Built in Self-Test (BIST) for Digital Circuits

Mara Sandeep Kumar[1], Thagaram Nithyasruthi[2], Manda Rahul[3], Dr. V R Seshagiri Rao[4]
[1,2,3] Department of Electronics and Communication Engineering, Institute of Aeronautical Engineering, Hyderabad, India
[4]Ass. Prof. Department of Electronics and Communication Engineering, Institute of Aeronautical Engineering, Hyderabad, India

Abstract—Built-In Self-Test (BIST) is a technology that incorporates both hardware and software components, enabling electronic devices to conduct self-assessments. The primary objective of BIST is to develop circuits capable of autonomously testing themselves to identify faults. This study concentrates on improving the fault tolerance of memory testing within the BIST framework. Specifically, we will simulate the MARCH C- algorithm for memory testing on a Xilinx FPGA while implementing mechanisms to detect and correct various memory faults, including Stuck-at faults (SAF) and coupling faults. BIST is primarily used for continuous memory monitoring without interrupting system functionality, functioning as an online testing method. By integrating fault-tolerant strategies, we aim to enhance the robustness and reliability of memory testing processes, with potential applications in RFID integrity verification and avionics systems.

Keywords: MARCH C- algorithm, Xilinx, memory faults, fault-tolerant techniques.

I. INTRODUCTION

Embedded memories are experiencing exponential growth in both size and density, presenting a significant challenge for design architects due to the inherent complexity of their design structures and the heightened risk of manufacturing defects. Unlike other embedded cores in System-on-Chip (SOC) designs, embedded memories lack direct access, complicating the testing and diagnosis process. Additionally, the limited bandwidth of the SOC's main input and the embedded core often makes external access difficult. To address these challenges, there has been a surge of interest in self-testing methodologies for embedded arrays. The March test function has become a comprehensive troubleshooting tool that provides effective diagnostic tools for memory errors such as stuck errors and transfer errors. These tests involve writing information to each memory location and verifying it by reading it back. The pass or fail status is determined by whether the read-back values match the initially written data.

One notable advantage of March tests is their ability to achieve high fault coverage while maintaining linear test times relative to memory size, making them practical for industrial application. These algorithms excel in locating and identifying fault types. As very large-scale integration (VLSI) technology progresses, millions of transistors are now integrated onto a single silicon wafer, and the complexity of the wafer has increased exponentially. Consequently, robust and sophisticated test methods are essential to ensure manufacturing yield and product reliability. Manufacturing tests play a crucial role in improving yield rates and controlling production costs, this situation has increased further due to the increase in test data and test duration.

Existing System: An implementation of March Test Algorithms for supporting online memory testing has been developed to enhance the reliability and efficiency of memory testing in electronic systems. This method capitalizes on the idle state of the system or equipment, performing memory tests while waiting to be accessed by an interrogator. By leveraging idle periods, the system can continuously check the memory without interrupting its primary operations. This approach ensures that memory faults are detected and corrected promptly, maintaining the system's overall performance and reliability. The design of the current system includes a state-of-the-art self-testing machine (BIST) that determines the access process for memory testing. This state machine is responsible for generating test patterns, applying these patterns to the memory, and analyzing the output to identify any discrepancies.

Details of the transparent BIST circuit architecture to prevent integration of the test process with the physical operation. This transparent integration is necessary to maintain system performance when performing memory tests. A series of tests were performed and analyzed on existing systems to evaluate the level of the March test algorithm. These tests demonstrate the algorithm's ability to detect and detect memory errors. However, despite the success of the module-level evaluation, the system still faces problems in full use.

In particular, there are still some integration issues that need to be resolved before the MARCH C algorithm can operate efficiently and effectively as a BIST solution.

Broadcast System: Built-In Self-Test (BIST) is a technique that adds extra hardware and software features to electronic systems, allowing them to perform self-tests. The core concept of BIST is to develop circuits that can evaluate their own operation and identify faults. In this study, we focus on enhancing memory testing performance within the BIST framework by addressing the current system's limitations.

In particular, we use advanced error detection techniques to identify and fix various memory errors, including cluster errors (SAFs) and link errors, while simulating the MARCH C-algorithm for memory evaluation of Xilinx FPGAs. This approach enables the identification and fixing of errors by overcoming the half-baked measures of current systems. As an online test, BIST facilitates background checks in memory without affecting the actual performance of the system. This approach eliminates the dependency on idle states, allowing for consistent testing regardless of physical activity level. By incorporating forensic techniques such as error detection and error correction codes, the system aims to increase the robustness and reliability of the testing process.

II. LITERATURE SURVEY

Renju Thomas John, Sreekanth KD, Sivanantham S [1] concluded that low-cost measurement on transitions can be achieved by introducing a dynamic transition which can adjust the power measurement as a step-by-step process in the scan chain during the changeover. This results in a relationship with the activity vectors. This design improves the balance between measurement and power conversion. Power movement with the uncertainty of the test system is reduced, thus reducing the effect of high-power loss in the CUT. Since the switching controller occupies a portion of the dead space, overhead area overhead can be avoided.

The paper discusses how the BIST scheme effectively addresses testing challenges associated with VLSI devices, as highlighted by Costas Efstathiou and Voyiatzis Ioannis [2]. Input vector analysis in synchronous BIST allows for evaluation during the circuit's normal operation, eliminating the need for the circuit to be offline, thus mitigating issues faced by offline BIST techniques. The evaluation of these solutions is based on hardware overhead and CTL, which measures the time required to complete testing while the circuit is operational. In this introduction, we propose a novel synchronous BIST architecture that utilizes an SRAM cell-like structure to track input vector occurrences during normal operation. This new approach demonstrates greater efficiency compared to earlier vector strategies in terms of overhead and CTL during simultaneous BIST monitoring.

In the paper “An Efficient Parallel Transparent BIST Method for Multiple Memory Buffers,” Huang D.C, W. Jone, and S. Das [3] introduce a transparent BIST method capable of testing multiple memory arrays that are spatially distributed across a wafer. This approach involves creating a transparent connection through experimental design and response evaluation functions, which integrate a scan loop with several multiplexers, resulting in reduced hardware requirements. The test responses are assessed using an international MISR, minimizing energy consumption. Additionally, the authors propose a robust signature analysis technique that bypasses less effective signature estimation methods while maintaining low hardware costs. The TRSMarch walk was also developed based on historical memory data to generate test models and results effectively.

Nur Touba and Jinku Lee [4] explored the use of LFSR reseeding as an effective method for minimizing test storage. Their proposed coding scheme aims to lower the power consumption of LFSR reseeding while maintaining or even enhancing compression performance. The block size can be easily modified to further reduce power usage in electronic devices. This scheme is applicable in BIST environments or experimental compression strategies utilizing LFSR reseeding, helping to meet power efficiency requirements.

Dr. Senthil Kumar, G. Sathish, Ramesh and G. Sudhagar, [5] discuss a novel scanning architecture in their paper “Using Novel Architecture for VLSI Testing” that enhances power efficiency and reduces test time for System-on-Chip (SoC) testing. This architecture minimizes power consumption during testing by preventing changes in the scan chain from affecting the Circuit Under Test (CUT). Since there are no multiplexers in the main scan path, devices can be scanned at a higher clock rate. Their findings indicate that this design can also generate test vectors from various compressed data streams. Additionally, the structural design facilitates compression, which can significantly shorten test times and decrease reliance on Automatic Test Equipment (ATE). The model also evaluates runtimes based on appropriate and inappropriate changes relative to the number of indicators.

III. METHODOLOGY AND IMPLEMENTATION

This approach includes the selection of the MARCH C algorithm for memory evaluation and evaluation of violations such as parity and error correction codes. We seamlessly integrate these components into the BIST framework to optimize the algorithms for Xilinx FPGAs. Tests to ensure the performance and performance of the system under different conditions and to increase the reliability of the memory test system. With the development of fault tolerance, we aim to increase the power and reliability of electronic systems with RFID integrity checks and avionics applications.

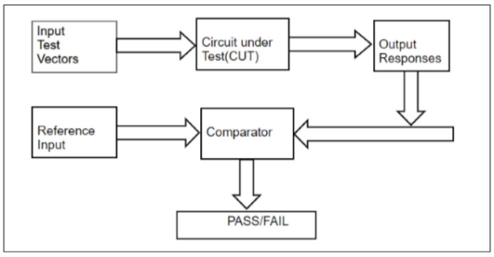

Fig 1: Basic Architecture

The concept of test generation is illustrated in Figure 1. The input test vector serves as a binary model for the inputs of the Circuit Under Test (CUT), while the output reflects the values obtained from the CUT. To validate the test, a response comparison method is employed to ensure that the output matches the expected results. If all responses align with the correct output data, the CUT is considered to have passed the test and is deemed fault-free. Various techniques that utilize test vectors are applied to the CUT, focusing on how the responses are evaluated against expected outcomes.

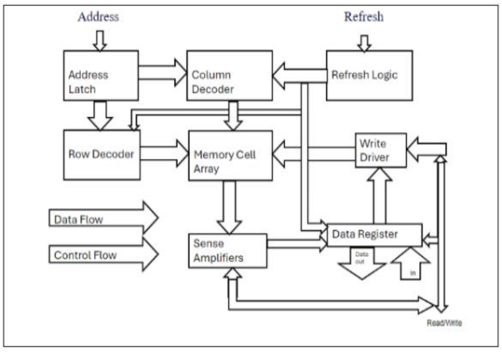

Fig 2: Functional Memory Module

There are two primary approaches for electrical testing: external testing with automatic test equipment (ATE) and internal testing utilizing Built-In Self-Test (BIST). In external testing, input test vectors and corresponding correct response data are stored in the memory of the ATE. Test Vector Generation (ATPG) tools are employed to create these input test vectors, while accurate response data is derived from circuit simulations.

There are two main approaches to memory testing: electronic (technology-based) and functional (technology-independent). Electronic memory testing involves a device that conducts various measurements, including DC and AC testing, IDDQ testing, and assessments for recovery, hold, and unbalance conditions. DC and AC parametric tests ensure that the device adheres to electrical specifications, such as voltage and current levels, and that dead pins are properly adjusted. Since embedded memory in a System-on-Chip (SoC) typically lacks I/O ports directly connected to chip pins, parametric evaluation is generally unnecessary for this type of memory. IDDQ and dynamic testing require detailed procedural descriptions. This paper emphasizes technology-independent testing of operational memory, focusing on analyzing memory behavior. Memory performance testing is recognized in the industry as a cost-effective solution for evaluating short-term algorithms, as it does not demand extensive memory knowledge during the testing process. This section outlines the theoretical background and describes the experimental model for working memory and the March algorithm.

ALGORITHM	DESCRIPTION
MATS	{ ⇕(w0); ⇕ (r0,w1); ⇕(r1) }
MATS+	{ ⇕(w0); ⇑ (r0,w1); ⇓(r1,w0) }
MATS++	{ ⇕(w0); ⇑ (r0,w1); ⇓ (r1,w0,r0) }
MARCH X	{ ⇕(w0); ⇑ (r0,w1); ⇓ (r1,w0); ⇕ (r1,w0); ⇕ (r0)}
MARCH C-	{ ⇕(w0); ⇑ (r0,w1); ⇑ (r1,w0); ⇓ (r0,w1); ⇓ (r1,w0); ⇕ (r0) }
MARCH A	{ ⇕(w0); ⇑ (r0,w1,w0,w1); ⇑ (r1,w0,w1); ⇓ (r1,w0,w1,w0); ⇓ (r0,w1,w0) }
MARCH Y	{ ⇕(w0); ⇑ (r0,w1,r1); ⇓ (r1,w0,r0); ⇕ (r0) }
MARCH B	{ ⇕(w0); ⇑ (r0,w1,r1,w0,r0,w1); ⇑ ⇓ (r1,w0,w1); (r1 ⇓ w0,w1,w0); (r0,w1,w0) }

Table 1: Irredundant MARCH Algorithms

Numerous algorithms have been designed to assess semiconductor signals, with the March test being one of the most widely recognized and effective. The March test includes a series of read and write operations that must be executed on each memory cell. It is capable of detecting various fault patterns, including Stuck-at Faults (SAF), Address Faults (AF), and some Coupling Faults (CF). The operations performed on the cells include writing zero (w0), writing one (w1), reading zero (r0), and reading one (r1). The read operation verifies whether the value stored in a cell matches the expected value. Cells can be accessed in either ascending or descending order. An example of a RAM testing algorithm is MATS++, which is also applicable to EEPROM testing. It comprises three March elements: M0: ↕ (w0); M1: ↑ (r0; w1); and M2: ↓ (r1; w0; r0), with elements separated by commas or semicolons, and the entire sequence enclosed in braces.

IV. RESULTS AND DISCUSSIONS

For Executing the project, these steps are involved: Install necessary software tools (e.g., Xilinx Vivado for FPGA design and implementation). Set up the development environment, including any required IDEs and libraries. Create HDL Modules: Write the Hardware Description Language (HDL) code (Verilog or VHDL) for the BIST circuit. This includes: Circuit Under Test, Response Analyzer, Test Pattern Generator Verifies functionality of the BIST design using simulation tools such as ModelSim. Apply the comparison to the CUT and compare the output to the expected results. Debug and refine the HDL code based on simulation results. Below are the results for SRAM1 and SRAM2 using MARCH4n Algorithm. The SRAM 1 module is designed as fault free such that while testing no errors will occur.

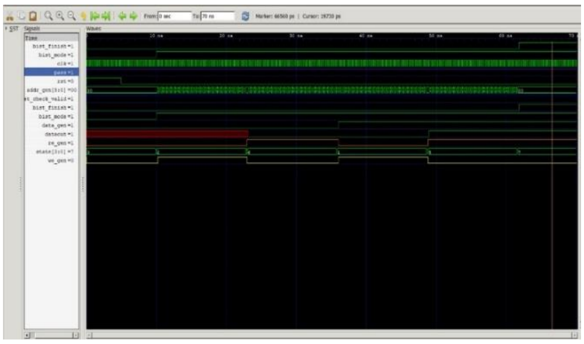

Fig 3:SRAM 1 using MARCH 4n

The faulty memory is SRAM 2, designing of SRAM 2 is consisting of fault addresses to show the error while testing using Bist controller.

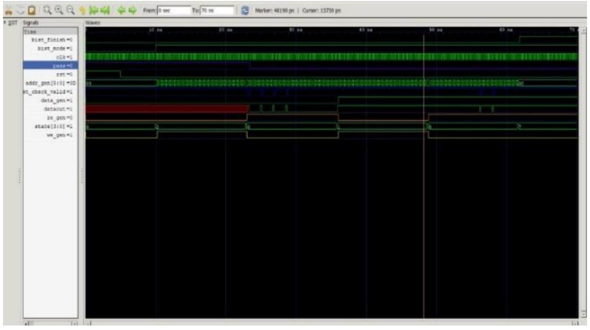

Fig 4: SRAM 2 using MARCH 4n

When the pass value is 1 then there is no error, if not the memory module consists of errors in MARCH 4n algorithm. When the repair-success value is 1 then there is no error, if value obtained is 0 then result in error in MARCH 6n algorithm. MARCH 6n Algorithm uses 6 memory cells for READ and WRITE operations. Also, same memory modules are used (SRAM1 & SRAM2). SRAM1 is fault free and SRAM2 is faulty.

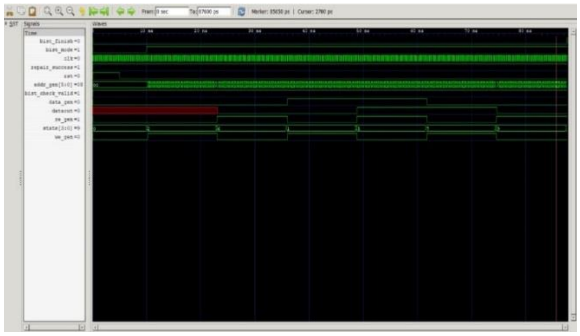

Fig 5: SRAM 1 using MARCH 6n

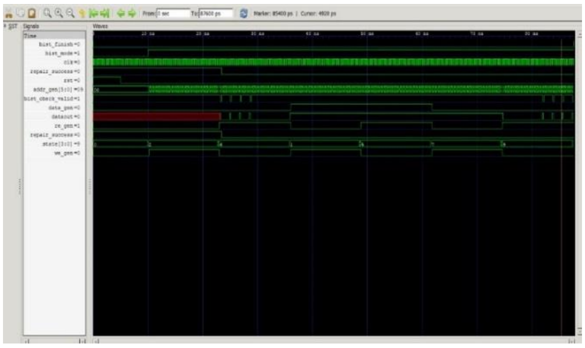

Fig 6: SRAM 2 using MARCH 6n

V. CONCLUSION

We Additionally have tested SRAM circuits using the March 6 algorithm to verify the effectiveness of our approach. In our tests, SRAM1 was identified as fault-free, while SRAM2 was detected as faulty. The Verilog analysis output was crucial in this verification process, where a repair success of 1 indicated no error, and a repair success of 0 indicated the presence of an error. This detailed analysis demonstrated the capability of the March 6 algorithm to detect and report faults accurately. The detection of errors in SRAM2 highlighted the algorithm's precision and reliability, ensuring thorough fault coverage. Ensuring fault-free operation in SRAM1 validated the robustness of our BIST implementation, confirming that the system can effectively distinguish between functional and defective memory units. This testing phase provided a clear distinction between fault-free and faulty memory modules, reinforcing the system's diagnostic capabilities. The successful identification and correction of memory faults are critical for maintaining system reliability, especially in applications requiring high dependability. The BIST approach, combined with the March 6 algorithm, offers a comprehensive solution for memory fault detection and correction. This integrated solution not only detects but also addresses various memory faults, enhancing overall system performance. By accurately identifying and correcting faults, the system ensures that memory components operate correctly, reducing the risk of system failures. This increased reliability is essential for critical applications such as RFID integrity checks and avionics systems. The results from testing SRAM circuits confirm that the March 6 algorithm is effective in a practical implementation. The precise fault detection and correction mechanism bolsters confidence in using BIST for real-time memory testing. This approach ensures continuous background checking without interrupting the system's primary functions. Ultimately, the integration of BIST and the March 6 algorithm significantly enhances the robustness and reliability of memory testing processes.

Future Scope of Study: Building upon the successful implementation and testing of the SRAM circuits using the March 6 algorithm, several future research and development directions can be explored to further enhance the robustness and applicability of this approach. 'Scalability to Larger Memory Systems' Expanding the March 6 algorithm and BIST (Built in Self-Test) framework to handle larger and more complex memory systems, such as DRAM and Flash memory, ensuring the same level of fault detection and correction efficiency. 'Integration with Advanced Technologies' Investigating the integration of this fault detection and correction mechanism with emerging memory technologies like MRAM (Magneto resistive RAM) and ReRAM (Resistive RAM), to ensure compatibility and effectiveness in future memory architectures. 'Adaptive Testing Algorithms' Developing adaptive and machine learning-based algorithms that can dynamically adjust testing patterns and strategies based on real-time analysis of memory behavior, further improving fault coverage and reducing testing time. 'Enhanced Fault Diagnosis and Prediction' Incorporating predictive analytics to forecast potential memory failures before they occur, allowing for preemptive maintenance and reducing unexpected system downtimes. Low-Power Testing Solutions: Designing low-power BIST implementations to cater to energy-constrained environments, such as IoT devices and wearable technology, ensuring reliable memory testing without significant power overhead. 6. Extended Fault Coverage: Extending the fault model coverage to include more complex fault types such as dynamic faults, retention faults, and soft errors caused by environmental factors like radiation, which are crucial for space and military applications. 7. Multi-Layered Memory Systems: Applying this fault detection and correction framework to multi-layered memory systems used in modern computing architectures, ensuring comprehensive fault coverage across different memory hierarchies.

REFERENCES

[1] R. T. John, K. D. Sreekanth and S. Sivanantham, "Adaptive Low Power RTPG for BIST based test applications," 2013 International Conference on Information Communication and Embedded Systems (ICICES).

[2] Ioannis Vogiatzis, C. Efstathiou, "Input vector monitoring concurrent BIST architecture using SRAM cells," July 2014IEEE Transactions on Very Large Scale Integration (VLSI) Systems.

[3] D. C. Huang, W. B. Jone and S. R. Das, "An efficient parallel transparent BIST method for multiple embedded memory buffers," VLSI Design 2001. Fourteenth International Conference on VLSI Design.

[4] Jin-Kyu Lee, Nur A. Touba, "LFSR-Reseeding Scheme Achieving Low-Power Dissipation During Test" March 2007IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems.

[5] Sudhagar Govindh swamy, S. Senthil Kumar, Ramesh G, G. Sathish Kumar, "Implementation of a novel architecture for VLSI testing" January 2013 Conference: Emerging Trends in VLSI, Embedded System, Nano Electronics and Telecommunication System (ICEVENT).