International Journal of Advanced Research in Information and Communication Engineering (IJARCSA) ISSN: 2321-8940
Vol.5, Issue No 4, 2025

A Hybrid Approach for Mitigating Transient and Permanent Faults in Memory Subsystems Using EDC, ECC, and BIST

[1]Kumar pari , [2]M.Abhinav, [3]P.Abhiram, [4]P.Enosh, [5]M.lokesh, [6]M.Bhanu

[1]Assistant Professor, Dept of Electronics & Communication Engineering,
St. Ann's College of Engineering and Technology Autonomous,
Chirala 523155, India.

[2,3,4,5,6]U.G. Student, Dept of Electronics & Communication Engineering,
St. Ann's College of Engineering and Technology Autonomous,
Chirala 523155, India.

ABSTRACT

A fault-tolerant 64x16 Random Access Memory (RAM) system has been designed and functionally verified using System-Verilog-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 – Double Error Detection) 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.

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.

EXISTING SYSTEM

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.

PROPOSED SYSTEM

The proposed system presents a hybrid fault mitigation framework for memory subsystems by integrating Error Detection Codes (EDC), Error Correction Codes (ECC), and Built-In Self-Test (BIST) techniques. The system is designed to handle both transient and permanent memory faults efficiently. EDC is employed for fast detection of soft errors caused by radiation and noise. ECC is used to correct single-bit errors and detect multi-bit errors during normal memory operation. BIST is periodically activated to identify permanent faults such as stuck-at and bridging faults. The hybrid architecture ensures continuous online protection as well as offline fault diagnosis. Memory read and write operations are monitored using EDC and ECC logic. When an error is detected, ECC attempts correction and reports uncorrectable errors. BIST isolates faulty memory locations during idle cycles. Faulty memory blocks are marked and remapped if redundancy is available. The system reduces performance overhead by selectively activating BIST. Power consumption is optimized by adaptive fault monitoring. The hybrid approach improves system reliability and fault coverage. It minimizes data corruption and system crashes. The design is suitable for safety-critical and high-reliability applications. The proposed system ensures high memory availability. It offers scalability for modern memory architectures. Overall, the hybrid solution provides robust and efficient fault tolerance.

SYSTEM ARCHITECTURE

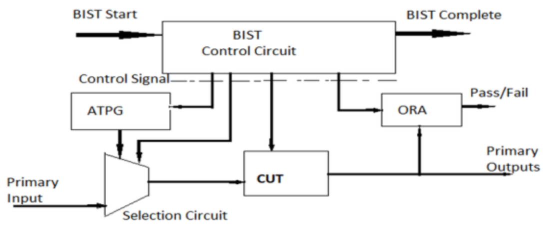

Fig.1 System Architecture

METHODOLOGY

DESCRIPTION

design of a hybrid memory fault mitigation architecture integrating EDC, ECC, and BIST modules. During normal memory operations, Error Detection Codes are generated and stored alongside data. On every read operation, EDC verifies data integrity to detect transient faults. If an error is detected, the Error Correction Code module is activated. ECC corrects single-bit errors and identifies multi-bit errors in real time. Corrected data is forwarded to the processor without interrupting operation. Uncorrectable errors are logged for further analysis. Built-In Self-Test is scheduled during system idle or startup phases. BIST applies predefined test patterns to memory cells. The responses are compared with expected outputs to detect permanent faults. Identified faulty locations are isolated from active memory usage. Redundancy or remapping techniques are used if available. The system continuously monitors fault statistics. Adaptive control logic adjusts testing frequency based on fault occurrence. The integration ensures minimal performance overhead. Power-efficient operation is maintained by selective module activation. The methodology ensures both online and offline fault coverage. Reliability metrics are evaluated after fault mitigation. The overall approach enhances memory robustness. This methodology provides effective fault tolerance for modern memory subsystems.

RESULTS & DISCUSSION:

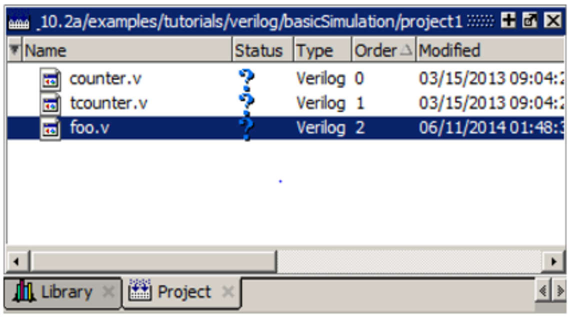

Fig.2 Home Page

The Files The question marks in the Status column in the Project window indicate that either the files have not been compiled into the project or that the source has changed since the last compile. Procedure Choose Compile>Compile All or right-click in the Project window and choose Compile>Compile All. Results Once compilation finishes, click the Library window, expand the library work by clicking the "+", and you will see the compiled design units.

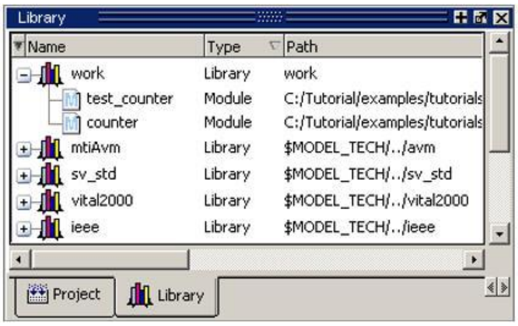

Fig.3 Sign In Page

Change Compile Order The Compile Order dialog box is functional for HDL-only designs. When you compile all files in a project, QuestaSIM by default compiles the files in the order in which they were added to the project. You have two alternatives for changing the default compile order: Select and compile each file individually Specify a custom compile order Procedure

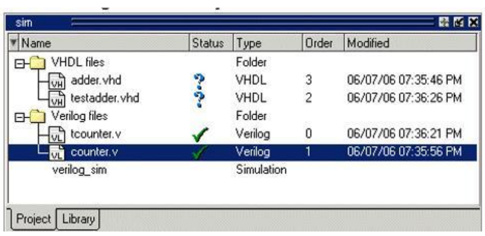

Fig.4 Overview Page

Column titles o Name – The name of a file or object. o Status– Identifies whether a source file has been successfully compiled.

Applies only to VHDL or Verilog files. A question mark means the file has not been compiled or the source file has changed since the last successful compile; an X means the compile failed; a check mark means the compile succeeded; a checkmark with a yellow triangle behind it means the file compiled but there were warnings generated.

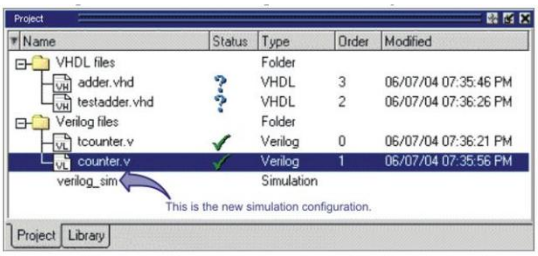

Fig.5 Simulation Page

When code coverage is enabled for an entire design, or for the purposes of debugging a particular segment of the design, you can use coverage exclusions to exclude coverage for individual design units, files, lines of code, objects, and so on. You can exclude coverage objects with the coverage exclude command, or with source code pragmas

CONCLUSION & FUTURE ENHANCEMENT

The design and verification of the 64x16 fault-tolerant Random Access Memory (RAM) system demonstrate a robust solution for ensuring data integrity in environments prone to transient and permanent faults. By integrating Error Correction Code (ECC) based on the Hamming SEC-DED scheme and a Built-In Self-Test (BIST) mechanism, the system effectively addresses both soft and hard errors. The ECC module enhances runtime reliability by detecting and correcting single-bit errors and flagging double-bit errors, while the BIST controller enables automated fault detection without the need for external testing infrastructure.

REFERENCE

[1]. PhondPhunchongharn, DusitNiyato, Member, Ieee, Ekram Hossain, Senior Member, IEEE, And Sergio Camorlinga, "An Emi-Aware Prioritized Wireless Access Scheme For E-Health Applications In Hospital Environments", In IEEE Transactions On Information Technology In Biomedicine, Vol. 14, No. 5, September 2010, Pp.1247-1258.

[2]. M. Radha Rani, Vijetha Institute of Technology, G. Rajesh Kumar, G. PrasannaKumar, and Sciences, Vishnu Institute of Technology, "Implementation of March Algorithm Based MBISTArchitecture for SRAM" in International Journal of Advanced Research in Computer Engineering & Technology Volume 1, Issue 3, May2012 2015,pp.250-253.

[3]. T.V.Sirisha, II year, M.Tech VLSI system design, T. Vasu Deva Reddy, B.E,

M.Tech(Ph.D.), Department of ECE, AssociateProfessor, B V Raju Institute of Technology, Hyderabad, "DESIGN AND ANALAYSIS OF MARCH C ALGORITHM FORCOUNTER BASED MBIST
CONTROLLER" , International Journal For Technological Research In Engineering Volume 3, Issue 3, November-2015, pp.376-380.

[4]. DarsiKoteswaramma, K.MuraliKrishna, Sr.Assistant, Dr M .Sailaja,U.Yedukondalu.,"Memory Testing and Repairing Using MBIST with Complete Programmability" IOSR Journal of Electronics and Communication Engineering (IOSR-JECE),Volume 9, Issue 2, PP 80-83.