IJDACR
ISSN: 2319-4863

International Journal of Digital Application & Contemporary research
Website: www.ijdacr.com (Volume 2, Issue 2, September 2013)

Built-In-Self Test for Embedded Memories by Finite State Machine

Swati Singh
singh06swatigmail.com

Dr. UBS Chandrawat
uday22bana@yahoo.co.in

Abstract – **In today's Integrated Circuits (IC's) designs Built-in Self-Test (BIST) is becoming important for the memory which is the most necessary part of the System on Chip. The March algorithm has been widely used to test memory core of System on chip (SOC).This paper describes the FSM based programmable BIST which is improved method as compared to previous method in terms of area (having lower gate counts). This technique is useful both for engineering debug as well as for product testing.**

Keywords – **IC, BIST, SOC.**

I. INTRODUCTION

In present scenario, memories are essential component for most of the electronic equipment. About all system chips contain some type of embedded memory, for example ROM, SRAM, DRAM, and flash memory. In the computer world, Alpha 21264, for example, has cache RAMs which represent 2/3 of total number of transistors in use and 1/3 of the entire chip area. In the embedded domain, embedded RAMs of the strong ArmSA110 occupy 90% of the total area. The projection is, by 2010, memory will represent more than 90% of the chip area in an average SOC environment. With the arrival of deep-submicron VLSI technology, the computer storage density and capacity is rising. The clock frequency is never higher. The predominant use of embedded memory cores along with emerging new architectures and technologies make providing a low cost test solution for these on-chip memories a very challenging task. Built-in self-test (BIST) has been shown to be one of the most cost-effective and widely used solutions for memory testing for the following reasons:

1. No external test equipment.
2. Reduced development efforts.
3. Tests can run at circuit speed to generate a more realistic test time.
4. On-chip test pattern generation to deliver higher controllability and observability.
5. On-chip response analysis.
6. Test can be on-line or off-line.
7. Flexibility to engineering changes.
8. Easier burn-in support.

II. BIST

In Built-In Self-Test (BIST), test generation and response evaluation hardware are included on-chip so that in-circuit tests can be performed with minimal need of external test equipment, if any. The standard BIST set-up is shown in figure 1. Under normal operation conditions, the additional test circuitry is transparent to the functionality of the device. Built-in self-test (BIST) is common used as a technique to test embedded arrays such as RAMs and ROMs. The primary use of BIST is for manufacturing or production testing but additional features can be added for diagnostics and debug. Conventional memory BIST implementations provide Pass/Fail information, which is usually sufficient for manufacturing test.

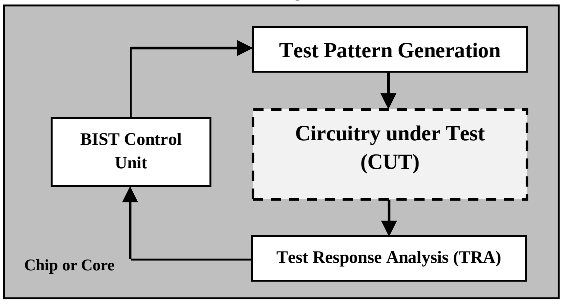

Figure 1: Programmable BIST Architecture [1]

There are two types of BIST; On-line and Off-line.

- On-line BIST has tests implemented on-chip. It has shorter test time but an area overhead of one to three percent.
- Off-line BIST, on the other hand has tests implemented off-chip. It has longer test time but no area overhead.
- On-line BIST can further be classified into three subgroups: Concurrent BIST, Non-Concurrent and Transparent BIST.
    - Concurrent BIST is a memory test mechanism where the memory can be

IJDACR
International Journal Of Digital Application & Contemporary Research

International Journal of Digital Application & Contemporary research
Website: www.ijdacr.com (Volume 2, Issue 2, September 2013)

tested concurrently with normal system operation. Thus, it has instant error detection and possible correction, but all faults will be detected within the restrictions of the method used. There is also certain hardware overhead associated with this scheme. For example, we need logic to write out, read in, and store redundant information generated during the test process. There will also incur certain performance penalty upon every memory access.

- Non-Concurrent BIST is test mechanism that requires interruption of the normal system function in order to perform tests; usually a special test mode is required. The advantage is there is no need to preserve the data yields certain space savings. The disadvantage is the circuit cannot detect faults that are not covered by the fault models used.
- Transparent BIST scheme is very similar to the Non-Concurrent scheme except the memory contents are preserved. Sliding Diagonal, Butterfly, MOVI, and etc., due to these imbalanced conflicting traits, the popularity of these algorithms is decreasing. MATS, MATS+, Marching 1/0, March C-, March Y, March A, March B, and etc. Since March-based tests are all simple and possess good fault coverage, they are the dominant test algorithms implemented in most modern memory BIST.

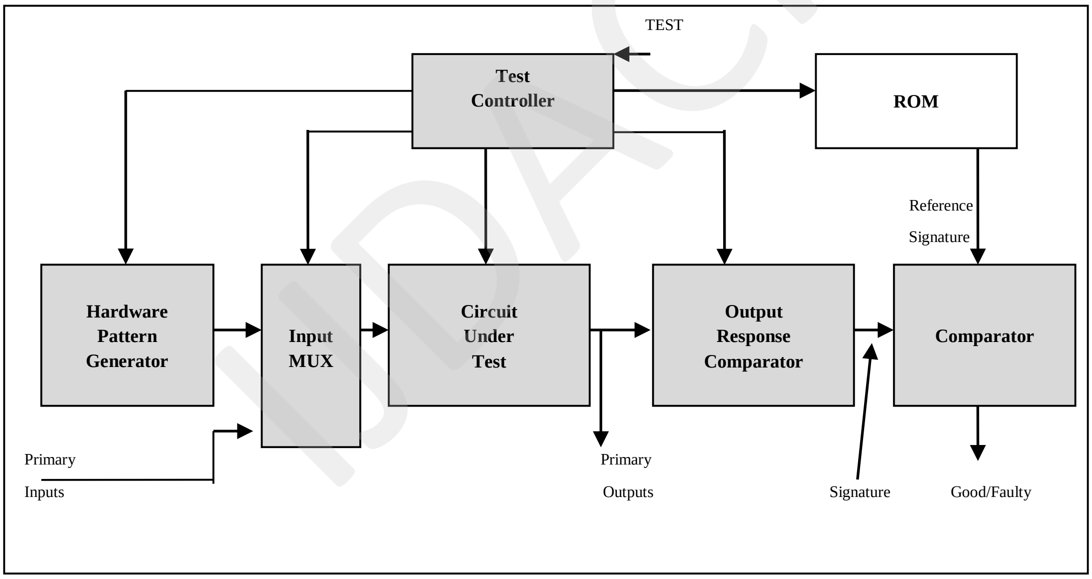

Figure 2: Detailed BIST architecture

III. METHODOLOGY

This section describes the structured design methodology to construct FSM - based programmable memory BIST. The proposed BIST can be programmed on-line, with a "macro command", to select a test algorithm from a memory BIST.

In order to verify whether a given memory cell is good, it is necessary 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. Most commonly used memory test algorithms are March tests, in which there are finite sequences of March elements. 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. The address of the next cell can be in either ascending or descending address order. The notations are summarized in the table 1.

When an algorithm reads a cell response will be either 0 or 1 and they are denoted as r0 and r1 respectively. similarly write 0(1) into a cell is denoted as w1(w0) .we show commonly used test algorithm in table with above notation For example, the MATS+ algorithm first writes a 0 to each cell in

IJDACR
International Journal Of Digital Application & Contemporary Research

International Journal of Digital Application & Contemporary research
Website: www.ijdacr.com (Volume 2, Issue 2, September 2013)

any order ((w0)). In the second March element, it first verifies if the content in a given cell is 0, and then writes a 1 into the same cell. The process is conducted from address 0 up to the last memory cell ((r0, w1)). In the last March element, the algorithm verifies if the content of a cell is 1 and then write 0 back to the cell, for all cells starting from the last one down to address 0 ((r1, w0)).

Table 1: Symbolisation of Operations

r	**A Read Operation**
w	**A Write Operation**
↑	**Up addressing order**
↓	**Down addressing order**
↕	**Any addressing order**

From Table 2, we can see that different test algorithms may have the same March elements, and thus we can design a simple and flexible BIST controller with shared components.

Table 2: Some Memory Test Algorithms

No.	Algorithm	March Elements Code
000	MATS+	{↕(w0); ↑(r0,w1); ↓(r1,w0)}
001	March X	{↕(w0); ↑(r0,w1); ↓(r1,w0); ↕(r0)}
010	March C-	{↕(w0); ↑(r0,w1); ↑(r1,w0); ↓(r0,w1); ↓(r1,w0); ↕(r0)}
011	March A	{↕(w0);↑(r0,w1,w0,w1);↑(r1,w0,w1);↓(r1,w0,w1,w0);↓(r0,w1,w0);}
100	March B	{↕(w0); ↑(r0,w1,r1,w0,r0,w1); ↑(r1,w0,w1);↓(r1,w0,w1,w0); ↓(r0,w1,w0)}
101	March U	{↕(w0); ↑(r0,w1,r1,w0); ↑(r0,w1);↓(r1,w0,r0,w1); ↓(r1,w0)}
110	March LR	{↕(w0); ↓(r0,w1); ↑(r1,w0,r0,w1); ↑(r1,w0);↑(r0,w1,r1,w0); ↑(r0)}
111	March SS	{↕(w0); ↑(r0,r0,w0,r0,w1); ↑(r1,r1,w1,r1,w0); ↓(r0,r0,w0,r0,w1); ↓(r1,r1,w1,r1,w0); ↕(r0)}

Description of Mats-plus algorithm

The first state of this algorithm is "Idle" state, indicating that, there is no any BIST operation is performed Mats plus algorithm will be in "Idle" state unless BIST_En signal is remain equal to "0" .The BIST operation start as soon as BIST_En signal is made equal to "1" then in this algorithm the first operation is "W0" which means that there is write "0" operation is to be performed hence "S0" is the first state in which write "0" operation performed. When "write_complete" signal equal to "1" then FSM entered in new state "S1". In "S1" state there are two elements, one is "r0" and another is "W1".When write one operation is performed then "write_complete" signal become equal to "1" then FSM will switch to "S2" state, where two operation required to be performed; read one and write zero. Thus when the last operation is completed then FSM will switch to idle state.

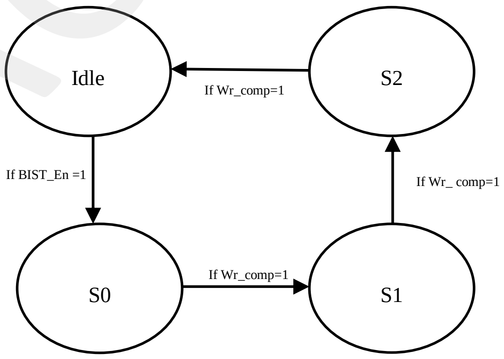

Figure 3: Matsplus algorithm

Description of March-X

The first state of this algorithm is "Idle" state, indicating that, there is no any BIST operation is performed. March-X algorithm will be in "Idle" state unless BIST_En signal is remain equal to "0".The BIST operation starts as soon as BIST_En signal is made equal to "1", then in this algorithm the first operation is "W0" which means that there is write "0" operation is to be performed hence "S0" is the first state in which write "0" operation performed. When "write_complete" signal equal to "1" then FSM entered in new state "S1". In "S1" state there are two elements, when write one operation is performed then "write_complete" signal become equal to "1"& Ora=1and Rd_complete=1 then FSM will switch to "S2" state. In "S2" state also

IJDACR
International Journal Of Digital Application & Contemporary Research

International Journal of Digital Application & Contemporary research
Website: www.ijdacr.com (Volume 2, Issue 2, September 2013)

there are two elements, when write one operation is performed then "write_complete" signal become equal to "1"& Orb=1and Rd_complete=1 then FSM will switch to "S3" state. At S3 state read one operation is performed,t hus when the last operation is completed then FSM will switch to idle state.

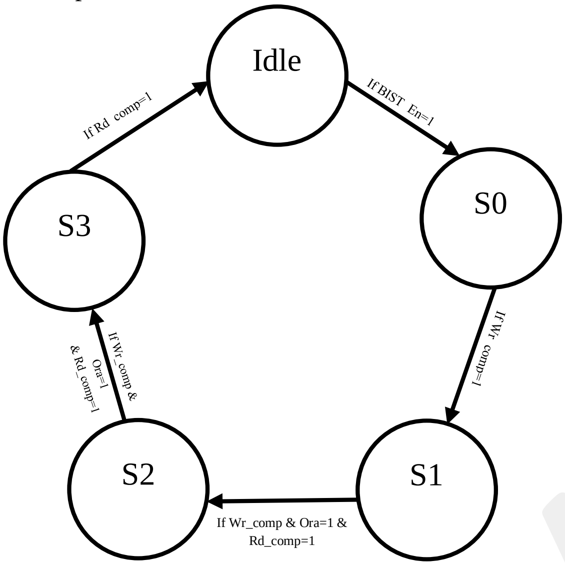

Figure 4: March-X

Description of C Minus algorithm FSM

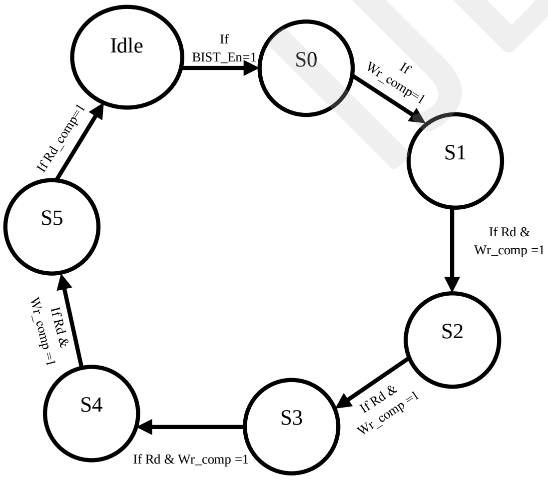

Figure 5: C Minus algorithm

March C Minus algorithm has been shown in figure 5. In which the first state "idle" which shows that this memory BIST is not working. Whenever BIST_En is made Equal to "1" then the first operation of this algorithm is performed in form of "Write 0" operation, this operation is defined by "S0" state when "write_complete" signal is made Equals to "1" then FSM switch to the "S1" state which is having two elements so two operation is needed to be performed, Rd & Wr_complete=1, so as soon as second operation of this state is performed then "Write_complete" signal equals to "1" results in FSM switches to the third state "S2". "S2" state is also having two elements so two operation is needed to be performed, Rd & Wr_complete=1, so as soon as second operation of this state is performed then "Write_complete" signal equals to "1" results in FSM switches to the fourth state "S3". Again "S3" state is having two elements so two operation is needed to be performed, Rd & Wr_complete=1, so as soon as second operation of this state is performed then "Write_complete" signal equals to "1" results in FSM switches to the fifth state "S4". Here "S4" state is having two elements so two operation is needed to be performed, Rd & Wr_complete=1, so as soon as second operation of this state is performed then "Write_complete" signal equals to "1" results in FSM switches to the sixth state "S5". At "S5" state read one operation is performed, then finally idle state is achieved by FSM.

Description of March A

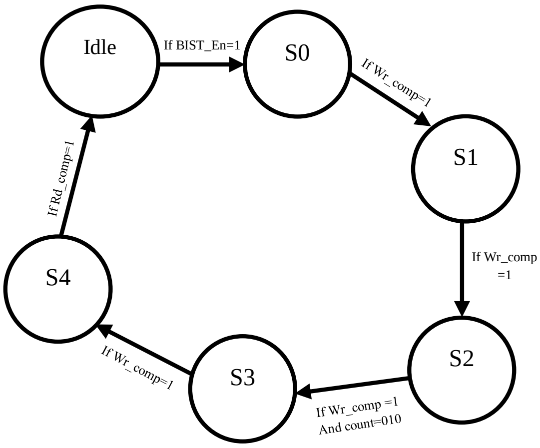

Figure 6: March A

The first state of this algorithm is "Idle" state, indicating that, there is no any BIST operation is performed. March A algorithm will be in "Idle" state unless BIST_En signal is remain equal to "0".The BIST operation starts as soon as BIST_En signal is made equal to "1" then in this algorithm the first operation is "W0" which means that there is write

IJDACR
ISSN: 2319-4863

IJDACR
International Journal Of Digital Application & Contemporary Research

International Journal of Digital Application & Contemporary research
Website: www.ijdacr.com (Volume 2, Issue 2, September 2013)

"0" operation is to be performed hence "S0" is the first state. Then at "S0" state, write "0" operation is performed, when "write_complete" signal equal to "1" then FSM entered in new state "S1". In "S1" state there are two operations are performed. When write one operation is performed then "write_complete" signal become equal to "1" and when count=100 then FSM will switch to "S2" State. In "S2" state also there are two operations are performed. When write one operation is performed then "write_complete" signal become equal to "1" and when one of count signal "010"=1 then FSM will switch to "S3" State. At "S3" state write "0" operation is performed, when "write_complete" signal equal to "1" then FSM entered in new state "S4". At "S4" state read one operation is performed, then FSM will be in idle state.

Description of March-B

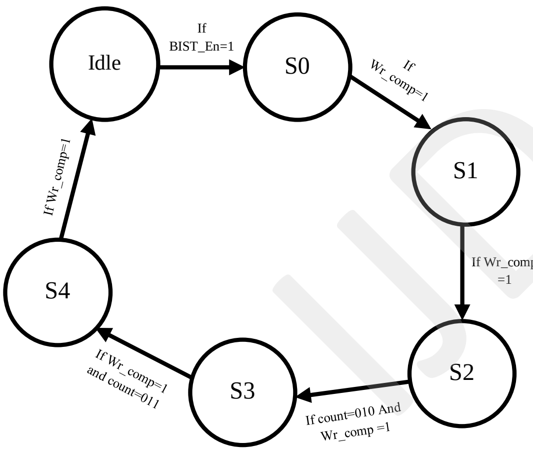

Figure 7: March-B

The first state of this algorithm is "Idle" state, indicating that, there is no any BIST operation is performed. March-B algorithm will be in "Idle" state unless BIST_En signal is remain equal to "0". The BIST operation starts as soon as BIST_En signal is made equal to "1" then in this algorithm the first operation is "W0" which means that there is write "0" operation is to be performed hence "S0" is the first state. Then at "S0" state, write "0" operation is performed, when "write_complete" signal equal to "1" then FSM entered in new state "S1". In "S1" state again write "0" operation is performed, when "write_complete" signal equal to "1" then FSM will switch to "S2" State. The "S3" state can be achieved if one of count signal "010"=1 similarly "S4" state can be accomplished while count signal "011"=1. At "S4" state write operation is performed, when "write_complete" signal equal to "1" then FSM will be in idle state.

Description of March U

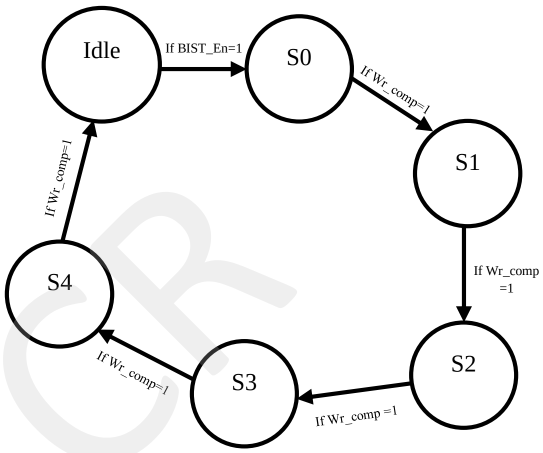

Figure 8: March U

FSM of March U algorithm is shown in figure 8. In which the first state is "idle" state which shows that this memory BIST is not working. Whenever BIST_En is made equal to "1" then the first operation of this algorithm is performed in form of "Write 0" operation. This operation is defined by "S0" state. When "write complete 0" signal is equal to "1" then FSM switch to the "S1" state. This state is having 4 elements, so four operations are needed to be performed. When the first operation of this algorithm is performed, the counter along with this operation is increased by one to define, first of two identical operation is performed (the counter is provided as there are two identical operation consecutively given). So as soon as fifth operation of this state is performed then "Write_complete" signal equal to "1". This results in FSM switches to the third state. Hence all operations of remaining states are performed in same manner and finally idle state is achieved.

Description of March LR

FSM of March LR algorithm is shown in figure 9. In which the first state is "idle" state which shows that this memory BIST is not working. Whenever

IJDACR
ISSN: 2319-4863

IJDACR
International Journal Of Digital Application & Contemporary Research

International Journal of Digital Application & Contemporary research
Website: www.ijdacr.com (Volume 2, Issue 2, September 2013)

BIST_En is made equal to "1" then the first operation of this algorithm is performed in form of "Write 0" operation. This operation is defined by "S0" state. When "write_complete" signal is equal to "1" then FSM switch to the "S1" state. At "S1" state again if "write_complete" signal is equal to "1" then FSM switch to the "S2" state. At "S2" state again if "write_complete" signal is equal to "1" then FSM switch to the "S3" state. At "S3" state again if "write_complete" signal is equal to "1" then FSM switch to the "S4" state. At "S4" state again if "write_complete" signal is equal to "1" then FSM switch to the "S5" state. At "S5" state read one operation is performed, then FSM will be in idle state.

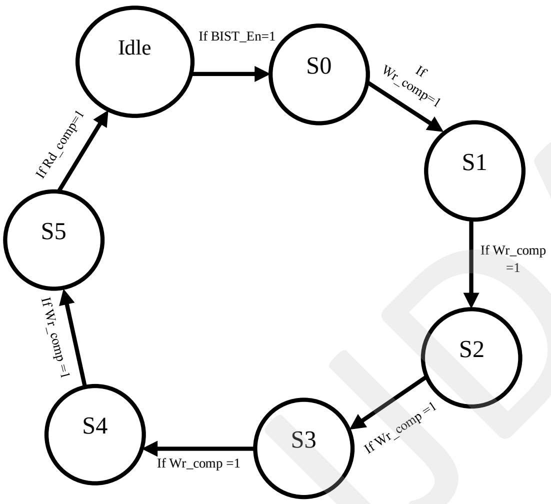

Figure 9: March LR

Description of March SS algorithm

FSM of March SS algorithm is shown in figure 10. In which the first state is "idle" state which shows that this memory BIST is not working. Whenever BIST_En is made equal to "1" then the first operation of this algorithm is performed in form of "Write 0" operation. This operation is defined by "S0" state. When "write_complete" signal is equal to "1" then FSM switch to the "S1" state. At "S1" state again if "write_complete" signal is equal to "1" then FSM switch to the "S2" state. At "S2" state again if "write_complete" signal is equal to "1" then FSM switch to the "S3" state. At "S3" state again if "write_complete" signal is equal to "1" then FSM switch to the "S4" state. At "S4" state again if "write_complete" signal is equal to "1" then FSM switch to the "S5" state. At "S5" state if "write_complete" signal is equal to "1", then FSM will be in idle state.

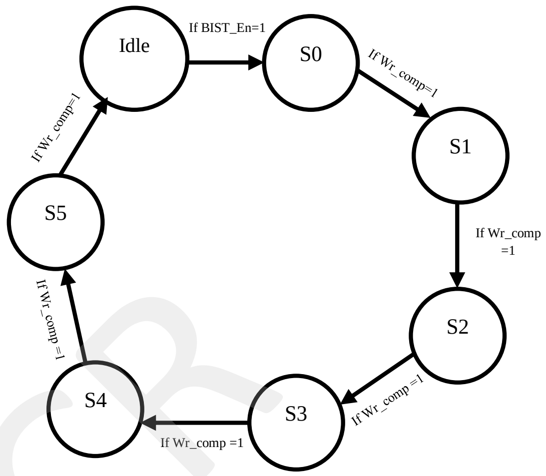

Figure 10: March SS algorithm

IV. SIMULATION & RESULTS

The proposed design is synthesized with Xilinx ISE9.2. To prove the better performance and area utilization, Table 3 has been shown.

Table 3: Result comparison

Algorithm	Gate Count of Previous method	Gate Count of Proposed method
MATS+	730	216
March X	768	241
March C-	762	281
March B	1,038	1,215
March LR	NI	905
March U	NI	885
March SS	NI	651

Here in simulation 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"

International Journal Of Digital Application & Contemporary Research

International Journal of Digital Application & Contemporary research

Website: www.ijdacr.com (Volume 2, Issue 2, September 2013)

needs to be written and "0" all eight location is written for 64 counts after which algorithm will be changed to next state.

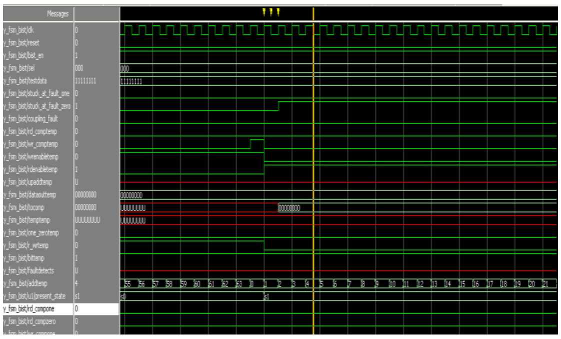

Figure 11: Simulation of faulty FSM-BIST

In this simulation 'sel' line is 000 is given to select the Mats+ algorithm in which S1 state is showing the read 0 operation during which output from RAM is transferred to comparator in test data to comparator is given as "1" hence stuck at 0 fault is being shown in this simulation.

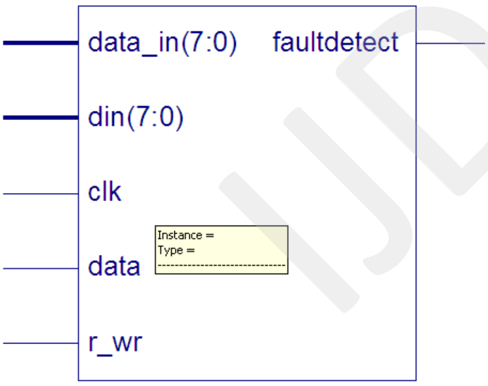

Figure 12: RTL view of fault detect

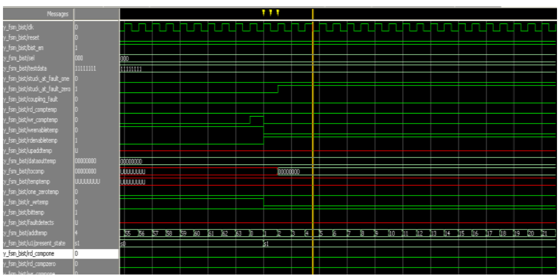

Figure 13: Final implementation

V. Conclusion

Built-in-self-test is implemented by finite state machine. Comparison of two FSM BIST in terms of gate count and power dissipation. In terms of future work, the next thing we can do is to implement the Logic BIST.

Reference

[1] R. K. Sharma AditiSood "Modeling and Simulation of Microcode-based Built-In Self-Test for Multi-Operation Memory Test Algorithms" IJCSI International Journal of Computer Science Issues Published in Volume 7, Issue 3, No 2, pp 36-40, May 2010.

[2] National Programme on Technology Enhanced Learning (NPTEL), Module 11: Built in Self-test (BIST), online available at: http://nptel.iitm.ac.in/courses/106103016/30.

[3] Zainalabedin Navabi Digital System Test and Testable Design: Using HDL Models and Architectures Springer New York Dordrecht Heidelberg London; 1st Edition. (December 20, 2010)

[4] Singh, B.; Khosla, A.; Bindra, S.; "Power Optimization of Linear Feedback Shift Register (LFSR) for Low Power BIST," Advance Computing Conference IACC IEEE International., vol., no., pp.311-314, 6-7 March 2009

[5] Ramesh, K.M.; Sumam, D.S.; , "Comprehensive address generator for digital Signal Processing," Industrial and Information Systems (ICIIS), 2009 International Conference on , vol., no., pp.325-330, 28-31 Dec. 2009

[6] Doshi N. A., Dhobale S. B., and Kakade S. R.LFSR Counter Implementation in CMOS VLSI. Journal of World Academy of Science, Engineering and Technology 48 2008

[7] Ogunti, E.; Frank, M.; Foo, S.;, "Design of a low power binary counter using BISTable storage element," Electronic Design, 2008. ICED 2008. International Conference, vol., no., pp.1-5, 1-3 Dec. 2008

[8] WonGi Hong, JungDai Choi, Hoon Chang, "A Programmable Memory BIST for Embedded Memory", IEEE, International SoC Design Conference (ISOCC), Volume: 02, 2008.

[9] Shilesh Malliyoor, Chao You, "Comparison of hardware implementation and power consumption of low-power multiple outputs linear feedback shift register", Journal of Engineering, Computing and Architecture, ISSN: 1934-7197, Volume 1, Issue 1, 2007.

[10] Veeramachaneni, S.; Avinash, L.; Kirthi, K.M.; Srinivas, M.B., "A novel high-speed binary and gray Incrementer / Decrementer for an address generation unit", IEEE, International Conference on Industrial and Information Systems, August 2007.

[11] Yarmolik, S.V.; Yarmolik, V.N., "Modified Gray and Counter Sequences For Memory Test Address Generation", IEEE, Proceedings of the International Conference on Mixed Design of Integrated Circuits and System, June 2006.

[12] R. Ubar, G. Jervan, H. Kruus, E. Orasson, I. Aleksejev. Optimization of the Store-and-Generate Based Built-In Self-Test. Baltic Electronic Conference (BEC), pp. 199-202, 2006.

[13] Chun-Yi Lee, James Chien-Mo Li Segment Weighted Random BIST (SWR-BIST) Technique For Low

IJDACR International Journal Of Digital Application & Contemporary Research

International Journal of Digital Application & Contemporary research
Website: www.ijdacr.com (Volume 2, Issue 2, September 2013)

Power Testing Bulletin of the College of Engineering, N.T.U., No. 93, February 2005, pp. 71–80

[14] E. Atoofian, S. Hatami, Z. Navabi, M. Alisafaee and A. Afzali-Kusha,"A New Low-Power Scan-Path Architecture," IEEE International Symposium, Vol.5, pp.5278 – 5281, 23-26 May 2005

[15] Mohammad Tehranipoor, MehrdadNourani, Nisar Ahmed, "Low Transition LFSR for BIST-Based Applications," ATS, pp.138-143, 14th Asian Test Symposium (ATS'05), 2005

[16] Po-Chang Tsai, Sying-Jyan Wang and Feng-Ming Chang "FSM-Based Programmable Memory BIST with Macro Command Proceedings of the 2005 IEEE International" Workshop on Memory Technology, Design, and Testing (MTDT'05)

[17] Sato, T. Sakuma, R. Miyamori, D. Fukaset, M. 2004. Performance analysis of wave-pipelined LFSR IEEE International Symposium on Communications and Information Technology, ISCIT 2004. Vol.2 pp.694 – 699

[18] Van de Goor, A.J., Gayadadjiev, G.N., Yarmolik, V.N., Mikitjuk, V.G.: March LR: A Test for Realistic Linked Faults. In: 16th IEEE VLSI ISE 9.2i Software Test Symposium, pp. 272–280 (1996)

[19] Mehta Huzefa, Michael Robert Owens, Irwin Mary Jane. 1996 Some Issues in Gray Code Addressing. Proceedings of the 6th Great Lakes Symposium Great Lakes Symposium on VLSI. pp 178

[20] J. van de Goor and A. Offerman, "Towards a uniform notation for memory tests," in Proc. European Design and Test Conf., pp. 420-427, 1996.

[21] G. Mikitjuk, V.n. Yarmolik, and A.J. van de Goor, "RAM testing algorithms for detecting multiple linked faults," in Proc. European Design and Test Conf., 1996, pp. 435-439.

[22] V. D. Agrawal, C. R. Kime and K. K. Saluja, "A tutorial on built-in selftest. 2. Principles," IEEE Design & Test of Computers, Vol. 10, No. 2, pp. 69-77, Mar 1993.

[23] Grant, D.M.; Denyer, P.B.; , "Address generation for array access based on modulus m counters," Design Automation. EDAC, Proceedings of the European Conference on, vol., no., pp.118-123, 25-28 Feb 1991

[24] Micheletti, G. Todescato, P. Morandi, C. Omiccioli, A. "On the design of easily testable LFSR counters for frequency division", IEEE Proceedings Computers and Digital Techniques Volume: 134, Issue: 6277 – 280, 1987.

[25] Bushnell M. L. and Agrawal V. D.Essentials of Electronic Testing. Kluwer Academic Publishers. Page(s): 529-543, 2000

[26] David A. Hodges. Analysis and Design of Digital Integrated Circuits, Third Edition, Tata McGraw-Hill Publishing Company Limited, 2003.