Research Paper Engineering

Implementation of Finite State Machine (FSM) Based March SS Algorithm for Testing of Memory

* Yogendra Yadav ** Neha Sharma
*** Shraddha Bansal

* Assistant Professor, Electronics & Communication Department, Prestige Institute of Engineering & Science, Indore (M.P.)

** Assistant Professor, Electronics & Communication Department, Prestige Institute of Engineering & Science, Indore (M.P.)

*** Assistant Professor, Electronics & Communication Department, Prestige Institute of Engineering & Science, Indore (M.P.)

ABSTRACT

As the number of components increases then the memory testing becomes very complicated task. There are several testing methods are used presently but The Built-In Self-Test (BIST) approach is most popular and widely used approach. Built-In Self-Test approach provides a cost effective solution for testing of the embedded memory. There are two architectures are used now-a-days for testing the embedded memory, which are FSM based memory BIST and Microcode based memory BIST. In these architectures March algorithms are widely used because March algorithms provides high fault coverage and easy to implement with these architectures. In this paper we discuss about implementation of March SS algorithm for memory BIST architecture based on finite state machine. We use Xilinx ISE 8.2i Tool for writing VHDL code of March SS algorithm for Memory BIST architecture. We use ModelSim SE 6.3f Tool for simulation purpose.

Keywords: Built-In Self-Test (BIST), Finite State Machine (FSM), VHDL, March Algorithm, Test Algorithms.

I. Introduction

The Memory BIST approach is commonly used because it provides efficient, less costly and speed testing solution for memory. The Built-In Self-Test approach eliminates the need of external test equipment. The memory BIST provides several advantages such as high fault coverage, high speed testing, low area and low cost than other testing methods. There are several testing algorithms are used for memory testing but The March Algorithms are preferred over other Test algorithms because March algorithms are easy to implement with BIST and better Fault coverage. The FSM based memory BIST having advantage of speed testing, small area and compact but less flexibility.

Basically the testing algorithms can be divided into two types: Traditional test algorithms and March based test algorithms. Traditional test are checkerboard, GALPAT, Walking 1/0, butterfly and etc. March algorithms are preferred over Traditional testing algorithms because March algorithms are highly linear, simple and good fault coverage.

In this paper we discuss about state diagram representation of March SS algorithm for Finite State Machine (FSM) Based Memory BIST architecture. In the state diagram representation of March SS algorithms having finite sequences of write and read operations. This paper is organized as follows. Section II describes various Memory faults. Section III describes proposed FSM based memory BIST architecture for March SS algorithm and FSM implementation of March SS algorithm. Section IV describes simulation results. Section V describes the conclusion of this paper.

II. MEMORY FAULTS

Basically functional memory faults can be divided into several types some of them are shown below.

1. Stuck-at Fault: The Stuck-at Faults occurs due to Permanently Stuck of cell or line to a value of 0 or 1. The Stuck at fault is divided into two types, Stuck-at 0 and Stuck-at 1. For Stuck-at 0, a line or cell is permanently Stuck-at 0 and For Stuck-at 1, a line or cell is permanently Stuck-at 1.

2. Coupling Fault: If the content of a cell is changed from 0 to 1 or 1 to 0 then the contents of other cell gets changed then this type of faults called coupling Faults. Coupling faults are divided into three types which are Idempotent Coupling Fault, Inverse Coupling Fault and State Coupling Fault.

3. Transition Fault: The Transition fault is classified into two types, Up-Transition fault and Down-Transition fault. If 0 to 1 transition is not possible into a cell then it is called Up-Transition Fault and If 1 to 0 transition is not possible in a cell then it is called Down-Transition Fault.

4. Address Decoder Fault: The address decoder faults can be occurred due to four reasons. These are; certain cell cannot be accessed by any address, certain cell accessed by multiple addresses, no cell accessed by certain address and multiple cells accessed by certain address.

5. Data Retention Fault: If a memory cell becomes unable to retain its logical value after some time, then this type of fault is known as data retention fault. This type of faults is caused by a broken pull-up resistor.

6. Neighborhood Pattern Sensitive Fault: The Neighborhood Pattern Sensitive Fault is a special case of coupling fault. In NPSF fault the contents of memory cell changed due presence of certain bit pattern in to the neighborhood cell. The NPSF can be classified in to three types, which are Static

NPSF, Passive NPSF and Active NPSF.

III. PROPOSED BIST ARCHITECTURE & FSM REPRESENTATION OF MARCH SS ALGORITHM

The March algorithm is having a finite sequence of March element. Each March element is having a finite sequence of read and write operations which are applied to the memory cell before processing to the next cell. The address of next cell can be in ascending or descending address order.

The proposed block diagram of Memory BIST Architecture Based on FSM is shown in figure 1. This Architecture consists of FSM block of March SS algorithm, Data generator, Address generator and Comparator block. In FSM based memory BIST the BIST controller is implemented through FSM.

The Address Generator block produces addresses in up or down addressing order which depends upon signals receiving from FSM block of March SS algorithm. The Data Generator block produces data value 0 or 1 for producing 1 byte or 0 byte. The Comparator block compares the output of memory with the expected data. If these are not matched it produces faultdetect signal to high. If these are matched then faultdetect signal will be at logical level 0 (Low). After completion of performing all March element of March algorithm the BIST controller block produces bist_end signal to high.

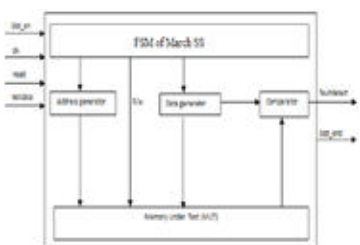

Figure 1: Block Diagram of Memory BIST Architecture.

In the proposed FSM based Memory BIST, bist_en, clk, reset, faultdetect and bist_end are basic signals. If bist_en signal goes to high, then proposed BIST is come into the work mode. The clk signal implies the clock pulse applied to the circuit. The faultdetect signal indicates the detection of fault. The bist_end signal indicates completion of March algorithm. If bist_en signal set to 1, reset signal set to 0, clk signal is applied and March algorithm is selected then write or read operations are performed onto memory locations in either ascending or descending address order, depends upon March algorithm sequence. The output of memory is compared with the expected data using comparator block, if it is not matched then it produces faultdetect signal to high and else it produces faultdetect signal to low. After performing all operations of March algorithm the BIST circuit produces the bist_end signal to high.

In State diagram of March SS algorithm first state denotes idle state, intermediate states (S0, S1, S2, S4, S5) denotes March elements of March SS algorithm and last state (S6) is used for bist_end state. The March SS algorithm is shown in below table I. State Diagram Representation of March SS Algorithm is shown in figure 2.

The Stuck-at 1 fault is detected by writing 0 (W0) to memory location and then reading expected value 0 (R0); if it is not matched then it is known as Stuck-at 1 fault. The Stuck-at 0 fault is detected by writing 1 (W1) to the memory location and reading expected value 1 (R1); if it is not matched then it is known as Stuck-at 0 fault.

TABLE I: MARCH SS ALGORITHM.

MARCH SS	{⇕(w0); ↑(r0,r0,w0,r0,w1); ↑(r1,r1,w1,r1,w0); ↓(r0,r0,w0,r0,w1); ↓(r1,r1,w1,r1,w0); ⇕(r0)}
(22n)	{S0,S1,S2,S3,S4,S5}

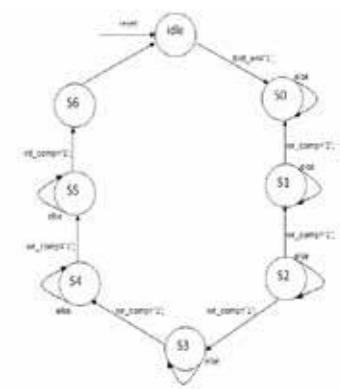

Figure 2: State Diagram Representation of March SS Algorithm.

V. SIMULATION RESULTS

The Mach SS algorithm is implemented using Very High Speed Integrated Circuit Hardware Description Language using Xilinx ISE 8.2i Tool. The Corresponding PIN Diagram for March SS Algorithm is shown in Figure 3. The RTL view of March SS algorithm is shown in Figure 4. The Simulation Waveform for the March SS algorithms is shown in Figure 5. This Simulation Result is carried out using Modelsim SE 6.3f Tool.

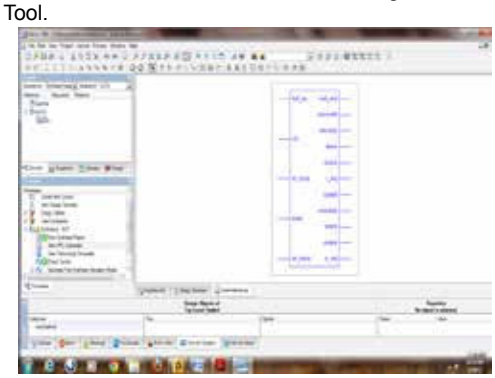

Figure 3: PIN diagram representation of March SS Algorithm.

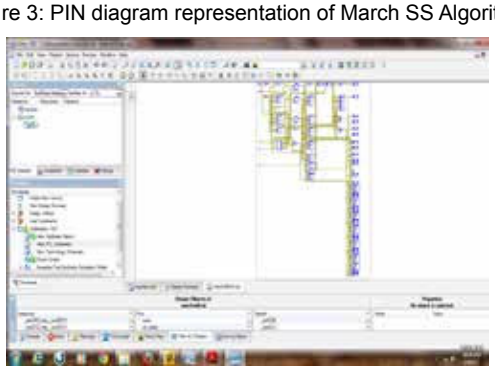

Figure 4: RTL View of March SS algorithm.

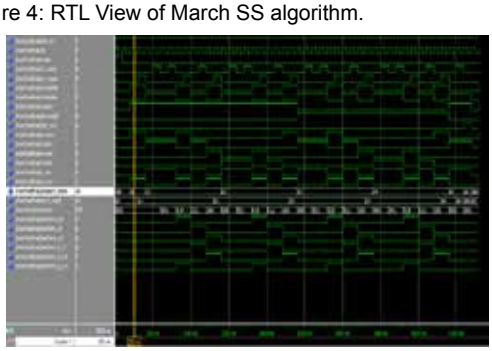

Figure 5: The Simulation result of March SS algorithm.

VI. CONCLUSION

In System-on-Chip (SoC) environment Embedded memory testing can be efficiently done by using March algorithms. Now-a-days several March Algorithms are widely used. In this paper we implement an efficient representation of March SS algorithm based on finite state machine representation. The March SS algorithm is used to detect all realistic simple static faults in RAMs.

The FSM Based BIST consists of several advantages such as speed testing, small area and compact than other Memory BIST approaches.

We can use more than one March algorithm for achieving better fault coverage. We can also implement other March algorithms such as March RAW, March DSS, March AB, March BDN etc. using for better fault coverage.

REFERENCES

[1] Po-Chang Tsai, Sying-Jyan Wang and Feng-Ming Chang; "FSM-Based Programmable Memory BIST with Macro Command," in Proceedings of The 2005 IEEE International Workshop on Memory Technology, Design and Testing, Aug. 2005, pp. 72-77. | [2] Nor Zaidi Haron, Siti Aisah Mat Junos, Abdul Hadi Abdul Razak and Mohd. Yamani Idna Idris; "Modeling and Simulation of Finite State Machine Memory Built-in Self Test Architecture for Embedded Memories", in Applied Electromagnetics, 2007. APACE 2007. Asia-Pacific Conference on 4-6 Dec. 2007, pp.1 – 5. | [3] Nur Qamarina Mohd Noor, Azilah Saparon, Yusrina Yusof and Mahmud Adnan; "Area-Effective Programmable FSM-Based MBIST for Synchronous SRAM", in proceedings of 1st Asia Symposium on Quality Electronic Design, 2009 (ASQED 2009), 15-16 July 2009, pp. 203 – 207. | [4] K . Zarrineh and S.J. Upadhyaya; "On programmable memory built-in self test architectures" in Proceedings of Design, Automation and Test in Europe Conference and Exhibition 1999, 1999, pp. 708 – 713. | [5] Subir K. Roy, Rubin A. Parekhji; "Modeling Techniques for Formal Verification of BIST Controllers and Their Integration in SOC Designs", VLSI Design, 2007, Held jointly with 6th International Conference on Embedded Systems., 20th International Conference on, Jan. 2007, pp. 364 – 372. | [6] Qiao Liyan, Bai Shi, Zhao Xin and Wang Ge; "Design of Generic Embedded Memory Built in Self Test Circuit", in proceedings of the ninth international conference on electronic measurement & instruments (ICEMI'2009), 16-19 Aug. 2009, 2-244 - 2-247. | [7] A. J. van de Goor and Z. Al-Ars; "Functional Fault Models: A Formal Notation and Taxonomy", in Proceedings of IEEE VLSI Test Symposium, 2000, pp. 281-289. | [8] S. Hamdioui, A.J. van de Goor and M. Rodgers, "March SS: A Test for All Static Simple RAM Faults", In Proc. of IEEE International Workshop on Memory Technology, Design, and Testing, pp. 95-100, Bendor, France, 2002. | [9] http://www.xilinx.com.