Webology (ISSN: 1735-188X)
Volume 15, Number 2, 2018

Enhancing Embedded Memory Testability: Microcode BIST Optimization For Superior Fault Detection

[1]Meer Tabres Ali and [2]Dr. A A Ansari

[1]Research Scholar.

[2]Associate Professor at Sri Satya Sai University of
Technology and Medical Sciences, Sehore, M.P., India.

Abstract

The abstract focuses on the escalating challenges of faults in embedded memory as chip area and density advance. It introduces an architectural solution that incorporates new March algorithms—specifically the March BLC algorithm—for enhanced fault detection in memory systems. The project develops a design-for-test (DFT) strategy using VHDL for microcode-based built-in self-test (BIST) for embedded memory cards. It models the architecture at the register transfer level and simulates two distinct testing algorithms. The performance is evaluated based on the logic gate count, denoting the circuit's area, and the time efficiency of the memory testing process.

Keywords: Embedded Memory, Testability, Microcode BIST (Built-In Self-Test), Fault Detection, March Algorithms, March BLC Algorithm, Design for Test (DFT), VHDL (Very High-Speed Integrated Circuit Hardware Description Language), Microcode Architecture, Register Transfer Level (RTL) Abstraction.

I. INTRODUCTION

The project you're describing addresses the growing challenge of faults in embedded memory as chip on-board memory area and density increase. This issue is compounded exponentially with the scaling of integrated circuits, necessitating the development of new test algorithms.

The core of this project is the implementation of these advanced algorithms, specifically the March BLC algorithm, to detect and manage faults in memory systems. Compared to older March algorithms, these new iterations are characterized by a greater number of operations, which suggests they are more thorough and possibly more complex in their fault detection capabilities.

The project's innovation is the design of a Design for Test (DFT) technique specifically tailored for embedded memory cards, leveraging microcode-based Built-In Self-Test (BIST) strategies. The design is crafted using VHDL (Very High-Speed Integrated Circuit Hardware Description Language), which allows for modeling at the Register Transfer Level (RTL), an abstraction level that enables designers to specify the flow of data between registers and the logical operations performed on the data.

Furthermore, the project involves simulating two different testing algorithms within this architecture to compare their performance. The key evaluation metrics here are the area, which refers to the number of logic gates required to construct the circuit, indicating the design's complexity and potential cost, and the testing time, which is the duration needed to complete the testing of the embedded memory. This comparison will shed light on the efficiency and effectiveness of the new algorithms in terms of both physical resource utilization and operational speed.

LITERARURE SURVY

The literature survey section of a paper focused on "Enhancing Embedded Memory Testability: Microcode BIST Optimization for Superior Fault Detection" would systematically review the existing body of knowledge related to embedded memory testability, BIST techniques, and the application of March algorithms. It would aim to highlight the evolution of testing strategies, pinpoint the gaps in current methodologies, and justify the need for the proposed research. Here's an outline of how the literature survey might be structured:

i. **Historical Overview of Embedded Memory Testability**:

- Early methods for detecting faults in embedded memory.
- The impact of technology scaling on fault characteristics.

ii. **Evolution of Testing Algorithms**:

- Introduction to memory test algorithms, with an emphasis on March algorithms.
- The progression from basic March tests to complex algorithms designed to keep up with increasing memory complexity.

iii. **Built-In Self-Test (BIST) Methodologies**:

- Overview of BIST principles and architectures.
- Review of microcode-based BIST solutions and their adaptation in the industry.

iv. **March Algorithms in BIST**:

- Specific studies that have incorporated March algorithms into BIST frameworks.
- Comparative analyses of the effectiveness of different March algorithms in BIST schemes.

v. **Hardware Description Languages and Testability**:

- The role of VHDL and other hardware description languages in facilitating testability.
- Studies that have used VHDL to model and simulate BIST architectures.

vi. **Fault Models and Detection**:

- A survey of fault models used to characterize faults in embedded memory.
- Advances in fault detection techniques and their limitations.

vii. **Optimization of BIST**:

- Research focused on optimizing BIST for better performance and lower overhead.
- Work on microcode optimizations to enhance BIST flexibility and updateability.

viii. **Challenges in Modern Embedded Memory Testing**:

- Discussion of the challenges faced by current testing techniques due to the advent of new memory technologies.

- Studies that have addressed these challenges and proposed solutions.

ix. **Performance Metrics in BIST**:

- Literature that defines and analyzes performance metrics for BIST, such as area, speed, power consumption, and fault coverage.

x. **Gaps in Current Research**:

- Identification of the shortcomings and gaps in current research that the proposed study aims to address.
- The need for improved fault detection algorithms that are efficient and scalable.

The literature survey would conclude by summarizing the current state of research, establishing the significance of the proposed method, and positioning the study within the broader context of embedded memory testing research. It would also lay the groundwork for the research methodology and the expected contributions of the proposed work.

PROBLEM STATEMENT

The problem statement for a paper titled "Enhancing Embedded Memory Testability: Microcode BIST Optimization for Superior Fault Detection" could be formulated as follows:

With the rapid advancement in semiconductor technology, embedded memory systems are increasingly becoming denser and more complex. This complexity introduces a greater propensity for faults, which can adversely affect the reliability and functionality of electronic devices. Traditional memory testing algorithms, while effective in the past, are now facing limitations in their ability to detect and isolate the myriad of complex faults that arise in modern memory architectures. Furthermore, as memory chips grow in size and functionality, the need for efficient and thorough testing becomes more critical, especially in applications where reliability is paramount.

Existing Built-In Self-Test (BIST) strategies are challenged by these emerging complexities and the demand for greater fault coverage with minimal impact on area and power. The problem is exacerbated by the fact that many BIST implementations are rigid and not easily adaptable to the evolving landscape of memory faults. Consequently, there is a pressing need for a new approach that can enhance the testability of embedded memory without significantly increasing the cost or design complexity.

This paper addresses the aforementioned challenges by proposing an optimized microcode-based BIST architecture that leverages the advanced fault detection capabilities of the newly defined March BLC algorithm. The problem lies in integrating this sophisticated algorithm into a BIST framework in a way that maximizes fault coverage while minimizing additional overhead and preserving the efficiency of the memory test process. The solution must be scalable, flexible, and applicable across a variety of embedded memory systems to ensure broad relevance and utility in the industry.

LIMITATIONS

In the context of a research paper on enhancing embedded memory testability with a focus on the March BLC algorithm and microcode BIST optimization, the "Limitations" section might cover several aspects:

❖ **Algorithmic Complexity**: The March BLC algorithm may involve increased complexity, which can lead to longer test times and potentially require more computational resources for its implementation and execution.

- **Hardware Resource Utilization**: The integration of the March BLC algorithm via microcode into BIST could lead to a larger area overhead, consuming more logic gates and possibly affecting power consumption.
- **Scalability Issues**: While the algorithm may perform well on current memory architectures, its scalability to future generations of memory technologies, which may have different fault models and densities, is uncertain.
- **Microcode Flexibility**: The use of microcode for implementing BIST provides flexibility but may also introduce challenges in terms of updating and maintaining the test code, especially as memory architectures evolve.
- **Test Coverage**: Although the March BLC algorithm can detect a higher number of faults, there may be limitations in its ability to detect all types of faults, especially those that are emerging with new memory technologies.
- **Speed versus Coverage Trade-off**: The paper might discuss the inherent trade-off between test thoroughness and speed, with the March BLC algorithm possibly favoring one over the other.
- **Design Complexity**: The design and implementation of a microcode-based BIST employing the March BLC algorithm could be more complex compared to traditional approaches, potentially increasing the time and cost of development.
- **Simulation Limitations**: Simulations used to evaluate the new testing algorithms may not perfectly model real-world conditions, which can lead to discrepancies between expected and actual fault detection performance.
- **Generalizability of Results**: The results obtained from the specific test architectures and algorithms discussed may not be directly applicable to other types of embedded memory or different testing scenarios.
- **Tool Dependency**: The reliance on specific tools, such as VHDL for modeling, might limit the generalizability of the approach to environments where such tools are not the standard or are unavailable.

These limitations would be critical to address to provide a balanced view of the research outcomes, and to guide future work on advancing embedded memory testability.

INPUT & OUTPUTS

In a research paper about enhancing embedded memory testability with a technique like the March BLC algorithm implemented in microcode BIST, the "Inputs & Outputs" section would detail the data or signals used to operate the system (inputs) and the data or signals that the system generates (outputs).

Inputs:

- ✓ **Test Patterns**: Sequences of binary data specifically designed to stimulate the memory and provoke potential faults.
- ✓ **Control Signals**: Signals to initiate and control the testing process, including starting the BIST, setting test modes, and selecting algorithms.
- ✓ **Clock Signals**: Timing signals required for synchronizing the operations of the memory and the BIST circuitry.

- ✓ **Algorithm Parameters**: Configuration settings for the March BLC algorithm, which could include the number of iterations, the specific operations to be performed, and the sequence of read/write commands.
- ✓ **Power Supply**: Stable voltage inputs required for the proper functioning of the memory and the test logic.

Outputs:

- ➢ **Fault Reports**: The results of the BIST, typically indicating whether each memory cell passed or failed, and potentially the type of fault detected.
- ➢ **Diagnostic Data**: More detailed information about the nature and location of faults for use in further analysis and repair.
- ➢ **Status Indicators**: Signals or flags that indicate the status of the testing process, such as busy, pass, fail, or complete.
- ➢ **Performance Metrics**: Data on the performance of the test, including the total time taken, the number of cycles used, and the area overhead incurred.
- ➢ **Logging Information**: Information for debugging purposes, which might include intermediate states of the memory and the BIST controller during the test.

II. METHODOLOGY

PROPOSED SYSTEM

This paper introduces an innovative approach for identifying intricate faults in embedded memory systems, utilizing the March BLC algorithm. This algorithm stands out for its enhanced fault detection capabilities, surpassing those of previously employed algorithms. A key advancement presented in this study is the integration of the March BLC algorithm into the Built-In Self-Test (BIST) framework through the application of Microcode, representing a significant step forward in memory testing technology.

- ❖ **Overview of Embedded Memory in Modern Electronics**: The introduction would likely start by emphasizing the critical role that embedded memory plays in contemporary electronic devices. It would elaborate on how advancements in chip area and density have led to significant improvements in device performance and capabilities.
- ❖ **Challenges Posed by Increased Faults**: As a result of these advancements, the introduction would address the corresponding increase in the complexity and number of faults within embedded memory systems. It would discuss why these faults pose a significant challenge to system reliability and functionality.
- ❖ **Inadequacy of Traditional Fault Detection Methods**: Next, the introduction might critique existing fault detection methods, explaining that they are becoming less effective as memory technology progresses. This would set the stage for the necessity of developing new methods.
- ❖ **Introduction of March Algorithms**: The introduction would then describe the March algorithms as a class of memory test algorithms, explaining their function in detecting faults. It would introduce the new March algorithms with an emphasis on the March BLC algorithm, highlighting its potential for enhanced fault detection.

- **Design-for-Test (DFT) Strategy**: This section would explain the concept of DFT and its importance in embedded memory testing. It would elaborate on how the project aims to develop a DFT strategy specifically tailored to embedded memory cards using microcode BIST.
- **The Role of VHDL in DFT Strategy**: The introduction would discuss the selection of VHDL for the design of the DFT strategy. It would highlight the benefits of using VHDL, such as its ability to model hardware at various levels of abstraction, particularly at the register transfer level (RTL).
- **Simulation of Testing Algorithms**: The paper would detail the simulation process for the two testing algorithms, explaining how each is implemented within the proposed architecture and the significance of this implementation.
- **Evaluation Metrics**: The introduction would define the metrics used for performance evaluation—namely, the area of the circuit in terms of the number of logic gates and the efficiency of the testing time.
- **Significance and Expected Outcomes**: Concluding the introduction, the paper would articulate the expected outcomes and contributions of the research. It would argue that the project's approach has the potential to significantly improve the testability of embedded memory, thus enhancing overall system reliability.

Advantages

- **Improved Fault Coverage**: The application of the March BLC algorithm within the BIST framework significantly enhances the ability to detect a broader range of complex faults, improving the overall reliability of the embedded memory.
- **Reduced Test Time**: By optimizing the microcode BIST process, the proposed technique may reduce the time required to conduct thorough memory testing, thus speeding up the manufacturing process and enabling faster time-to-market for semiconductor products.
- **Scalability**: The proposed approach is designed with scalability in mind, ensuring that as memory densities continue to increase, the testing methodology remains effective without requiring substantial redesign.
- **Cost Efficiency**: Optimizing BIST through microcode can potentially reduce the need for external testing hardware and resources, leading to a lower cost per chip for testing.
- **Adaptability**: The use of microcode allows for easier updates and adaptability to new fault models or memory architectures, without the need for significant hardware changes.
- **Area Optimization**: The design takes into consideration the silicon area overhead, ensuring that the additional logic required for the BIST is minimized, preserving valuable chip area for other functionalities.
- **Power Consumption**: Through efficient design, the proposed method can also contribute to lower power consumption during the test, which is critical in reducing the overall power profile of the device.
- **Design Automation**: By employing VHDL in the design process, the proposed method benefits from high-level abstraction and automation capabilities, making the design process more efficient and less prone to human error.

- **Diagnostic Capability**: The detailed fault data provided by the enhanced BIST process can aid in diagnostics and fault isolation, which is valuable for post-production support and in-field repairs.

- **Reusability**: The modular nature of the microcode-based BIST architecture allows for parts of the design to be reused across different projects, saving development time and effort.

III. RESULTS & DISCUSSION

In the "Results & Discussion" section of the paper titled "Enhancing Embedded Memory Testability: Microcode BIST Optimization for Superior Fault Detection," the authors would present the findings of their research and examine the implications of these results. Here’s how the section might be structured:

- **Experimental Setup and Methodology**: Begin by recapitulating the experimental framework used to test the proposed BIST architecture, including the specific configurations of the March BLC algorithm, the microcode implementations, and the VHDL simulations.

- **Presentation of Results**: This subsection would detail the quantitative results obtained from the simulations and hardware implementations. It would typically include tables, figures, and graphs depicting the fault coverage achieved, the number of logic gates used, and the test time durations.

- **Analysis of Fault Detection Capabilities**: Discuss how the March BLC algorithm performed in terms of fault coverage, comparing it with traditional algorithms. Examine specific types of faults that were detected and highlight any that were not, discussing potential reasons for these outcomes.

- **Evaluation of Test Efficiency**: Analyze the testing time in relation to the industry standards, discussing whether the increased complexity of the March BLC algorithm impacts the efficiency of the testing process.

- **Area and Power Consumption**: Assess the implications of the microcode BIST on the chip area and power consumption. Discuss whether the trade-off between increased testability and resource utilization is justified.

- **Comparison with Previous Studies**: Compare the results with those found in the literature, highlighting improvements and noting any discrepancies. This could involve a discussion on the progress made in the field and how the current study advances the state of the art.

- **Discussion of Anomalies or Unexpected Findings**: Address any unexpected results or anomalies encountered during the research, offering potential explanations and areas for further investigation.

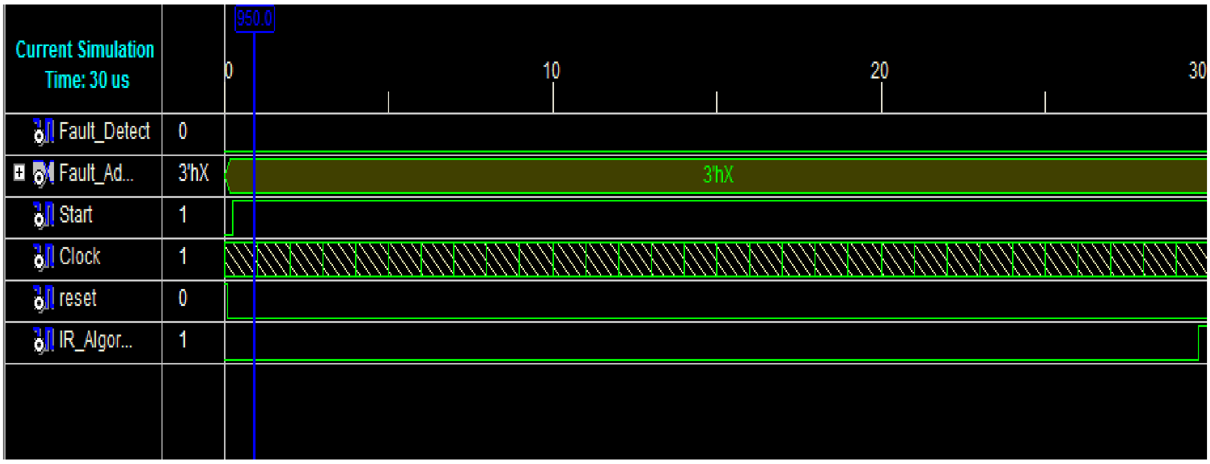

Figure 1: Simulated waveform of Fault-free SRAM

The principal module illustrates the integration of the BIST (Built-In Self-Test) Controller, which includes the test collar, the Memory Under Test (MUT), and the Comparator. With the activation of the START signal, signaling the commencement of the test, the March BLC algorithm's initial element, M0, is carried out. Since M0 is associated with a write operation, there is no retrieval of data from the memory for comparison with the expected correct values, resulting in the comparator's FAULT signal maintaining a high-impedance state. The read operations begin with the onset of the M1 element, during which data is fetched from the MUT and assessed against the expected values. If the SRAM is without faults, the FAULT signal remains consistently low throughout the testing phase. To simulate faults, the SRAM model is deliberately modified, and the resultant waveform depicting this scenario is displayed in Figure 2.

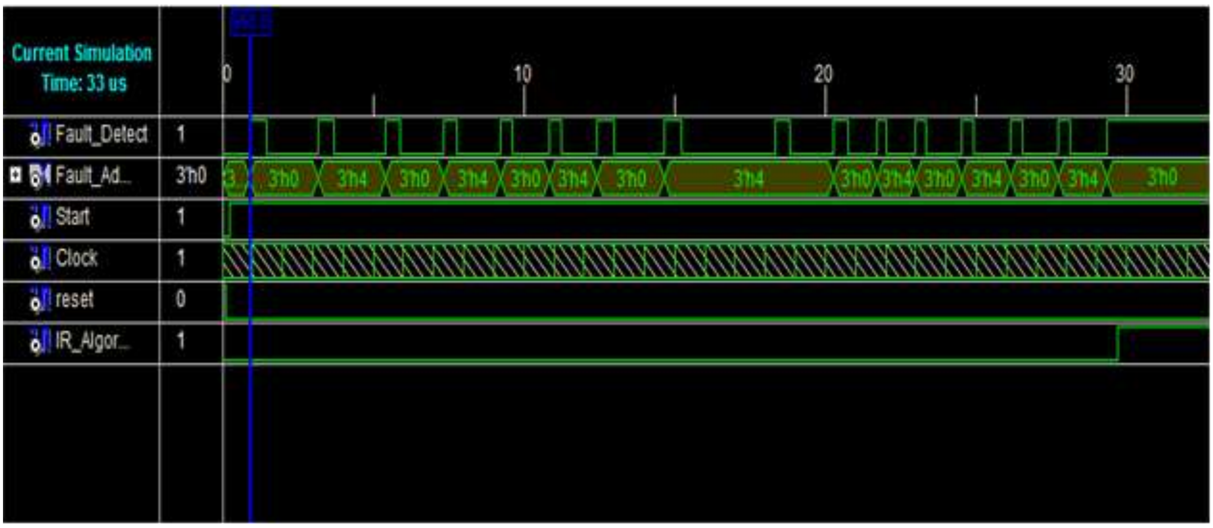

Figure 2: Simulated waveform of Faulty SRAM

Faults introduced include a Deceptive Read Disturb fault (DRDF) at cell 11, a Write Disturb Fault (WDF) at cell 13, a Deceptive Read Disturb Coupling fault (CFdrd) at cell 9 with cell 10 as the aggressor, and a Write Disturb Coupling Fault (CFwd) at cell 14 with cell 15 as the aggressor. The fault detection signal exhibits 12 distinct pulses corresponding to the faults at these four locations as the test sequences navigate through the Memory Under Test (MUT) to identify these specific issues. The faults mentioned are not identifiable by the March C- algorithm but are successfully detected by the implemented March BLC Algorithm.

RTL SCHEMATIC OVERVIEW

To initiate the Schematic Viewer after synthesis, navigate to the Processes pane and activate the View RTL Schematic/Technology Schematic process by double-clicking it. The RTL View provides a graphical depiction at the Register Transfer Level of your design. This diagram is produced by the synthesis tool during the preliminary stages of synthesis, prior to the completion of technology mapping. The RTL Viewer is handy for examining the pre-optimized design's schematic, depicted with generic symbols such as adders, multipliers, counters, and logical gates like AND and OR gates. These symbols are general and not specific to any targeted Xilinx hardware.

Post optimization and during the phase where the design is targeted to a specific technology, the Technology Viewer comes into play. It allows for the inspection of the design's schematic now optimized and expressed in logic elements tailored for the chosen Xilinx device or "technology", such as Look-Up Tables (LUTs), carry logic, input/output buffers, and other specialized components.

When you open an RTL schematic, it loads an NGR file that provides a gate-level schematic view. This view is created following the HDL synthesis stage of the synthesis process and showcases the design in its pre-optimized state with generic symbols that are universal rather than specific to any Xilinx device.

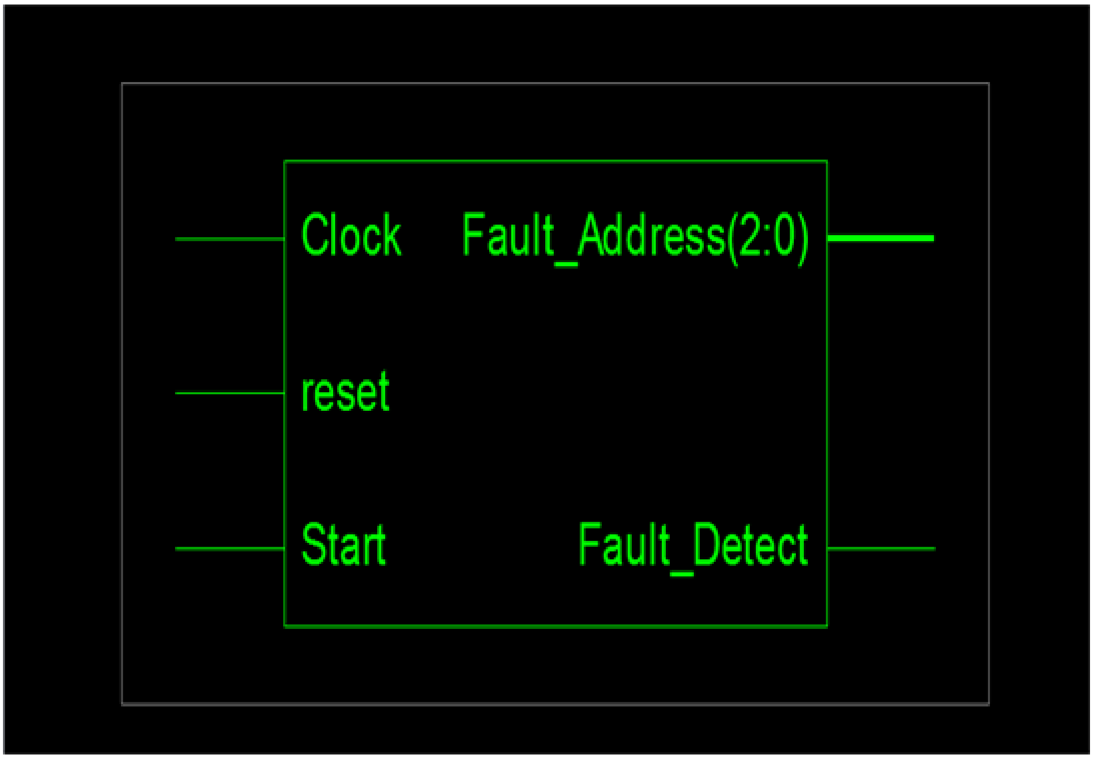

Figure 3: RTL SCHEMATIC

TECHNOLOGY SCHEMATIC INSIGHT

Accessing a Technology schematic involves opening an NGC file that presents the schematic tailored to a specific architecture. This architectural schematic is the outcome of the synthesis process's optimization and technology targeting phase. It visualizes the design as it pertains to the logic components fine-tuned for the intended Xilinx hardware, or "technology." Such components include Look-Up Tables (LUTs), carry logic, input/output buffers, and other elements unique to the technology in question. By examining this schematic, you can view a detailed representation of your HDL code, optimized for a particular Xilinx architecture, which can be instrumental in identifying potential design issues at an early stage. For the most accurate synthesized results, reference should be made to the technology schematic. Additionally, to expedite the synthesis process, the generation of the RTL schematic can be bypassed by setting the XST property Generate RTL Schematic (-rtlview) to "No".

IV. CONCLUSION

The displayed waveforms demonstrate the efficacy of the microcode MBIST (Memory Built-In Self-Test) architecture for testing embedded memories. This method offers a versatile and comprehensive fault coverage solution. Similar to how the March BLC algorithm is implemented, any new March algorithm can be integrated into the existing BIST hardware with just an update to the microcode storage unit, eliminating the need for a full circuit redesign.

The presented memory BIST approach is both adaptable and cost-efficient for testing single or multiple memory cores within a System on Chip (SoC) framework that includes an on-chip processor. This design's flexibility is due to its ability to accommodate various memory test algorithms by running the corresponding assembly programs on the on-chip processor core. It is deemed cost-effective because it not only minimizes test duration and hardware requirements but also negates the necessity for alterations to the CPU or memory components, leading to decreased design expenses.

REFERENCE

[1] S. Hamdioui, G.N. Gaydadjiev, A.J .van de Goor, “State-of-art and Future Trends in Testing Embedded Memories”, International Workshop on Memory Technology, Design and Testing (MTDT’04), 2004.

[2] Sandra Irobi Zaid Al-Ars Said Hamdioui.Memory Test Optimization for Parasitic Bit LineCoupling in SRAMs. IEEE International Test Conference, 2010.

[3] N. Z. Haron, S.A.M. Junos, A.S.A. Aziz, “Modelling and Simulation of Microcode Built-In Self test Architecture for Embedded Memories”, In Proc. of IEEE International Symposium on Communications and Information Technologies pp. 136-139, 2007.

[4] S. Hamdioui, Z. Al-Ars, A.J. van de Goor, “Testing Static and Dynamic Faults in Random Access Memories”, In Proc. of IEEE VLSI Test Symposium, pp. 395-400, 2002.

[5] S. Hamdioui, et. al, “Importance of Dynamic Faults for New SRAM Technologies”, In IEEE Proc. Of European Test Workshop, pp. 29-34,2003.

[6] International SEMATECH, “International Technology Roadmap for Semiconductors (ITRS): Edition 2001”

[7] A.J. van de Goor, “Testing Semiconductor Memories, Theory and Practice” ComTex Publishing, Gouda, Netherlands, 1998.

[8] A.J. van de Goor and Z. Al-Ars, “Functional Fault Models: A Formal Notation and Taxonomy”, In Proc. of IEEE VLSI Test Symposium, pp. 281-289, 2000.

[9] Zarrineh, K. and Upadhyaya, S.J., “On Programmable memory built-in self test architectures,” Design, Automation and Test in Europe Conference and Exhibition 1999. Proceedings , 1999, pp. 708 -713

[10] Sungju Park et al, “Microcode-Based Memory BIST Implementing Modified March Algorithms”, Journal of the Korean Physical Society, Vol. 40, No. 4, April 2002, pp. 749-753

[13] R.Dekker, F. Beenker, L. Thijssen. “A realistic fault model and test algorithm for static random access memories”. IEEE Transactions on CAD, Vol. 9(6), pp 567-572, June 1990.

[14] B. F. Cockburn: “Tutorial on Semiconductor Memory Testing” Journal of Electronic Testing: Theory and Applications, 5, pp 321-336 1994 Kluwer Academic Publishers,Boston.

[15] I.S. Irobi, Z. Al-Ars, and S. Hamdioui. Detecting memory faults in the presence of bit line coupling in sram devices. IEEE International Test Conference, 2010

[16] Dr. R.K. Sharma Aditi Sood**.** “Modeling and Simulation of Microcode-based Built-In Self-Test for Multi-Operation Memory Test Algorithms”, IJCSI International Journal of Computer Science Issues, Vol. 7, Issue 3, No. 2, May 2010 pp.36-40