International Journal of Psychosocial Rehabilitation, Vol. 24, Issue 10, 2020
ISSN: 1475-7192

Coupling Faults Detection In Memories Using With Finite State Machine And Microcode Based And Microcode Based Memory Built In Self Test(Mbist)

[1]P Anitha, [2]CH Moses, [3]R Purshotham Naik ,[4] P John Paul

Abstract--- *Built-in Self-Test, or BIST, is the technique of designing additional hardware and software features into integrated circuits to allow them to perform self-testing, testing of their own operation using their own circuits, thereby reducing dependence on external automated test equipment (ATE). BIST is also the solution to the testing of critical circuits that have no direct connections to external pins, such as embedded memories used internally by the devices. In the near future, even the most advanced tester may no longer be adequate for the fastest chip, a situation wherein self-testing may be the best solution. Microcode-based and FSM-based controllers are two widely known architectures used for programmable memory built-in self-test. These techniques are popular because of their flexibility of programming new test algorithms. In this paper, the architectures for both controllers are designed to implement a new test algorithm MARCH SAM that gives a better fault coverage in detecting single-cell fault and all intra-word coupling fault (CF).The components of each controllers are studied and designed. The both of the controller are written using Verilog HDL and implemented FPGA. The simulation and synthesis results of both architectures are presented.*

Keywords--- *Built in self test , FSM, Coupling faults, march Sam, fault, FPGA.*

I INTRODUCTION

Now a days techniques are improved to detecting the errors towards to Static Random Access Memory. many effective algorithms are integrated in Memory Built In Self-Test Architecture[1]. The MBIST is technique played a major role as embedded memories for System On Chip[2]. The constellation of this component is tremendous and crowded due to the large number of data to be stored [3]. However the complexity because of the embedded memories are making challenge among the problems are encountered while testing the memories. Memory BIST has been used successfully for years to solve the test issues of embedded memories. In addition to testing embedded memories using expensive external memory tester, BIST is considered a good alternative solution[4]. MBIST Architecture has is proposed for memory testing in itself. But the thing is limited data patterns are generated and memory accessing schemes also. Large number of programmable architectures has been proposed to solve the problems[5]

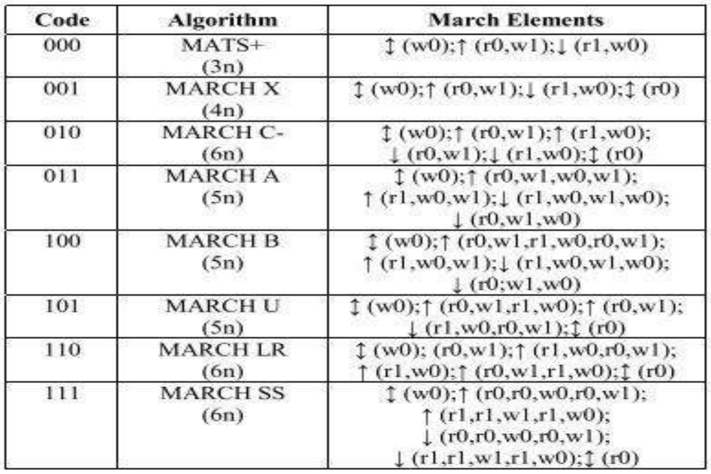
1,2, Assistant Professor, Department of Electronics and communication Engineering
3,4, Professor, Department of Electronics and communication Engineering
Malla Reddy College of Engineering, Maisammaguda, Hyderabad, Telangana,India - 500100
1, anithakrishna77p@gmail.com, 2, moses.chellaece@gmail.com, 3, rpnaik.naik@gmail.com, 4, jppulipati@yahoo.com

Received: 27 Feb 2019 | Revised: 20 Mar 2019 | Accepted: 30 Apr 2020

II LITERATURE SURVEY

Facing achieve optimal System on Chip (SOC),the mechanism must be implemented to test the embedded memories[3]. To achieve optimal SOC yield, an at-speed testing mechanism must be implemented to test these embedded memories [6]. Consequently the only practical solution is derivable from Built In Self-Test. BIST is having two patterns those are 1. Pseudorandom Pattern 2. Deterministic pattern Pseudorandom pattern usually preowned to test with logical test sequences and the other one deterministic pattern is used to test memories with MARCH Algorithm. Hence the only practical solution available is by employing built-in-self-test (BIST). BISTs have either two patterns; pseudorandom pattern or deterministic pattern. Pseudorandom pattern is usually used to test logic circuits while the deterministic pattern is applied to test memories. MARCH. The below table 01. Is showing binary code corresponding with algorithms and March Elements.

TABLE 1: TEST PATTERNS OF BUILT IN SELF TEST



Code	Algorithm	March Elements
000	MATS+ (3n)	↕ (w0);↑ (r0,w1);↓ (r1,w0)
001	MARCH X (4n)	↕ (w0);↑ (r0,w1);↓ (r1,w0);↕ (r0)
010	MARCH C- (6n)	↕ (w0);↑ (r0,w1);↑ (r1,w0); ↓ (r0,w1);↓ (r1,w0);↕ (r0)

011	MARCH A
	(5n)
		↕ (w0);↑ (r0,w1,w0,w1);
		↑ (r1,w0,w1);↓ (r1,w0,w1,w0);
		↓ (r0,w1,w0)

100	MARCH B
	(5n)
		↕ (w0);↑ (r0,w1,r1,w0,r0,w1);
		↑ (r1,w0,w1);↓ (r1,w0,w1,w0);
		↓ (r0;w1,w0)

101	MARCH U (5n)	↕ (w0);↑ (r0,w1,r1,w0);↑ (r0,w1); ↓ (r1,w0,r0,w1);↕ (r0)
110	MARCH LR (6n)	↕ (w0); (r0,w1);↑ (r1,w0,r0,w1); ↑ (r1,w0);↑ (r0,w1,r1,w0);↕ (r0)

111	MARCH SS
	(6n)
		↕ (w0);↑ (r0,r0,w0,r0,w1);
		↑ (r1,r1,w1,r1,w0);
		↓ (r0,r0,w0,r0,w1);
		↓ (r1,r1,w1,r1,w0);↕ (r0)

The familiar technique is Pseudorandom Built In Self-Test (PBIST) are generated the test sequences widely for integrated circuits and systems[7]. The Test Pattern Generators The Pseudorandom generators includes, linear feedback shift registers (LFSRs) [2], cellular automata [5], and accumulators driven by constant value [3].

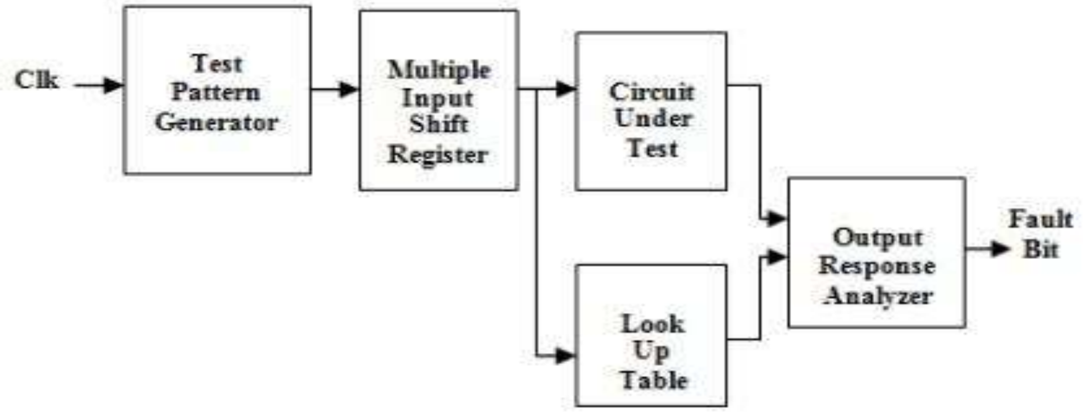


Figure.1 Block Diagram of Test Pattern Generator (TPG)

In line-up to recognize the faults in any circuits or devices, the large number of random patterns are to be generate before high fault coverage is rack up[8]. Therefore, weighted pseudorandom techniques have been suggested. Weighted random pattern generation methods relying upon an individual weight assigning generally fail to achieve complete fault coverage applying a reasonable number of test patterns since, although the weights are computed to comprise for most faults, several faults can require long test sequences to be perceived on this weight assignment whenever they do not match their activation and extension demands[9][15]. Multiple weight assignments have been suggested for the case that a different fault involve various biases of the input combinations applied to the circuit, to assure that a comparatively smaller number of patterns can detect all faults[10][11] [14]. Although, get bigger in the circuit complexity was in the view of the embedded memories verification more challenging. Among the obstacle encountered while testing the memories are 1. Controllability of the logic elements problem. The controllability is involved in the desired values on the internal signal of the circuit by exploiting an suitable test vector input combination to the inputs of the primary. 2. The Observability of the logic element is facing problem. Observability is an internal signal can be propagated through primary output for comparison with expected the value of the application of an appropriate primary input combination. 3. Insufficiency fault coverage in the testing of embedded memory.4. Sprouting the testing data to be stored and resolved. 5. The state of the art jet set and expensive testers are needed to test the embedded memory

III MOTIVATION OF THE BIST VERSES ATE

There are two main approaches for testing embedded memories: external test by direct access using Automatic Test Equipment, (ATE) and internal testing using Built-In-Self Test (BIST). When external testing is employed, the input test vectors and correct response data are stored in the ATE memory. For external testing, the comparison is carried out on the tester.ATE limitations make BIST technology an attractive alternative to external test for complex chips. BIST is a Design for Test method where part of the circuit is used to test the circuit itself where the test vectors are generated and test responses are analyzed on-chip.

Received: 27 Feb 2019 | Revised: 20 Mar 2019 | Accepted: 30 Apr 2020

ATE, as well as very long testing time since tester channels are timeshared by different memories under test. On the other hand, BIST provides at-speed and high bandwidth access to the embedded memory cores and it only needs a low cost to initialize the test sessions and to inspect the final results either status bits pass or fail. However, although BIST is may induce excessive power, in addition to performance and area overhead[12][13].

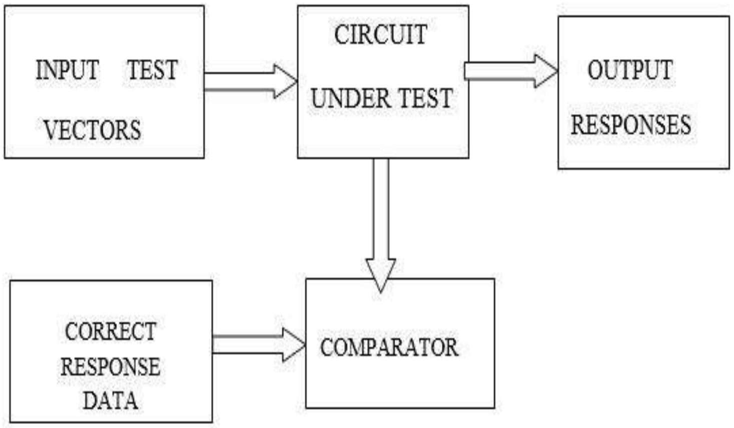


Figure. 2 Test pattern Application for BIST

IV PROPOSED ARCHITECTURE

Both Microcode-based and FSM-based controllers are two widely known architectures for programmable memory built-in self test process. These two widely techniques are much popular because of their flexibility of programming new test algorithms. In this proposed paper, the architectures for both controllers are designed to implement a new better test algorithm MARCH SAM, which gives a better fault coverage in both detecting single-cell fault and all intra-word coupling fault (CF).Memory built-in self test (MBIST) system mainly consists of three main components such as controller, comparator and fail/pass register. The MBIST controllers are either microcode-based or FSM-based controllers. Both of the controllers made of address generator, read/write sequence generator and data generator. The address generator produces memory address to be tested in ascending order, descending order and hold order.

Read/write sequence generator controls the assertion of test data to memory and retrieval of test data from the memory. The test data sequences are applied to the memory under test (MUT) and also wait at the comparator for comparison with tested data from the memory. The results (pass/fail) from the comparator is stored in the register. Some of the latest MBIST design are combined with micro code and Finite state machine based architectures are used to compensate the are versus speed issues Developing and MBIST engine for a memory or a group of memories and its verification can become a challenge when Time-To-Market (TTM) factor is considered. EDA Companies in the recent years had developed various tools for automating MBIST hardware generation it can be argued that adapting such MBIST engine generator could be a good alternative to develop an “in-house” MBIST architecture.

While most ASIC companies employ memory compilers to generate their required memories and microprocessors companies heavily concentrate on custom memory designs to meet their requirements in terms of speeds of GHz range, wide data buses ranging on addresses of notes, error detection and correction capabilities for reliability , and huge memory capacity. An efficient memory testing solution in terms of parameters such as test quality, test time and failure bit mapping capabilities would need to accommodate all these requirements, which may not be achievable with a generic commercial solutions. Furthermore, the MBIST based solutions should provide a bridge to the post – silicon activities. Availability of a high number of features, such as various operating modes and addressing modes, necessitates the existence of an efficient test program generation methodology. The MBIST engine should enable quick access to these operating modes to generate failure bit map for yield improvement and memory repair. To meet the challenges imposed by the state of the art memory designs and their requirements, we have decided to develop our code by MBIST RTL code generator as well as all the infrastructure to perform verification and post – silicon activities

V MICRO CODE BIST

Micro code or predefined instructions are used to wire the selec0ted test algorithms. The written tests are loaded into the Memory BIST controller. The Micro code-based type of BIST allows changes in the chosen test algorithm with no impact on the hardware of the controller. However the flexibility is there but cost of the higher logic is over head for the controller. A recent Micro-code based type of Memory BIST allows changes in the selected test algorithm with no impact of hardware of the controller. A recent Micro code based MBIST implementing modified proposed MBIST can be widely used for the embedded memory testing, especially under SOC design environment because of superior flexibility and expendability.

VI HARD WIRE –BASED BIST:

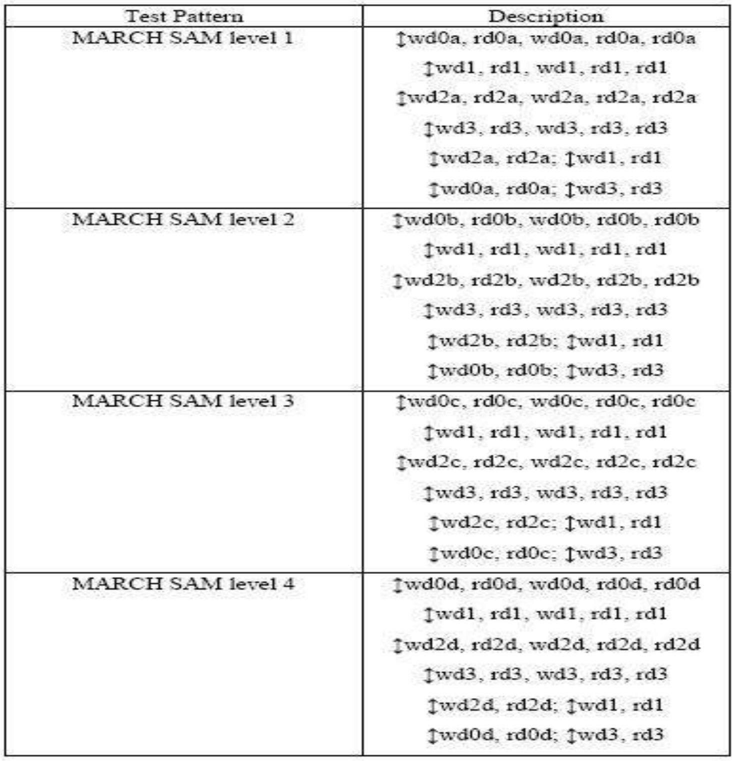
A hardwired-based controller is a hardware realization of a selected memory test algorithm, usually in the form of a Finite State Machine (FSM). This type of memory BIST architecture has optimum logic overhead, however, lacks the flexibility to accommodate any changes in the selected memory test algorithm. This results in re-design and reimplementation of the hardwired-based memory BIST for any minor changes in the selected memory test algorithm. Although it is the oldest memory BIST scheme amongst the three, hardwired-based BIST is still much in use and techniques have been kept developing. Table 2: Trade-offs between Different Memory BIST Schemes Table 1 gives the summary of this comparison of the three implementations. The four evaluation metrics used are: test time, area overhead, routing overhead, and flexibility. The routing overhead is directly translated into design efforts and time to market. The flexibility is.

VII MARCH SAM TEST ALGORITHM:

The test patterns used in the designs are MARCH SAM This algorithm is applied for all intra-word coupling fault detection. Table 2 shows types of test algorithms i.e. stored in the ROM and their read /write sequnces . Upward and downward arrows represents the ascending and descending order respectively. The d and d’ indicates the true and inverted of the chosen data back grounds(DBs). The notations of d 0-d9 are the specific DBs to test intraword coupling faults (CF). Design efforts and possesses the least flexibility. On the opposite end of spectrum, the Processor-based BIST is the most flexible, zero area or routing overhead, but incur long test time. The Microcode-based designs is somewhere in between these two extreme prototypes.

Received: 27 Feb 2019 | Revised: 20 Mar 2019 | Accepted: 30 Apr 2020

TABLE 2: MARCH SAM TEST ALGORITHM



Test Pattern	Description
MARCH SAM level 1	↕wd0a, rd0a, wd0a, rd0a, rd0a
	↕wd1, rd1, wd1, rd1, rd1
	↕wd2a, rd2a, wd2a, rd2a, rd2a
	↕wd3, rd3, wd3, rd3, rd3
	↕wd2a, rd2a; ↕wd1, rd1
	↕wd0a, rd0a; ↕wd3, rd3
MARCH SAM level 2	↕wd0b, rd0b, wd0b, rd0b, rd0b
	↕wd1, rd1, wd1, rd1, rd1
	↕wd2b, rd2b, wd2b, rd2b, rd2b
	↕wd3, rd3, wd3, rd3, rd3
	↕wd2b, rd2b; ↕wd1, rd1
	↕wd0b, rd0b; ↕wd3, rd3
MARCH SAM level 3	↕wd0c, rd0c, wd0c, rd0c, rd0c
	↕wd1, rd1, wd1, rd1, rd1
	↕wd2c, rd2c, wd2c, rd2c, rd2c
	↕wd3, rd3, wd3, rd3, rd3
	↕wd2c, rd2c; ↕wd1, rd1
	↕wd0c, rd0c; ↕wd3, rd3
MARCH SAM level 4	↕wd0d, rd0d, wd0d, rd0d, rd0d
	↕wd1, rd1, wd1, rd1, rd1
	↕wd2d, rd2d, wd2d, rd2d, rd2d
	↕wd3, rd3, wd3, rd3, rd3
	↕wd2d, rd2d; ↕wd1, rd1
	↕wd0d, rd0d; ↕wd3, rd3

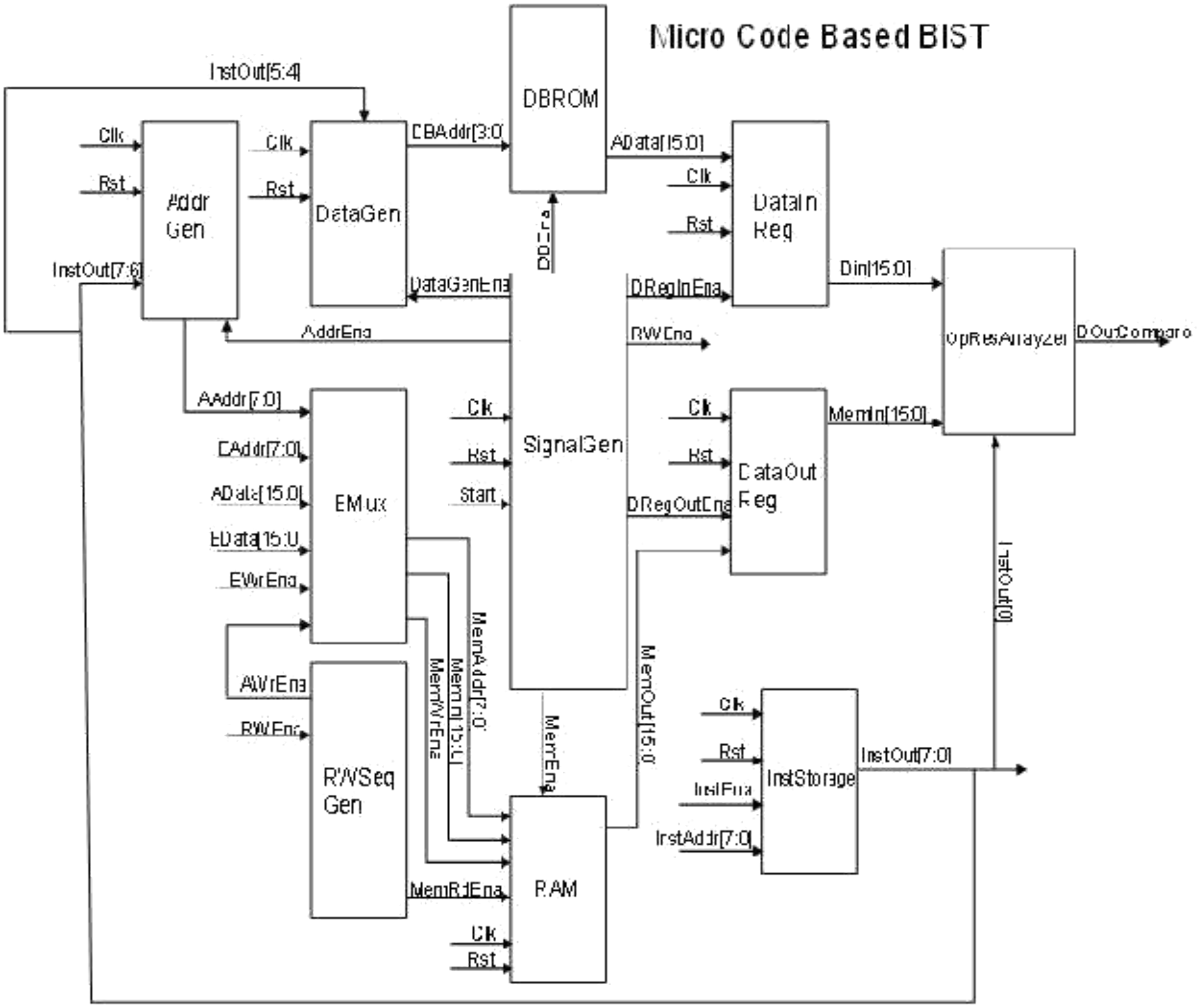


Figure. 3 Block Diagram of Micro Code Based Architecture

The controller comprises of instruction storage, address generator, data generator, read/write sequence generator, data input/output registers and output response analyzer. It uses counters to determine the instruction cycles and the selected data background .

The instruction storage stored all the MARCH SAM's microcode instructions. Each microcode instruction is 8 bit wide. Bit 7 and 6 represents the memory address status (increment, decrement or hold).Bit 5 selects the desired data background while bit 4 represents the true or inverted data background. Read/write sequences are controlled by bit 3. Looping feature is represented by bit 2 and bit 1 but this feature is temporary halted since it is not used in this design. The last bit is set to compare the expected data and the tested data. The last two bits i.e 7 & 6 are given to address generator which indicates whether to increment or decrement or hold the memory location in the RAM. the5and 4 bits are to data generator and the output of data generator is of 4 bits which indicates which data back ground to be selected is given to DBROM. The DBROM selects the desired data background whether to take the true value or inverted value of the background. Its output is given to the MUX which is used to take the address given by

Received: 27 Feb 2019 | Revised: 20 Mar 2019 | Accepted: 30 Apr 2020

the DBROM or the externally given address and the output of EMUX is given to RAM.Depending on whether to read or write bit the data from the RAM is read or data is written into the RAM. Another EMUX is used which takes the address given by address generator or an external address and the output of EMUX is given to RAM. The RAM gives the output data to the data output register. The output response analyzer compares both values present in data output and data input registers and gives the response to the pass/fail register

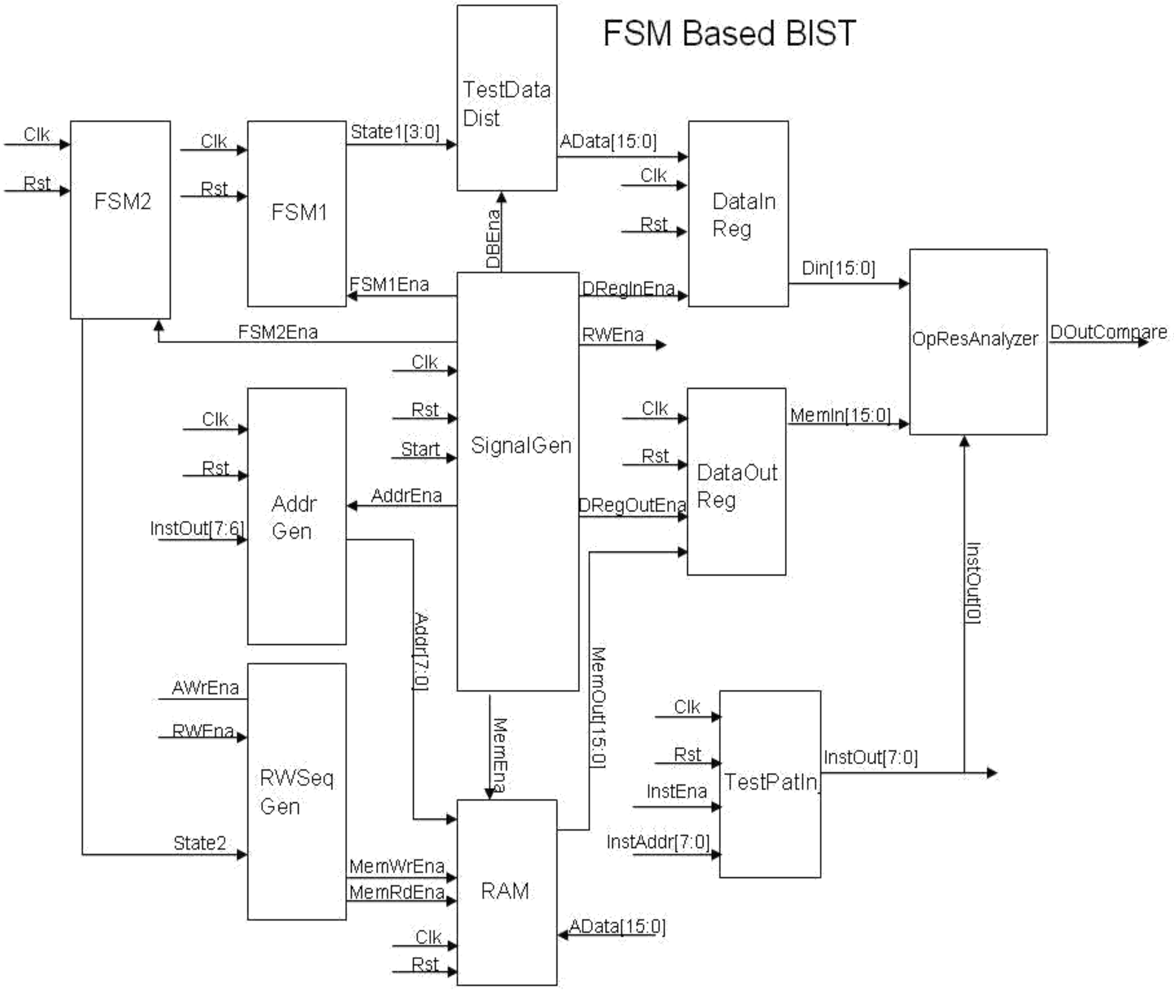


Figure.4 Block Diagram of Finite State Machine Architecture

The design of FSM based MBIST using MARCH SAM for WOM is developed. The controller comprises of 2 FSMs; state1 to inject each DB's sequences to be tested and state2 to control the read/write sequences. The test patterns can directly inputted in the FSM but to allow flexibility of inserting different kind of test patterns, ROM is chosen for storage. The controller embodies the test pattern injector block, test data distributor block, r/w sequences generator and address generator.State1 FSM has 11 states Sdb_init, Sdb0, Sdb1,Sdb2, Sdb3, Sdb4, Sdb5, Sdb6, Sdb7,Sdb8,sdb9,sdb10).Each states (Sdb0-Sdb10) will loop back to the state Sdb_init after their operation is finished. Counter at Sdb_init will determine which states will be treated next. Clk_db is the clock that controls the states' operation.State2 FM has 4 states namely Sx , Sr , Sw. Sx is the initialization state which are vital to be included in the design to ensure read/write sequences on each DBs follows any desired test algorithms. The pass and flag is controlled mainly by the memory clock and the active low of read/write sequences clock. If the state2 is Sr and the output equals the injected data, the pass flag will be up.

VIII SIMULATION RESUTLS AND FLOOR PLAN DESIGN

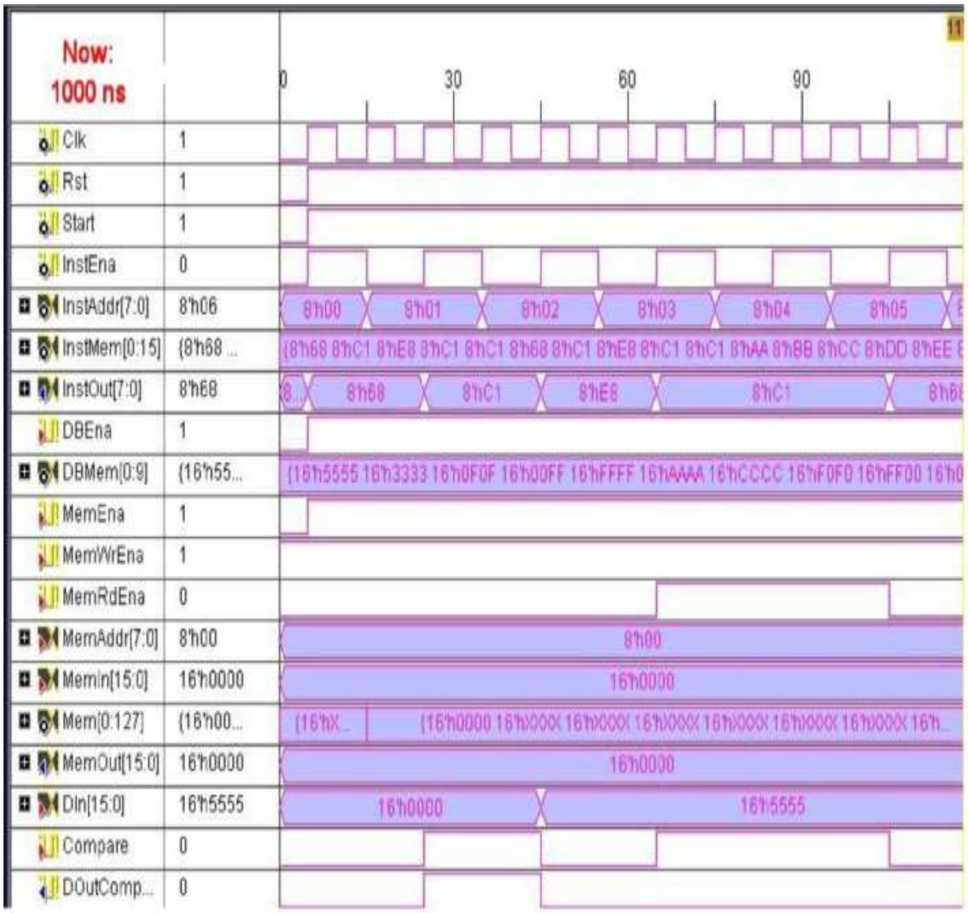


Figure 5. Top Module of Top Module for FSM Output

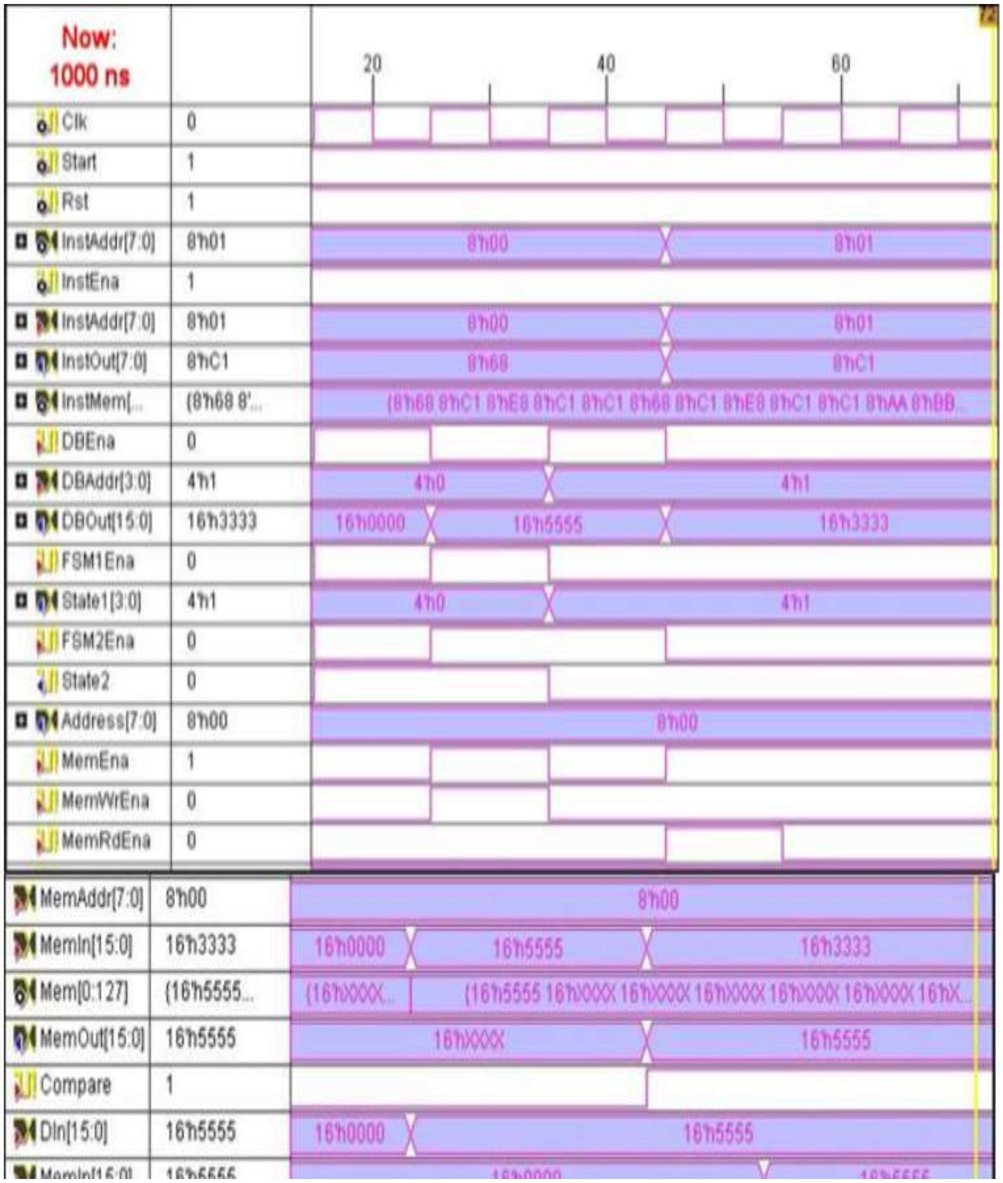


Figure .6 Top Module of Micro Code Based Output

Figure 7. floor plan Design for Finite State Machine BIST

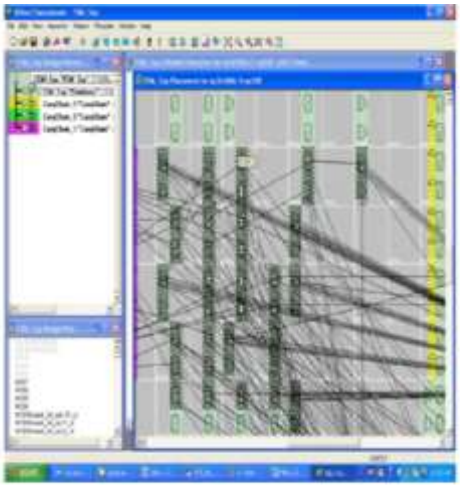


Figure.8 Floor plan design for Micro code based BIST

TABLE 3: AREA VS SPEED COMPARISON FOR FSMBASED BIST VS MICROCODE-BASED BIST

Types of controllers	**Area (No of LUTs utilized)**		**System Frequency in Mhz**	
	Existing work	Proposed work	Existing	Proposed work
FSM based Controller	77	116	80.27	195.377
Microcode Based Controller	86	355	176.87	212.262

In the Table 3 is having information about occupation of are with number of LUTs and speed in terms of frequency in MHz of proposed methods (FSM and Micro Code base controllers) as compared with existing work.

IX CONCLUSION

Micro Code based and FSM based MBIST controllers are designed to detect coupling faults. The architectures for both controllers are designed to implement a new test algorithm MARCH SAM that gives a better fault coverage in detecting single-cell fault and all intra-word coupling fault (CF).The components of each controllers are studied and designed. Both of the controllers are written using Verilog HDL. Architecture for Micro Code based and FSM based controllers is designed. Individual modules are designed and integrated. Simulation results for individual and integrated modules are verified. Synthesis results are obtained. Synthesis is done using Spartan3E device. Total equivalent gate count for Micro Code based MBIST Controller top module is 19,519 and for FSM based MBIST Controller top module is 5,647. So Micro Code based controller has higher area consumption compared to FSM based controller.

REFERENCES

[1] Manoharan, Rajesh, et al. "Selection of Intermediate Routes for Secure Data Communication Systems using Graph Theory Application and Grey Wolf Optimization Algorithm in MANETs." *IET Networks* (2020).

[2] Rajesh, M., Gnanasekar, J.M. Path Observation Based Physical Routing Protocol for Wireless Ad Hoc Networks. *Wireless Pers Commun* **97,** 1267–1289 (2017). https://doi.org/10.1007/s11277-017-4565-9

[3] Rajesh, M. Streamlining Radio Network Organizing Enlargement Towards Microcellular Frameworks. *Wireless Pers Commun* (2020). https://doi.org/10.1007/s11277-020-07336-9

[4] T. Q. Bui, L. D. Pham, H. M. Nguyen, V. T. Nguyen, T. C. Le, and T. Hoang, "An Effective Architecture of Memory Built-In Self-Test for Wide Range of SRAM," pp. 121–124, 2016.

[5] Becker, "Burst Software Transparent On-line MBIST." G. Harutyunyan, S. Shoukourian, V. Vardanian, and Y. Zorian, "A new method for march test algorithm generation and its application for fault detection in RAMs," *IEEE Trans. Comput. Des. Integr. Circuits Syst.*, vol. 31, no. 6, pp. 941–949, 2012.

[6] T. Q. Bui, L. D. Pham, H. M. Nguyen, V. T. Nguyen, T. C. Le, and T. Hoang, "An effective architecture of memory built-in self-test for wide range of SRAM," Proc. - 2016 Int. *Conf. Adv. Comput. Appl.* ACOMP 2016, pp. 121–124, 2017.

[7] "Available Online at www.ijarcs.info Original Image Bitstream Encoder Decoder Decoded Image," vol. 5, no. 4, pp. 28–29, 2014.

[8] Y. J. Huang, C. W. Chou, and J. F. Li, "A low-cost Built-in Self-Test scheme for an array of memories," 2010 *15th IEEE Eur. Test Symp. ETS'*10, pp. 75–80, 2010.

[9] P. Moorthy and M. S. S. Bharathy, "An Efficient Test Pattern Generator for High Fault Coverage in Built-In-Self-Test Applications," pp. 4–7, 2013.

[10] Y. J. M. Shirur, B. C. Bhimashankar, and V. S. Chakravarthi, "Performance analysis of low power microcode based asynchronous P-MBIST," 2015 *Int. Conf. Adv. Comput. Commun. Informatics, ICACCI 2015*, pp. 555–560, 2015.

[11] Q. L. Rao, C. He, and Y. M. Jia, "A memory built-in self-test architecture for memories different in size," 2009 *IEEE Circuits Syst. Int. Conf. Test. Diagnosis, ICTD'09, 2009*.

[12] P. Tsai, S. Wang, and F. Chang, "FSM-Based Programmable Memory BIST with Macro Command," Memory, 2005.

[13] H. WonGi, C. JungDai, and C. Hoon, "A programmable memory bist for embedded memory," 2008 *Int. SoC Des. Conf. ISOCC 2008*, vol. 2, pp. 195–198, 2008.

[14] H. WonGi, C. JungDai, and C. Hoon, "A programmable memory bist for embedded memory," *2008 Int. SoC Des. Conf. ISOCC 2008*, vol. 2, pp. 195–198, 2008. Raj, N., Singh, A., K. and Gupta, A. K., "Low power high output impedance high bandwidth QFGMOS current mirror", *Microelectronics Journal,* Vol. 45, No. 8, pp. 1132-1142, 2014. Raj, N., Singh, A.K., and Gupta, A.K., "Low-voltage bulk-driven self-biased cascode current mirror with bandwidth enhancement", *Electronics Letters*, Vol. 50, No. 1, pp. 23-25, 2014.

[15] S. Venkataraman, J. Ra jski, S. Hellebrand, and S. Tar- nick "An efficient BIST Scheme Based on Reseeding of Multiple Polynomial Linear Feedback Shift Registers," Proc. *Int. Conf. on Computer-Aided Design*, pp. 572-577, November 1993.

[16] I. Hamzaoglu and J. H. Patel, \New Techniques for Deterministic Test Pattern Generation", *Journal of Electronic Testing: Theory and Applications*, vol. 15, no. 1/2, pp. 63-73, October 1999.

[17] K. Chakrabarty and B. T. Murray, \Design of BuiltIn Test Generator Circuits Using Width Compression," *IEEE Trans. on Computer-Aided Design*, pp. 1044-1051, vol. 17, no. 10, October 1998.

[18] V. D. Agrawal, C. R. Kime and K. K. Saluja, A Tutorial on Built-in Self Test (Part 1*)," IEEE Design and Test of Computers*, pp. 73-82, March 1993.

Received: 27 Feb 2019 | Revised: 20 Mar 2019 | Accepted: 30 Apr 2020