.2: - testbench

CONCLUSION

In this project, a fault-tolerant 64×16 RAM system with ECC and BIST is successfully designed and verified. The system is capable of correcting single-bit errors and detecting double-bit errors, improving data reliability. The BIST controller helps in identifying faults automatically without external support. Simulation results and coverage report confirm that the design works correctly under different conditions. Overall, the proposed system provides an efficient and reliable solution for memory fault detection and correction.

FUTURE ENHANCEMENTS

The proposed fault-tolerant RAM system can be further improved in several ways to enhance its performance and applicability. Firstly, more advanced error correction techniques such as BCH or Reed-Solomon codes can be implemented to handle multiple-bit errors instead of only single-bit correction.Secondly, the memory size can be scaled from 64×16 to larger configurations to support high-capacity applications. Power optimization techniques can also be introduced to reduce energy consumption, making the design suitable for low-power and portable devices.Additionally, the BIST architecture can be enhanced by incorporating March algorithms for more efficient fault detection and improved test coverage. The system can also be extended to support real-time fault monitoring and dynamic error recovery.

Finally, the design can be implemented on FPGA or ASIC platforms for hardware validation and performance analysis in real-world conditions.efficient and reliable

solution for memory fault detection and correction.

REFERENCES

[1] W. Stallings, *Computer Organization and Architecture*, 10th ed., Pearson, 2016.

[2] M. Morris Mano and M. D. Ciletti, *Digital Design*, 5th ed., Pearson, 2013.

[3] S. Lin and D. J. Costello, *Error Control Coding*, 2nd ed., Pearson, 2004.

[4] Z. Kohavi and N. K. Jha, *Switching and Finite Automata Theory*, 3rd ed., Cambridge University Press, 2010.

[5] IEEE Standard for Verilog Hardware Description Language, IEEE Std 1364-2005.

[6] IEEE Standard for SystemVerilog—Unified Hardware Design, Specification, and Verification Language, IEEE Std 1800-2017.

[7] Mentor Graphics, "QuestaSim User Manual," Siemens EDA, 2020.

[8] R. E. Bryant, "A Methodology for Hardware Verification Using Simulation and Coverage," *IEEE Transactions on Computer-Aided Design*, vol. 10, no. 9, pp. 1101–1110, 1991.

[9] R. W. Hamming, "Error detecting and error correcting codes," *Bell System Technical Journal*, vol. 29, no. 2, pp. 147–160, Apr. 1950.

[10] S. Lin and D. J. Costello, *Error Control Coding: Fundamentals and Applications*, 2nd ed. Upper Saddle River, NJ, USA: Prentice Hall, 2004.

[11] M. Bushnell and V. D. Agrawal, *Essentials of Electronic Testing for Digital, Memory and Mixed-Signal VLSI Circuits*. Boston, MA, USA: Springer, 2000.

[12] A. J. van de Goor, *Testing Semiconductor Memories: Theory and Practice*. Hoboken, NJ, USA: Wiley, 1991.

[13] J. Rajski and J. Tyszer, *Arithmetic Built-In Self-Test for Embedded Systems*.