Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X

IJCSMC, Vol. 2, Issue. 2, February 2013, pg.46 – 52

RESEARCH ARTICLE

High Speed FSM-based programmable Memory Built-In Self-Test (MBIST) Controller

Dr. K Padma Priya
Associate Professor, Faculty of Electronics and Communication Engineering, JNTUK College of Engineering and Technology, Vijayanagaram, India
kesaripadmapriya@yahoo.com

Abstract— This paper proposed a High speed FSM-based controller for programmable memory built-in self-test for testing memory devices. This technique is popular because of its flexibility of new test algorithms. The architecture of controller is designed to implement a new test algorithm has less number of operations and this algorithm emphasis testing of high density memory ICs either faulty or good .The components of controller is studied and designed using Verilog HDL. The analysis of the timing, logic area usage and speed are presented.

Indexed Terms: - Moore FSM controller, Built in Self-Test, Memory, at speed, DFT.

I. INTRODUCTION

Memories are the most universal component today almost all system chips contain some type of embedded memory, such as ROM, SRAM, DRAM, and flash memory. In the embedded domain, embedded RAMs of the StrongArmSA110 occupy 90% of the total area. The projection is, by 2014, memory will represent more than 94% of the chip area in average SOC environment, according to the International Technology Roadmap for Semiconductors 2007[7] the percentage of chip area occupied by memories in a design and the increasing trend predicted for the next decade, with the advent of deep-submicron VLSI technology, the memory density and capacity is growing but the clock frequency is never higher. The dominant use of embedded memory cores along with emerging new architectures and technologies make providing a low area overhead and high speed test solution for these on-chip memories a very challenging task.

Built- in self-test (BIST) [5] has been proven to be one of the most cost-effective and widely used solutions for memory testing because the tests can run at circuit speed to yield a more realistic test time, no external test equipment, reduced development efforts and on-chip test pattern generation to provide higher controllability and observability.

There are several FSM-based controllers proposed in [1-10].In FSM-based memory BIST controller, counters are the key component especially in FSM-based memory BIST controller but some FSM-based BIST controller [2] excluded counter from its design. This type of architecture has optimum area overhead however less flexible to allow any changes in the test algorithm. Usually, different counters [3], [4] are used to generate the address, test data and read/write sequences. Two types of FSM-based BIST controller architectures are proposed in [5]. Both are designed by using a counter for the test pattern generator and test controller but 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.

The FSM-based memory BIST has also progressed from non-programmable memory BIST to programmable memory BIST. One of the early FSM-based P-MBISTs has two controllers; upper controller and lower controller [6]. A two-dimensional circular buffer acts as the upper level controller which holds the necessary parameters for the low level controller. The low level controller is the programmable FSM which is pre-

programmed with instructions for read/write operation and addressing sequences for set of MARCH test algorithms. This type of architecture has optimum area overhead however still less flexible to allow selection of the test algorithms to be run on the memory cores.

Some of the latest MBIST [10] design combined both microcode-based and FSM based architecture to compensate the area versus speed issue. A programmable MBIST merging FSM and Microcode Techniques [11] using Macro Commands is designed by implementing clusters of microcode to control the read/write operation and test data injection. This technique results in optimal lower area overhead compensate area and speed using number of test algorithms.

The paper is organized as follows; section II introduces the proposed High Speed Programmable Memory Built-In Self-Test controller (HP-MBIST) for MARCH C+. The experimental results are discussed in Section III while Section IV concludes the paper.

II. MBIST CONTROLLER FOR MARCH C+

A. March C+ test algorithm

A test algorithm is a test procedure which uses a finite sequence of test elements for testing the memory and identifying and locating defects [8].A test element contains a number of memory operations (access commands), data pattern (background) specified for the read operation, address (sequence) specified for the read and write operations. The test pattern used in the MARCH C+ Algorithm, It has less number of operations (14n) where 14n is number of operation per memory word, due to less no of operations it requires less time to test the memory under test (MUT) and Addressing order and their respective operation shown in Table 1 and fig 1 shown read-write operation in memory cells in ascending and descending addressing order respectively.

S No	Operation
1	↑ (w0)
2	↑ (r0,w1,r1)
3	↑ (r1,w0,r0)
4	↓ (r0,w1,r1);
5	↓ (r1,w0,r0);
6	↓ (r0);

Table1: Description of Test Pattern Used

.

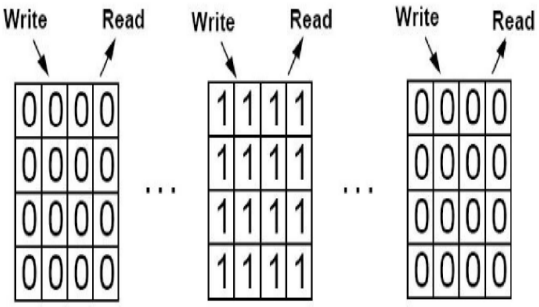


Fig 1: March C+ algorithm operation

B. FSM-based HP-MBIST controller

A FSM-based controller [2] as shown in Fig 2 is a hardware realization of a selected memory test algorithm, usually in the form of a Finite State Machine (FSM).This type of memory BIST architecture has optimum logic overhead and short test time.

The block diagram the FSM-Based MBIST controller consists of three blocks. They are

- Control unit
- Access Unit
- MUT and Comparator

a. Control Unit

The control unit is one of the blocks in FSM-Based controller. The main function of the control unit is to generate the addressing order and march element address, which 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 C+ algorithm, which is in the form of an FSM. By executing the FSM, control unit generates the March element address, which consists Read/write operations and addressing order. Then it sends a signal to the access unit, called start march, which is the indication of 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. Then 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, which means the test is completed.

The control unit is implemented by realizing an FSM. It executes the March C+ algorithm. The algorithm is shown below.

March C+: ↑ (w0); ↑ (r0,w1,r1); ↑ (r1,w0,r0); ↓ (r0,w1,r1); ↓ (r1,w0,r0); ↓ (r0);

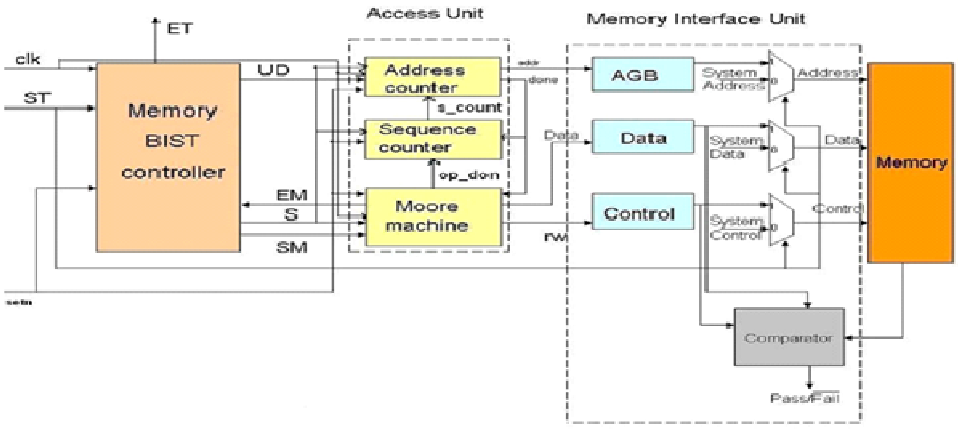


Fig 2: FSM-Based HP-MBIST Controller Block Diagram

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 C+ 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.

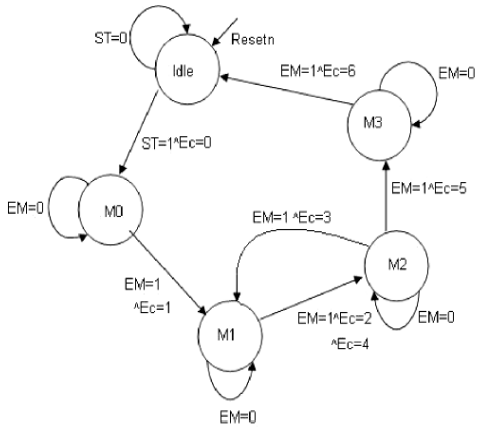


Fig 3: Realization of March C+ algorithm by an FSM

b. Access and Memory Interfacing 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.
- 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.

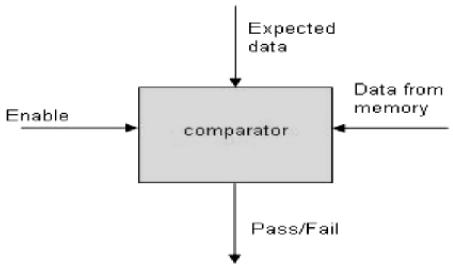


Fig 4: 2 Bit Comparator

III. EXPERIMENTAL RESULTS

MARCH C+ 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.

A. Simulation summary report

Test cases	Description	Pass/fail	Comments
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

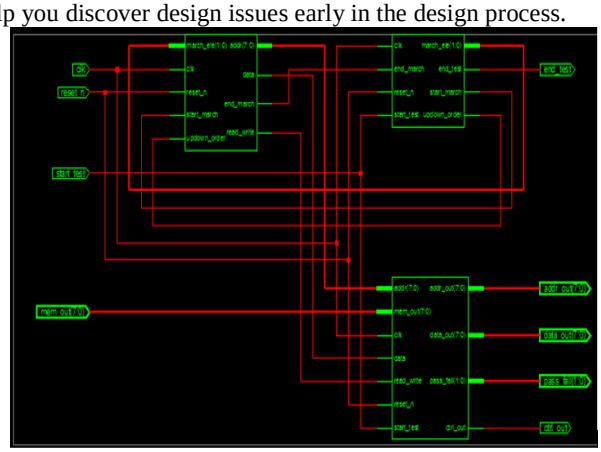
Table 2: Test cases of Top Module

B. Synthesis summary report

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 MARCH-SAM [10] and Hybrid P-MBIST [11] controllers.

C. RTL schematic

The synthesized design can be viewed as a schematic in the register transfer level (RTL) viewer. The RTL Schematic shows a representation of the pre-optimized design in terms of generic symbols such as AND gates and OR gates adders, multipliers, counter, that are independently of the targeted Xilinx device. Viewing this schematic allows you to see a technology-level representation of your HDL optimized for a specific Xilinx architecture, which may help you discover design issues early in the design process.



Figurer 5: RTL schematic of top module

D. Timing summary

Minimum period	4.068ns
Maximum Frequency	**245.821MHz**
Minimum input arrival time before clock	4.112ns
Maximum output required time after clock	4.191ns
Maximum combinational path delay	4.979ns

Table 3: Timing Summary

E. Speed Comparison

FSM-Based controller	**System frequency(MHz)**
PMBIST(MARCH SAM) [10]	80.27
Hybrid P-MBIST[11]	82.17
Our Design(HP-MBIST)	**245.82**

Table 4: Speed Comparison

G. Area overhead Comparison

FSM-Based controller	**Area(No of Instances used on device)**
PMBIST(MARCH SAM) [10]	77(LE)
Hybrid P-MBIST[11]	81(LE)
Our Design(HP-MBIST)	**76(LE)**

Table 5: Area overhead Comparison

IV. CONCLUSION

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 76 instances with clock frequency 245.82 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 eexperimental 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, "*FSMBased 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 McEvoy 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 FSMbasedProgrammable Memory Built-In Self Test(MBIST) Controller for Coupling Fault Detection",in* IEEE Symposium on Industrial Electronics and Applications , 2009.

[11] NurQamarina Mohd Noor, Azilah 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.

