e-ISSN: 2349-9745
p-ISSN: 2393-8161

Scientific Journal Impact Factor (SJIF): 1.711

International Journal of Modern Trends in Engineering and Research

www.ijmter.com

A Survey on the Architecture for an Efficient Memory Built in Self-Test For Configurable Embedded SRAM Memory

Nisha O. S.[1], Dr. K. Siva Sankar[2]

[1]*IT Department, Lourdes Matha College of Science and Technology*
[2]*IT Department, Nooral Islam University, Nagercoil, India*

Abstract-Today’s submicron VLSI technology has been emerged as integration of many VLSI ICs into a single Si Chip called System-on-Chip (SoC). The SoC architecture normally contains multiple processors along with either separate or centralized memory blocks as its core elements as well as many noncore elements Due to the increased demands for high data storage, the integration of on-chip memories ranging from Gigabytes to Terrabytes is becoming essential for the latest SoC technology. To improve the reliability and performance of SoCs due to technology miniaturization and increased memory density, there is a need to incorporate on-chip self-testing unit for testing these memory units. Further, to improve the yield and fault tolerance of on-chip memories without degradation on its performance, self-repair mechanism may be integrated on chip.

Keywords*:* MBIST, March algorithm, SRAM, Address generator

I. INTRODUCTION

Today’s semiconductor technology is making it feasible to integrate millions to billions of circuit elements e.g., diodes, transistors and other components such as resistors and capacitors, together with interconnections, within a very small Silicon area [1]. The shrinking of technology has created various mechanical and physical defects such as opens, bridges and shorts causing more parasitic capacitances, leakage power and their severe effects on the overall performance of the System-on-Chips (SoCs). Earlier, the SoCs were influenced by its functional cores, but because of the increased demand of high data storage, SoC area is currently being influenced by the on-chip Volatile or Non-volatile memory blocks [2], [3]. As a result of the survey made by Semiconductor Industry Association (SIA), the semiconductor memories are expected to occupy 94% of SoC area in 2014 as compared to 71% in 2005.The user data will be generally stored in volatile SRAM, whereas the nonvolatile ROMs store the system programs as well as the test program and test vectors by which the system will be tested for any manufacturing defects and functional errors during test mode [4]. The exact functionality of the memory chips is becoming vital in SoCs, because the control data/signals used for controlling and functioning of almost all the blocks in SoCs, their scheduling information in addition to the user data are being stored on-chip to reduce the latency as compared to that of storage of these data in an off-chip storage device.

The most commonly occurring functional faults [5], [6] in regular 2D memory arrays include Address decoder Faults (AFs), Stuck-At Faults (SAFs), Neighborhood Pattern Sensitive Faults (NPSFs) and Coupling Faults (CFs). As the memories in SoCs are mostly occupying the higher chip area, the study and application of fault models that can target these memory faults has become significant for cost effective memory testing. Unlike the logic blocks, the memory arrays have an identical 2-dimensinal structures, the MBIST algorithms may be commonly applied for testing of all memory arrays available in SoCs. The memory can be effectively tested using march algorithms [7], [8].

Power dissipation is becoming a challenging problem for the VLSI design engineers and testing engineers because the power consumed by the system in testing mode is 200% more than in its normal mode [9]. There are two types of power dissipation 1. Static power 2. Dynamic power and in this case we are going to consider the dynamic power dissipation .The dynamic power dissipation is calculated from an equation

$$P_{avg} = \alpha T . C_{load} V^{2}{}_{dd} \cdot f_{Clk} \qquad (1)$$

Where,

αT -Switching activity factor of the gate.

C_{load} -Total load capacitance

V_{dd} -Supply voltage

f_{Clk} -Operating frequency

In the above equation the average power is directly proportional to the ‘αT’. Therefore the power dissipation during testing can be reduced by controlling the switching activity .The advancement in submicron manufacturing technology and system-on-a-chip (SOC) design methodology has led to a large number of cores, especially the memory cores, are now integrated into a single chip. It has been predicted that by the year of 2014, memory cores may occupy 94% area of a typical SOC [10]. Memory thus plays an important role in SOC. Since the probability of memory fault is more compared to that of other type of faults in a circuit the need for testing of memory is more important. However due to the availability of a small number of I/O pins in a circuit BIST for Memory (MBIST) is used as a solution to this problem [11].In conventional MBIST the address bus, data bus, and read/write control signals that are generated by the test pattern generator of the BIST are applied to the memory under test. The address bus indicates the memory location, the Read/write control determines the operation (read/write) to perform in this particular memory location and the data bus include the data to write or to read in the memory location the address bus indicate.

This paper is organized as follows: section 2 describes about Built in Self-Test Architecture; Section 3 describes The Built-in-Self Test/Repair techniques and their architectures for self-testing/repairing of embedded memories are presented; Section 4 describes the March Algorithm; and Section 5 summarizes some concluding remarks.

II. BUILT-IN SELF TEST ARCHITECTURE

Built-In Self-Test (BIST) mechanism [20] used for testing of manufactured ICs has the capability of testing the circuit itself by incorporating the Automatic Test Pattern Generator (ATPG) and Output Response Analyzer (ORA) on-chip within a marginally increased logic overhead and Silicon area. The BIST techniques are classified into (i) Online BIST and (ii) Off-line BIST. Using BIST, the circuit can be tested both on-line and off-line. In on-line BIST, the testing will be carried during the interval when normal operation carries. The online testing however is capable of detecting the faults in the circuit, cannot be used for fault diagnosis. In off-line BIST, the circuit is normally tested at the time of system boot up and/or during system on reset period by executing self-test program with pre-defined test vectors stored in non-volatile memories. The offline BIST may also be carried out periodically by suspending the CUT from its functional mode of operation. The off-line testing being a periodic testing does not guarantee the detection of temporary and soft-error faults. However, the offline BIST are normally preferred due to its fault diagnosis capability and the possible repair work.

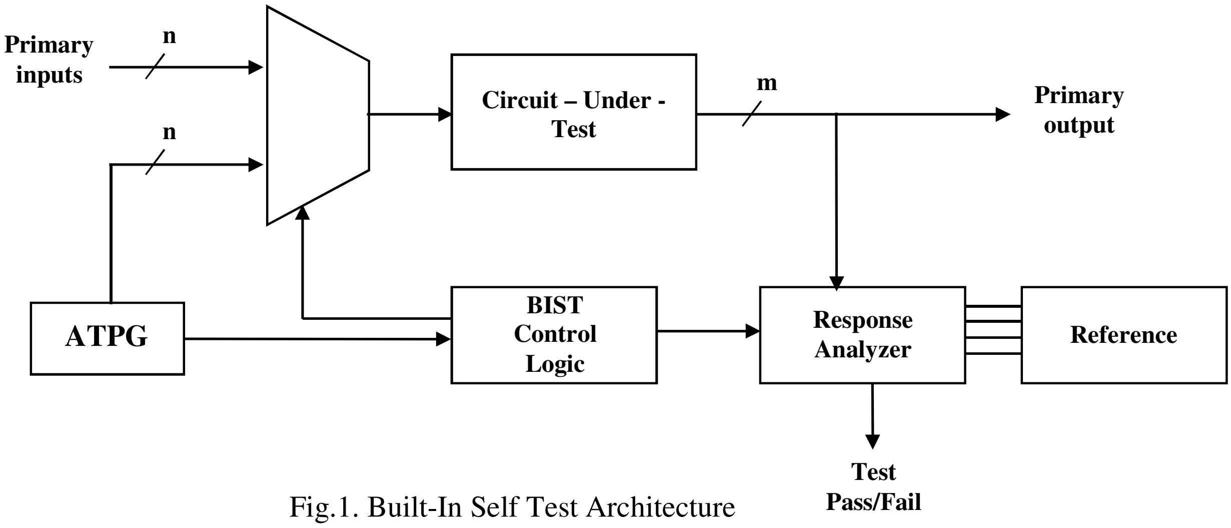

Fig.1. Built-In Self Test Architecture

A generalized BIST architecture shown in Figure 1 consists of a Test Pattern Generator (TPG), Circuit-Under- Test (CUT) and Output Response Analyser (ORA). The TPG produces test vectors for the CUT during off-line test. The ORA compares the CUT output with the reference outputs to check whether the circuit is faulty or not. The reference patterns may be stored in a non-volatile storage. For reducing the size off-line storage memory, the reference patterns and the corresponding CUT output responses may be compressed and/or compacted and stored. The BIST controller produces the required control and timing signals for proper functioning of BIST architecture.

III. MEMORY BUILT-IN SELF TEST/REPAIR (MBISTR) ARCHITECTURE

As mentioned earlier, a major portion of Silicon area in SoCs is being dominated by on-chip memories. The integration of high capacity memories in a single Silicon chip has resulted in the appearance of various faults such as Stuck-at Faults (SAFs), Transition Faults (TFs), Coupling Faults (CFs), Address Decoding Faults (ADFs) and Physical neighbourhood pattern-sensitive faults (NPSFs). These commonly occurring faults in memory arrays can be effectively addressed using Memory BIST (MBIST) techniques, which is briefly described in the next sub-section. The subsequent sub-section discusses the Memory Built-In Self Repair (MBISR) technique [21] which allows the test and repair of memory with the help of redundant rows or columns.

A. Memory built-in self-test (mbist) architecture

The memory devices have a densely packed memory cells in two dimensional (2-D) structures called memory arrays as core memory along with Row- and Column address decoders and sense amplifiers. With memory BIST architecture shown in Figure 2, the testing of the entire memory can be implemented on-chip. The self-testing of embedded memories [22] requires the following test hardware equipped with the memory-under test (MUT)

a. An Address generator for off-line BIST operation.
b. A MUX circuit feeding the memory during self-test from the controller.
c. A Comparator for response checking.
d. MBIST Controller.

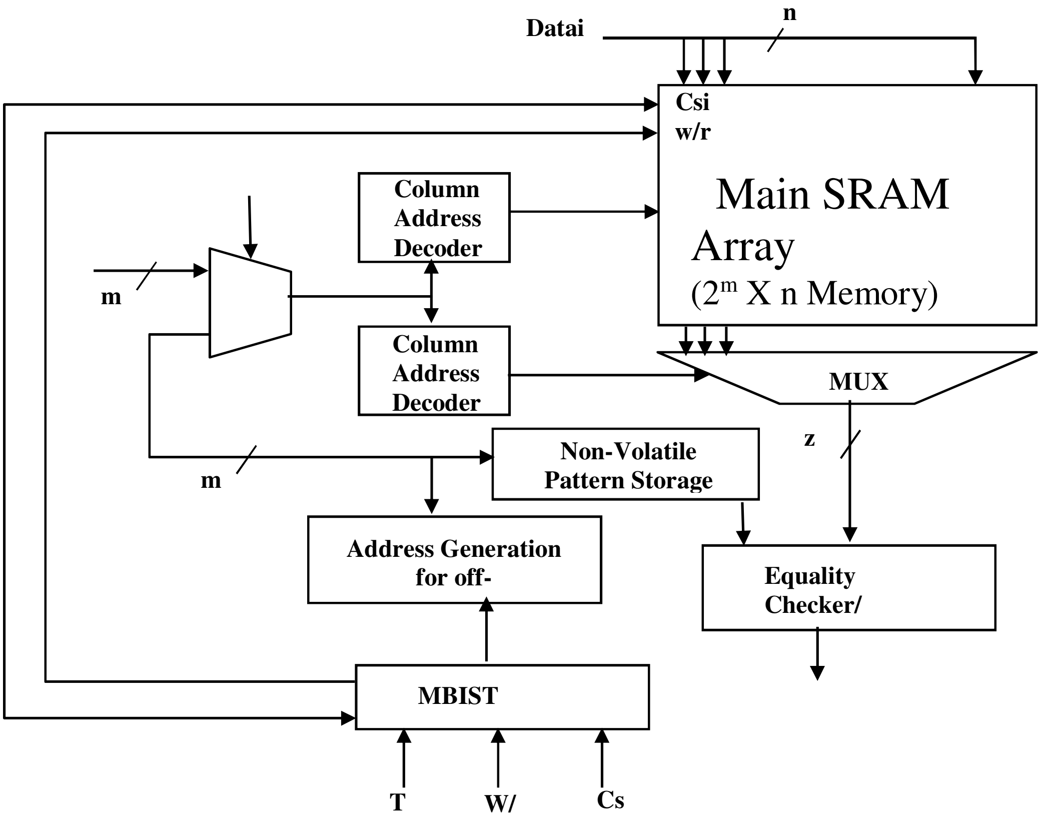

Fig.2. Memory Built-In Self-Test (MBIST) Architecture

The architecture of memory is a 2-dimensional regular array and hence Algorithmic Test Sequences (ATS) may be adopted for the functionality testing of memories. The March-based algorithms [23] are being extensively used for Built-in Self-Test of Memory arrays. The march-based algorithms involve a finite sequence of write and read operations, called March elements. A sequence of March operations are performed to the memory cells to check whether the given memory cell is fault-free or not. During March operation, the memory cells may be addressed either in an ascending or descending order. Table 1 [24] shows the general notations used for memory addressing and memory read/write operations in March algorithms.

TABLE I. DESCRIPTION OF NOTATIONS

Notation	Description
R0/1	The response 0/1 from a cell during memory read operation
W0/1	The data 0/1 written into a cell during memory write operation
⇕	Addressing with any order
⇑	Addressing with increasing order
⇓	Addressing with decreasing order
N	Number of address locations/memory cells

B. Memory built-in self repair (mbisr) architecture

The MBIST is capable of detecting the presence of faults in the memory array so that the faulty memory block may be replaced by the spare units available within the system-on chip. The MBISR [23] supports fault-diagnosis by self-repairing the Memory- under Test in case it is found to be a faulty. Various self-test mechanisms discussed in the previous section may be used for detection of faults in memory array. The MBISR architecture shown in Figure 3 consists of a Built-In Self- Test (BIST) module, a redundancy logic, a spare location locator and MBISR controller. The Built-In Self-Test of memory arrays is performed using one of the efficient March algorithms. Once one or more memory locations are found to be faulty locations, the faulty addresses are stored in redundancy logic. The redundancy logic can be either spare columns or rows [22, 23] or a block of main memory not being used as normal memory during BISR operation. The spare location allocator maps the input address to one of the address location in the redundant area. During memory write operation, the data will be written in both user memory as well as the mapped address location of the redundant array simultaneously. The multiplexer placed at the output will accept correct data either from main memory or from redundancy array based on the absence or presence of faults in the memory location being addressed respectively. The BISR controller generates various control signals for proper operation of the BISR circuitry.

The capability and complexity of BISR lies on the size of the redundant memory array. The large redundant array however enhances the number of faulty memory locations it can tolerate but increases the area and complexity.

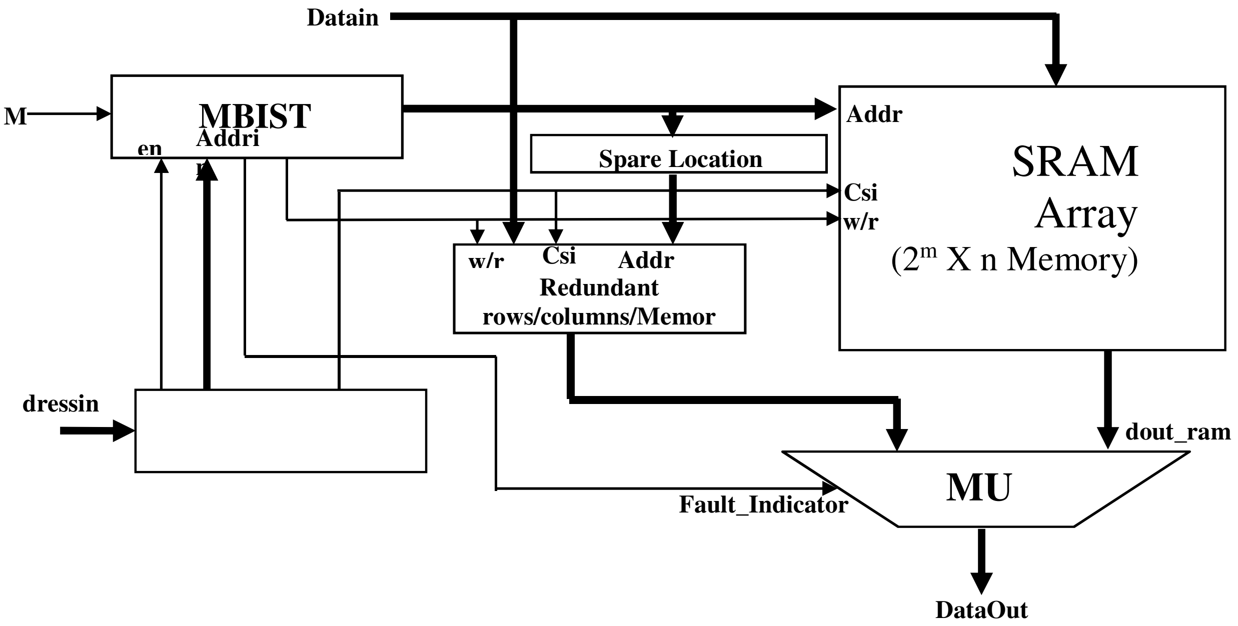

Fig. Memory Built-In Self Repair (MBISR) Architecture

The previous literatures have proposed various memory self- test algorithms as listed in Table 2. These March algorithms [4-7] differ in terms of number of March elements, test sequences and targeted faults in the memory. In general, the number and the order of memory read/write operations depend on the targeted faults.

TABLE II. COMPARISONS OF VARIOUS MARCH ALGORITHMS

Sl. no:	March Algorithm	No:March elements	No:March steps	Test Sequence	Targeted Faults
1	ATS	4N	3	$\{\Updownarrow W0; \Uparrow (R0, W1), R1\}$	SAFs,ADFs
2	MATS	4N	3	$\{\Updownarrow W0; \Uparrow (R0, W1), R1\}$	SAFs,ADFs
3	MATS +	5N	3	$\{\Updownarrow W0; \Uparrow (R0, W1), (R1, W0)\}$	SAFs,ADFs
4	MATS ++	6N	3	$\{\Updownarrow W0; \Uparrow (R0, W1), (R1, W0, R0)\}$	SAFs,ADFs,TTFs,CFs
5	MARCH A	15N	5	$\Updownarrow W0, \Uparrow (R0, W1, W0, W1), \Downarrow (R1, W0, W1),$ $\Downarrow ((R1, W0, W1, W0), \Downarrow (R0, W1, W0))\}$	SAFs,ADFs,TTFs

6	MARCH B	17N	5	$\Updownarrow W0, \Uparrow (R0,W1,R1,W0,R0,W1),$ $\Uparrow (R1,W0,W1), \Downarrow ((R1,W0,W1,W0)),$ $\Downarrow (R0,W1,W0)\}$	SAFs,ADFs,TTFs,CFs
7	MARCH C	11N	7	$\{\Updownarrow W0; \Uparrow (R0,W1), \Uparrow (R1,W0),$ $\Updownarrow R0, \Downarrow (R0,W1), \Downarrow (R1,W0), \Updownarrow R0\}$	SAFs,ADFs,TTFs,some CFs
8	MARCH X	6N	4	$\{\Updownarrow W0; \Uparrow (R0,W1),(R1,W0 \Updownarrow R0)\}$	CFs
9	MARCH Y	8N	4	$\{\Updownarrow W0; \Uparrow (R0,W1,R1), \Downarrow$ $(R1,W0,R0 \Updownarrow R0)\}$	SAFs,ADFs,TTFs,CFs

IV. MARCH ALGORITHM

A *MARCH test* consists of a finite sequence of March elements, while a *March element* is a finite sequence of operations applied to every cell in the memory array before proceeding to the next cell. An *operation* can consist of writing a 0 into a cell (w0), writing a 1 into a cell (w1), reading an expected 0 from a cell (r0), and reading an expected 1 from a cell (r1).

MARCH Test Notations:

Some of the most popular notations for MARCH tests which will be used through out his paper are shown below

⇓:*Addresss sequence changes in ascending order*

⇑: *Addresses sequence changes in descending order*

⇕: *Address sequence can change either way*

R0: read operation (reading a 0 from a cell)

R1: read operation (reading a 1 from a cell)

W0: write operation (writing a 0 to a cell)

W1: write operation (writing a 1 to a cell)

SRAM

Static random-access memory (SRAM or static RAM) is a type of semiconductor memory that uses bistable latching circuitry to store each bit. The term *static* differentiates it from *dynamic* RAM (DRAM) which must be periodically refreshed. SRAM exhibits data remanence but it is still *volatile* in the conventional sense that data is eventually lost when the memory is not powered.

SRAM (static RAM) is random access memory that retains data bits in its memory as long as power is being supplied. Unlike dynamic RAM (DRAM), which stores bits in cells consisting of a capacitor and a transistor, SRAM does not have to be periodically refreshed. Static RAM provides faster access to data and is more expensive than DRAM. SRAM is used for a computer's cache memory and as part of the random access memory digital-to-analog converter on a video card .SRAM is more expensive and less dense than DRAM and is therefore not used for high-capacity, low-cost applications such as the main memory in personal computers.

V. RELATED WORKS

K. Murali Krishna and M. Sailaja in [12] implemented an LFSR based address generator with reduction in switching activity for low power MBIST. In that method the address patterns were generated by a combination of LFSR and a 2-bit pattern generator (Modified LFSR) and two separate clock signals. By using the modified architecture switching activity was reduced. Since the switching activity was proportional to the power consumed, reducing the switching activity of the address generator reduces the power consumption of the MBIST. They had designed and stimulated their address generator using Xilinx ISE tools and compared with the switching activities of the conventional LFSR and BS-LFSR. Results showed a reduction in switching activity and a reduction of more than 90% of the total dynamic power when compared to conventional LFSR.

Balwinder Singh *et.al* in [13] implemented all March algorithms used for memory testing in verilog for the testing of 1Kb and 4kbmemories with BIST. After designing and implementation, their performance is compared on the basis of their length, number of cycles used during writing and reading of memory, and area overhead. All that comparison was presented in the tabular form. From comparison on the basis of area overhead, it is observed that for the memories of smaller in size March X is most efficient and March AB is the least efficient, and Vice versa in the case of March Y and March LA . During the analysis power consumption it was examined that March X and March C consume least power in case of 1 Kb memory and 4 Kb memories respectively. March C, X and Y takes the least number of cycles for memory writing and reading in 1Kbmemory and March Y in case of 4 Kb memory.

Che-Wei Chou *et.al* in [14] presented a low-cost built-in self-diagnosis (BISD) scheme for NAND flash memories, which could support the March-like test algorithms with page-oriented data backgrounds. Two simple test time reduction techniques were also designed to reduce the test time. Experimental results showed that the proposed BISD circuit for a 2M-bit flash memory only needs 1.7K gates. Also, their new test time reduction techniques could effectively reduce the test time. Analysis results showed that they can reduce the test time to 48.628% of the normal test scheme for a 4G-bit flash memory tested by the March-FT test algorithm with solid data backgrounds.

A systematic approach in testing flash memories, including the development of March-like test algorithms, cost effective fault diagnosis methodology, and built-in self-test (BIST)scheme was presented in [15] by Jen-Chieh Yeh *et.al* . The improved March-like tests algorithms can detect disturb faults derived from the IEEE STD 1005 and conventional faults. As the memory array architecture and/or cell structure varies, the targeted fault set may change. They have developed a flash-memory fault simulator called RAMSES-FT, with which they could easily analyze and verify the coverage of targeted faults under any given test algorithm. In addition, the RAM test algorithm generator test algorithm generator by simulation has been enhanced based on RAMSES-FT, so that one can easily generate tests for flash memories, whether they are bit- or word-oriented. Their newly designed fault diagnosis methodology helps improve the production yield. They also developed a built-in self-diagnosis (BISD) scheme a BIST design with diagnosis support. The BISD circuit collects useful test information for off-chip diagnostic analysis. It had unique test mode control that reduces test time and diagnostic data shift-out cycles by a parallel shift-out mechanism.

In [16] Masnita et.al, had designed a data and read/write controller as a finite state machine (FSM) BIST that would generate test patterns based on the march-based diagnostic algorithm developed to distinguish between stuck-at and transition faults. A description related to SAFs and TFs were presented with the intention of covering the aspect of distinguishing both of these faults. Related design based on the selected algorithm was also presented with simulation results to show the functionality of the design. The design of the controller can be used to build a complete MBIST engine to test the effectiveness of the proposed algorithm in distinguishing SAFs from TFs.

Manikandan.B and Praveen kumar.J implemented a FSM-based programmable memory built in self test (MBIST) Controller used for testing the memory devices in [17]. The MBIST controller was designed to implement a new test algorithm known as March based test algorithm. The controller and test algorithm are studied and designed using verilog HDL and implemented in SPARTAN-3E FPGA. The simulation portrays that the tested data and the expected data are able to be compared in the architecture. The implemented controller has the ability to detect faulty or good memory ICs. Synthesis result shows that the FSM -based HP-MBIST controller employed only 75 instances with clock frequency 246.15 MHz with a less usage of Logic Elements (LE) with High speed testing of memories. It was also justified that the FSM-based HP-MBIST controller consumes less area overhead and high speed while the other compared designs consumed more area overhead and less speed. The experimental results also showed that the presented BIST can be implemented with low area overhead.

K Padma Priya in [18] presented a high speed FSM-based controller for programmable memory built-in self-test for testing memory devices. Her technique was popular because of its flexibility of new test algorithms. The architecture of controller was designed to implement a new test algorithm has less number of operations and the designed algorithm emphasis testing of high density memory ICs either faulty or good .The components of controller was analyzed and designed using Verilog HDL. The analysis of the timing, logic area usage and speed are also presented.

In [19] M. Jahnavi *et.al.* Presented the implementation of online test scheme for RFID memories based on Memory Built in Self-Test (MBIST) architecture. In that work they presented the, Symmetric transparent version of March SS algorithm, implementation of Memory BIST. The comparison between the different march algorithms and the advantage of the March SS algorithm over all other is also presented. The whole design was implemented using Verilog HDL and was, in turn, verified on Xilinx ISE 13.2 simulator, and synthesized.

VI. CONCLUSION

The literature survey revealed that a lot of researches have gone in this area for a low power consumption system. This paper presents various methods for memory testing. Another peculiarity of this paper is the introduction of various test mechanisms to improve the reliability and performance of SoCs due to technology miniaturization and increased memory density, there is a need to incorporate on-chip self-testing unit for testing these memory units**.** Most recent research areas trying to incorporate all the techniques described in this paper to make a perfect architecture for an efficient memory built in self-test.

REFERENCE

[1] Kiran Gupta and T R.Gopalakrishnan.Nair , “A Study of VLSI Technology, Wafers and Impact on Nanotechnology”, IEEE National Conference on Advanced VLSI/DSP, Bangalore, 2008.

[2] David Choi, Kyu Choi, and John D. Villasenor, “New Non-Volatile Memory Structures for

FPGA Architectures”, IEEE Transactions on Very Large Scale Integration (VLSI) Systems, Vol. 16, No. 7, July 2008.

[3] Technical report on “Non-Volatile Memory: Emerging Technologies And their Impacts on Memory Systems “Porto Alegre, Sep 2010.

[4] Venkatapathi Naidu Rayapati, “VLSI semiconductor random access memory functional testing”, Microelectronics Reliability, Vol.30, No 5, pp: 877–889, 1990.

[5] R. Dekker, F. Beenker, A. Thijssen, “Fault modeling and test algorithm development for static random access memories,” International Test Conference 1988.

[6] D. Niggemeyer, M. Redeker, J. Otterstedt, "Integration of nonclassical faults in standard march tests," International Test Conference 1998, pp. 91-6.

[7] S. Hamdioui, A. J. van de Goor, and M. Rodgers, "March SS: A test for all static simple RAM faults," in *Proc. IEEE Workshop MTDT, Jul.* 2002, pp. 95–100.

[8]. L. Dilillo, P. Girard, S. Pravossoudovitch, and A. Virazel, "Dynamic read destructive fault in embedded-SRAMs: Analysis and March test solution," in *Proc. IEEE Eur. Test Symp., May 2004, pp. 140–145.*

[9]Luca Benini, Member, Alessandro Bogliolo, and Giovanni De Micheli, "A Survey of Design Techniques for System-Level Dynamic Power Management," IEEE Transactions on Very Large Scale Integration (VLSI) Systems, Vol. 8, No. 3, June 2000.

[10] E. J. Marinissen, B. Prince, D. Keitel-Schulz, Y. Zorian, "Challenges in Embedded Memory Design and Test", Proc. of Design, Automation and Test in Europe, 2005, pp. 722 – 727.

[11] NurQ.Noor, YusrinaYusof, and AzilahSparon, "Low Area FSM-Based Memory BIST for Synchronous SRAM", proceeding of the international colloquium of Signal Processing and Its application, 2009, pp. 409-412.

[12] K. Murali Krishna and M. Sailaja, "Low Power Memory Built in Self Test Address Generator Using Clock Controlled Linear Feedback Shift Registers", Journal of Electronic Testing, Theory and Applications (JETTA), Vol.30, No.1, pp: 77-85, Feb.2014.

[13] Balwinder Singh, Arun Khosla and Sukhleen Bindra Narang, "Area Overhead and Power Analysis of March Algorithms for Memory BIST", International Conference on Communication Technology and System Design, pp: 930 – 936, 2012.

[14] Che-Wei Chou ,Chih-Sheng Hou and Jin-Fu Li "Built-in self-diagnosis and test time reduction techniques for NAND flash memories" International Symposium on VLSI Design, Automation and Test (VLSI-DAT),April 2011.

[15] Jen-Chieh Yeh, Kuo-Liang Cheng, Cheng-Wen Wu and Yung-Fa Chou, "Flash Memory Testing and Built-In Self-Diagnosis with March-Like Test Algorithms", IEEE Transactions on Computer-Aided Design Of Integrated Circuits and Systems, Vol. 26, No. 6, pp: 1101-1113, JUNE 2007.

[16] Masnita, Wan Zuha W.H, R.M. Sidek and A. Halin, "The data and read/write controller for March-based SRAM diagnostic algorithm MBIST", proceedings of 2009 IEEE conference on Research and Development, Nov.2009.

[17] Manikandan.B and Praveen kumar.J, "A VLSI Implementation of High Speed FSM-based programmable Memory BIST Controller", Journal of Electronics and Communication Engineering Research, Vol.1, No.2, pp: 01-06, 2013.

[18] K.Padma Priya, "High Speed FSM-based programmable Memory Built-In Self-Test (MBIST) Controller", International Journal of Computer Science and Mobile Computing, Vol. 2, No. 2, pp:.46-52, February 2013.

[19] M. Jahnavi, P. S. Indrani, M. J. C. Prasad, "Implementation of Concurrent Online MBIST for RFID Memories using March SS Algorithm", International Journal of Innovative Technology and Exploring Engineering (IJITEE), Vol.3, No.6, November 2013.

[20] G. Prasad Acharya, M. Asha Rani, "Survey of Test Strategies for System-On Chip and it's Embedded Memories", 2013 IEEE Recent Advances in Intelligent Computational Systems (RAICS),

[21] M.H. Tehranipour, Z. Navabi, S.M. Fakhraie, "An efficient BIST method for testing of embedded SRAMs," International Symposium on Circuits and Systems 2001, pp. 73-6.

[22] M. Franklin, K.K. Saluja, "Embedded RAM testing," IEEE Memory Technology, Design, and Test Workshop 1995, pp. 29-33.

[23] D. Youn, T. Kim, S. Park, "A microcode-based memory BIST implementing modified march algorithm," Asian Test Symposium 2001, pp. 391-5.

[24] J. van de Goor, "Using march tests to test SRAMs," IEEE Design &Test of Computers, vol. 10, no. 1, pp. 8–14, Mar. 1993.