Quest Journals
Journal of Electronics and Communication Engineering Research
ISSN:2321-5941 Volume1 ~ Issue 2 (2013) pp: 01-06
www.questjournals.org

Research Paper

A VLSI Implementation of High Speed FSM-based programmable Memory BIST Controller

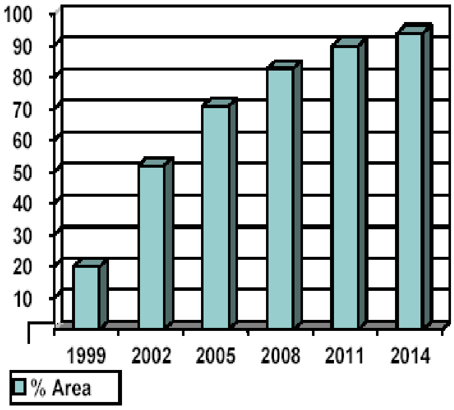
Manikandan.B[#1], Praveen kumar.J[#2]
M.E. Department of VLSI Design (ECE) SVCET (Thiruvallur)[#1],ARMCET(Maraimalai Nagar)[#2]

Received 21 September, 2013; **A**ccepted 30 October, 2013© The author(s) 2013. Published with open access at **www.questjournals.org**

ABSTRACT: *All system chips contain some type of embedded memories, such as RAM, ROM and Flash memory. This memory is likely to be affected by manufacturing faults. A FSM-based programmable memory built in self test (MBIST) Controller used to testing the memory devices[3]. The MBIST controller is designed to implement a new test algorithm. This algorithm is known as March based test algorithm. March based test algorithms are MATS+, March c-, March A and etc[7]. This algorithms are less number of operations and testing of high density memory ICs either faulty or good. The controller and test algorithm are studied and designed using verilog HDL and implemented in SPARTAN-3E FPGA. In this paper analysis of the timing, logic area usage and speed are presented.*

I. INTRODUCTION

Today semiconductor memories are an integral part of modern VLSI circuits. The memory share of the chip area increases and is expected to be 94% in 2014. Advances in the memory technology make memory testing more and more complicated due to appearance of the new defect mechanisms in memory devices. The memory testing will become a major cost factor in the production of the modern VLSI.



Chip Area Occupied by the memories

The Built- in self-test (BIST) has been proven to be one of the most cost-effective and widely used solutions for memory testing for the following reasons.

- Tests can run at circuit speed to yield a more realistic test time.
- No external test equipment
- Reduced development efforts.
- On-chip Test Pattern generation to provide higher controllability and Observability.
- On-chip response analysis.
- Test can be on- line or off-line.
- Adaptability to engineering changes.
- Easier burn-in support.

FSM based memory BIST controller type of architecture has optimum area overhead however less flexible to allow any changes in the test algorithm. Two types of FSM-based BIST controller architectures. First one is using MISR which is a part of the BIST controller block for output response analyzer (ORA) while another one is using comparator which acts as an external block in the BIST system for ORA. Both are designed by using a test pattern generator and test controller. The programmable FSM is low level controller which is Pre-programmed with instructions for read/write operation and addressing sequences for set of MARCH test algorithms. The latest MBIST design combined both microcode-based and FSM based architecture to compensate the area versus speed issue.

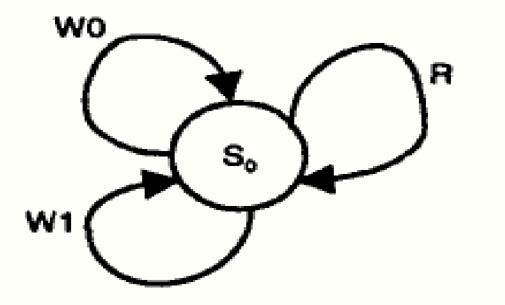
The efficient testing algorithms have been proposed to detect different fault models. The testing algorithms can be divided into two types. They are Traditional test algorithms and March based test algorithms. Traditional test are checkerboard, GALPAT, Walking 1/0, butterfly and etc [9]. In this test algorithm is old test algorithm. Traditional test algorithm overcome by March based test algorithm. Because March algorithms are highly linear, simple and good fault coverage.

In this paper, MATS+ as the test algorithm and the comparison is made on the logic usage and speed by implementing the architecture on SPARTAN-3 FPGA. The paper is organized as follows; section II introduces for manufacturing faults; section III introduces the proposed High Speed Programmable Memory Built-In Self Test controller (HP-MBIST); section IV introduces for MARCH test algorithm. The experimental results are discussed in Section V while Section VI concludes the paper.

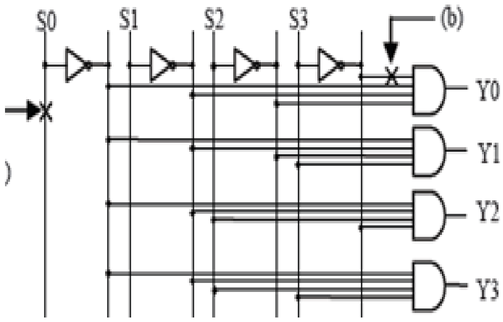
II. MEMORY FAULTS

The memory is intensively accessed through read and writes operations, it is more likely to be affected by manufacturing faults[6]. The following is a brief description about the main types of these faults.

A. Stuck-at Faults: This is the most common type of faults in memory. The memory cell itself may be stuck at 0 or stuck at 1.



B. Address Decoder Fault (AF): This type of faults can be classified under the stuck-at faults since one of the nodes in the address decoder may stuck at 0 or 1 leading to accessing wrong address, no address, or multiple addresses.



C. Transition Fault (TF): This fault causes the memory cell state to go into one direction only. For example, if the cell contains 0 and 1 has been written to it, then it cannot be written back to 0 and vice-versa.

D. Coupling Fault (CF): In this fault, transition in one cell may affect its neighboring cell and cause it to go into erroneous state. The cell that causes the coupling fault in its neighbor is called the aggressor whereas the affected cell is called the victim cell**.**

III. FSM-BASED HP-MBIST CONTROLLER

A FSM-based controller as shown in Fig. It is a Hardware realization of a selected memory test algorithm, usually in the form of a Finite State Machine (FSM).This FSM-based HP-memory BIST architecture has optimum logic overhead and short test time.

The FSM Based MBIST controller consists of three blocks. They are

- Control unit
- AM Unit
- MUT and Comparator

A. Control Unit

The main function of the control unit is to generate the addressing order and march elements address are supplied as inputs to the access unit. After power on, when reset (active low) signal is applied, all the registers in the control unit automatically selects the March A algorithm, which is in the form of an FSM [8]. By executing the FSM control unit generates the March element address. It consists read/write operations and addressing order. Then it sends a signal to the access unit. To start the execution of march element with a particular addressing order. When the execution of element is completed, access unit sends a signal back, called end march, which is the indication of that a particular march element is completed. The control unit sends the next march element address and addressing order to access unit. When all the march elements are completed control unit asserts the end of test signal. Therefore test is completed.

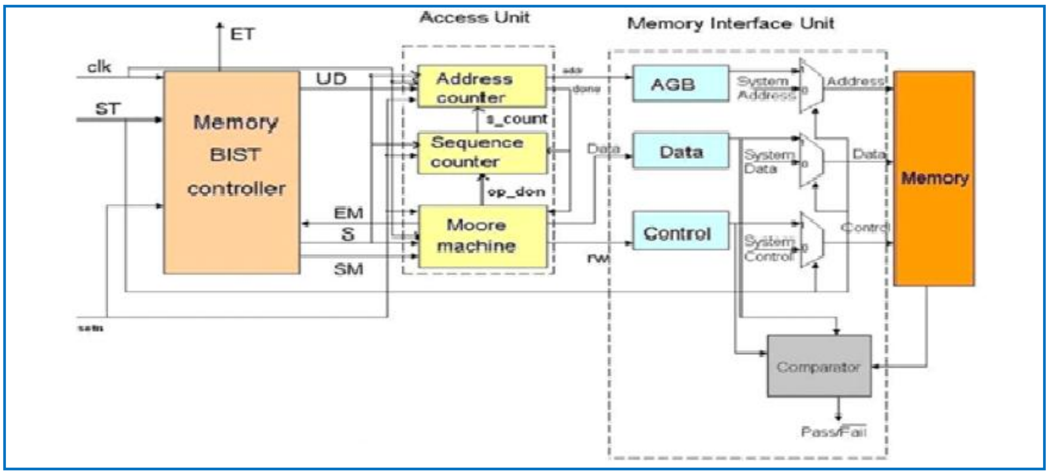
The Operation of control unit explained step by step initially the controller is in idle state. It changes the state only when the start_test =1.

□ □ If start_test =1, the controller is initiated to test mode. Then it sends the March element to access unit by asserting start_march=1. The end_march value will be zero when the access unit executing March element.

□ □ When end_march =1, then the state is changed from M0 to M1 and increments the s_count by 1 i.e. the current March element is completed and waiting for the next March element.

□ □March U algorithm has 4 March elements and 6 transitions. The transition is taken place according to the element counter.

□ □ After completion of all the March elements, the BIST controller enters into idle mode.



FSM-Based HP-MBIST Controller Block Diagram

B. Access Unit

Access unit consists of three individual blocks called Address counter, Sequence counter and Moore machine.

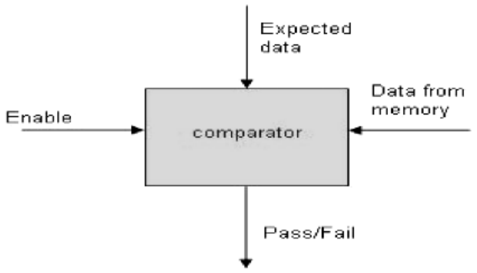
- Address counter it takes the addressing order from the control unit and generates the address of MUT (memory under test). If addressing order is ‘1’, the address is incremented from ‘0’ to maximum address of MUT. If it is ‘0’ address is decremented from maximum address to ‘0’. When it reaches the maximum or minimum address it assets a done signal, which means all the operations in all memory locations in a MUT are completed.
- Sequence counter value is incremented by 1 When a march operation is completed in moore machine, For march elements ‘0’ and ‘3’ the maximum sequence count value is 0 and for elements ‘1’ and ‘2’

the maximum sequence count value is 1. When the march operation is completed, the incremented sequence count value is supplied to address counter

- In the Moore Machine the March element operations are realized by an FSM. Depending upon the march operation moore machine generates the address and control signals. When a particular march operation is completed in the Moore machine asserts the signal called op_done and it is supplied to sequence counter.

C. Memory Interfacing Unit

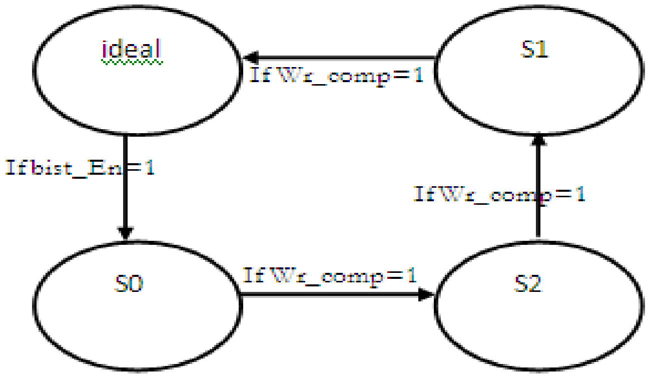
- Memory Interface Unit will take the address, data and rw (read/write) signals from the access unit and stored in the Address, Data and Control registers. On start_test value, respective signals are selected. If start_test =1 then test data and control will be considered otherwise memory is used for normal functions.
- Address Generation Block stores the address of memory on which the operations to be performed. This address will be used for the diagnosis purpose if the fault is occurred in the respective location · Data register stores the data to be written onto the memory location.
- Control Register consists of the memory operation to be performed on the memory location indicated by Address Generation Block.
- Comparator is used to compare the expected data and data from the memory and tells the memory location is pass or fail. Comparison is done when the enable signal is‘0’.The enable signal is nothing but control signal.



2 Bit Comparator

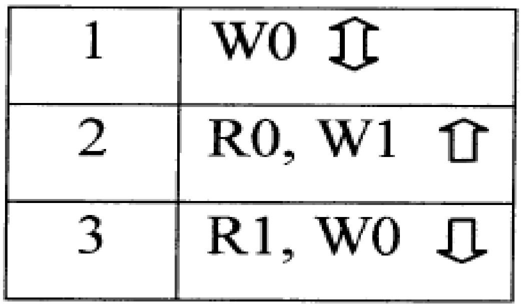
IV. MARCH TEST ALGORITHM

March algorithms are highly linear, simple and good fault coverage. March algorithms are MATS+, March A, March B, March C-, March U and etc. A March element is a finite sequence of read (r) or writes (w) operations applied to a cell in memory before processing the next cell [1]. The address of the next cell can be in either ascending or descending address order.



MATS+ algorithm Architecture

For test a given memory cell is good, it required to conduct a sequence of write and read operations to the cell. The actual number of read/write operations and the order of the operations depend on the target fault model.



1	W0 ⇕
2	R0, W1 ⇑
3	R1, W0 ⇓

MATS+ Pattern

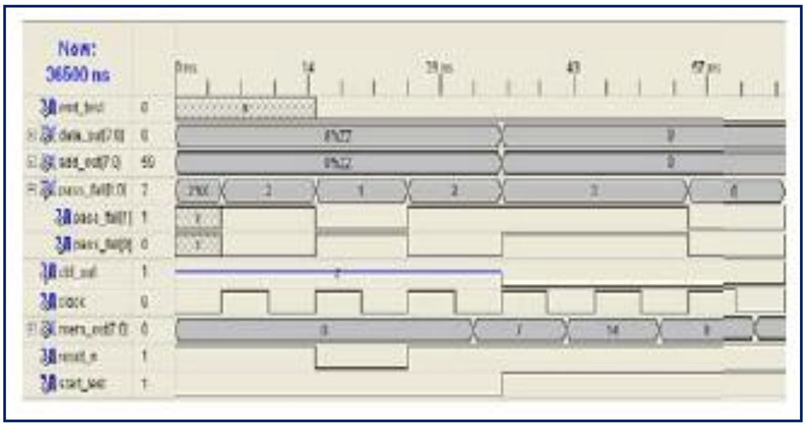
The MATS+ algorithm first writes a 0 to each cell in any order ((w0)). Here in simulation we are showing the Faulty FSM BIST in which test input of 8 bit has been given to FSM BIST also “001” has been shown by “sel” signal to the MATS+ in which 1 test vector “0” needs to be written and “0” all eight location is written for 64 counts after which algorithm will be changed to next state.

V. RESULTS

MATS+ is the selected test algorithm of the FSM-based controller for simulation. The simulated waveform as shown in the Fig 7 gives the pass/fail information of a memory which is under test. The information is given by the comparator which is a part of memory interface unit. The two bit comparator compares the memory output with the expected data. Comparator is enabled only during the read operation and posedge clock. For the first time the comparator output is undefined that is high impedance because negedge clock. When comparator is enabled and memory output is equals to expected data, the output is ‘11’(3) otherwise its output is ‘00’(0). When it is disabled its output is ‘10’(2). When all the locations in MUT (memory under test) have been tested BIST controller asserts the end_test signal which is the indication of end of test.

Test cases	Operation	Pass/fail	Results
1	Reset”0” Start_test”0” Ctrl_out”xx”	1(2”b01)	Pass_fail 2’b01 reset all registers address counter and sequence counter mean resetting the block
2	Reset”1” Start_test”1” Ctrl_out”0”	3(2’b11) Fail	pass_fail 2’b00 gives mem_data is not equal to test_data
3	Reset”1” Start_test”1” Ctrl_out”0”	0(2’b00) Pass	pass_fail 2’b11 gives mem_data is equal to test_data

Test cases of Top Module



Simulation of FSM-based HP-MBIST controller

Minimum period 4.068ns	4.068ns
Maximum Frequency	**246.15MHz**
Maximum output required time after clock	4.191ns
Maximum combinational path delay	4.979ns

Timing Summary

FSM-Based controller	System frequency(MHz)
PMBIST(MARCH SAM) [10]	80.27
Hybrid PMBIST[11]	82.17
Our Design(HPMBIST)	**246.15**

Speed Comparison

The proposed design is synthesized using Xilinx 9.2 ISE Synthesis Tool order to acquire the area usage and speed. The Synthesized Summary Report of the FSM-based HP-MBIST controller gives a RTL Schematic, Timing Summary and Area Vs Speed comparison with the FSM-based PMBIST controller using MATS+ [10] and Hybrid P-MBIST [11] controllers.

FSM-Based controller	Area(No of Instances used ondevice)
PMBIST(MARCH SAM) [10]	77(LE)
Hybrid PMBIST[11]	81(LE)
Our Design(HPMBIST)	**75(LE)**

Area overhead Comparison

VI. CONCLUSION

In this paper MATS+ algorithm is implemented to the FSM based HP-MBIST. The simulation portrays that the tested data and the expected data are able to be compared in the architecture. Hence it is concluded that this controller has the ability to detect faulty or good memory ICs. Synthesis result shows that the FSM-based HP-MBIST controller employs only 75 instances with clock frequency 246.15 MHz our design gives less usage of Logic Elements (LE) with High speed testing of memories as shown in above Tables. It is justified that the FSM-based HP-MBIST controller consumes less area overhead and high speed while the other design [9][10] consumes more area overhead and less speed the experimental results also shows that the proposed BIST can be implemented with low area overhead.

REFERENCES

[1]. K.Zarrineh, and S.J. Upadhyaya, “On programmable memory built in-self test architecture” in Proc. IEEE Design, Automation and Test in Europe Conference, 1999.

[2]. R. Dean Adam, High Performance Memory Testing, Chapter 11: State Machine BIST, Kluwer Academic Publisher, 2003.

[3]. Po-Chang Tsai, Sying-Jyan Wang and Feng-Ming Chang, “FSM Based Programmable Memory BIST with Macro Command” in IEEE International Workshop on Memory Technology, Design, and Testing, 2005.

[4]. Charles E.Stroud, A Designer’s Guide to Built-In Self Test, Chapter 12: BIST for Regular Structure, Kluwer Academic Publisher, 2002.

[5]. Chung-Fu Lin and Yeong-Jar Chang , “An area-efficient design for Programmable memory Built-In Self- Test” in IEEE International Symposium VLSI Design, Automation and Test, 2008.

[6]. K.Zarrineh, and S.J. Upadhyaya, “On programmable memory built in-self test architecture” in Proc. IEEE Design, Automation and Test Conference , 1999.

[7]. Paul Mc Evoy and Ronan Farrell, “Built-In Self Test Engine For Memory Test ” , in IEEE IC Test Workshop, 2004.

[8]. A J van de Goor and I.B.S Tlili, “March Tests for Word Oriented Memories”, in Proc.Design Automation Test Europe, 1998.

[9]. Said Hamdioui and John Eleazar Q. Delos Reyes, “New Data Background Sequences and Their Industrial Evaluation for Word-Oriented Random Access Memories”, in IEEE Transaction on Computer Aided Design of Integrated Circuits and System, 2005.

[10]. Nur Qamarina Mohd Noor, Azilah Saparon, Yusrina Yusof, “An Overview Of Microcode-based and FSM based Programmable Memory Built-In Self Test(MBIST) Controller for Coupling Fault Detection”, in IEEE Symposium on Industrial Electronics and Applications , 2009.

[11]. NurQamarina Mohd No or, Azil ah Saparon, Yusrina Yusof, “Programmable MBIST Merging FSM and Microcode Techniques Using Macro Commands”, in 25th International Symposium on Defect and Fault Tolerance in VLSI Systems,2010.