iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 9 **Issue:** VI **Month of publication:** June 2021

DOI: https://doi.org/10.22214/ijraset.2021.35875

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 9 Issue VI Jun 2021- Available at www.ijraset.com

Finite State Machine based Programmable Memory Built-in Self-Test

B V S Sai Praneeth[1], T Harshith[2], G Akshitha[3], M Suman Chandra[4], K Jeevan Reddy[5]
[1,2,3]*Student, Department of Electronics & Communication Engineering, Sreenidhi Institute of Science and Technology, Hyderabad, Telangana, India*
[4]*Student, Department of Electronics & Communication Engineering, National Institute of Technology, Kurukshetra, Harayana, India*
[5]*Associate Professor, Department of Electronics & Communication Engineering, Sreenidhi Institute of Science and Technology, Hyderabad, Telangana, India*

Abstract: ***We propose a methodology to design a Finite State Machine(FSM)-based Programmable Memory Built-In Self Test (PMBIST) which includes a planned procedure for Memory BIST which has a controller to select a test algorithm from a fixed set of algorithms that are built in the memory BIST. In general, it is not possible to test all the different memory modules present in System-on-Chip (SoC) with a single Test algorithm. Subsequently it is desirable to have a programmable Memory BIST controller which can execute multiple test algorithms. The proposed Memory BIST controller is designed as a FSM (Finite State Machine) written in Verilog HDL and this scheme greatly simplifies the testing process and it achieves a good flexibility with smaller circuit size compared with Individual Testing designs. We have used March test algorithms like MATS+, March X, March C- to build the project.***

Keywords: Finite State Machine(FSM), Programmable Memory Built-In Self Test(PMBIST), System on Chip(SoC), MATS+, March X, March C-

I. INTRODUCTION

According to Moore's law, the number of transistors in IC's doubled for every 2 years. This can be explained by the progression from Small Scale Integration to Very Large Scale Integration devices which are used in the current generation. The increase in the number of transistors drastically reduced the transistor size konwn as feature size of the transistor which led to increased operating frequencies.

As the feature size reduces, it increases the probability of a manufaturing defect in the IC. This results in the production of a faulty transistor which corrupts the entire chip and affects its functionality at required operating frequency. But the defects created during manufacturung process can't be avoided and so some of the IC's are expected to be faulty / Thus the use of testing, especially memory testing holds primary importance to ensure fault free VLSI devices.

The objective of this project is to design a on-chip testing device which is able to detect faults in the memory. Different types of faults can be detected using one or more test algorithms. Idea behind this project is instead of having an individual design for each algorithm, a design can be modeled to merge the testing algorithms and make a single design which will have the test algorithms to detect different faults. By having multiple algorithms in a single design instead of an unique test design for each and every algorithm ,the area of the SoC(System on Chip) for used by each testing design will be merged into a single testing are resulting in a reduced area of the SoC(System on Chip).

II. METHODOLOGY

In this report, we propose a FSM-based BIST engineering that is controlled by BIST controller. To execute a chose test algorithm, we only need to initiate the BIST controller by sending a select signal, and the test vectors will be automatically produced by the proposed Memory BIST. The Memory BIST consists of two FSMs, one of which is used to select a test algorithm and the other controls the individual read/write operation. By using this architecture, we can simplify the test process and reduce test time as well. The algorithms used in the architecture are March algorithms like MATS+, March X, March C-.

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 9 Issue VI Jun 2021- Available at www.ijraset.com

III. PROPOSED SYSTEM

The proposed programmable MBIST design consists of two sections, as demonstrated in Figure1. The Sequencer has a selection table to select algorithms and with the assistance of that it selects March components according to the chosen testing algorithm in desired order. The Test Controller produces Read/Write signal for the memory and furthermore has Address generation logic which gets to the memory in right order. The Comparator compares a read information from the memory and an original information from March component. Accordingly, it determines if the current memory cell is acceptable or faulty. Here we have used a 6-bit memory for the implementation.

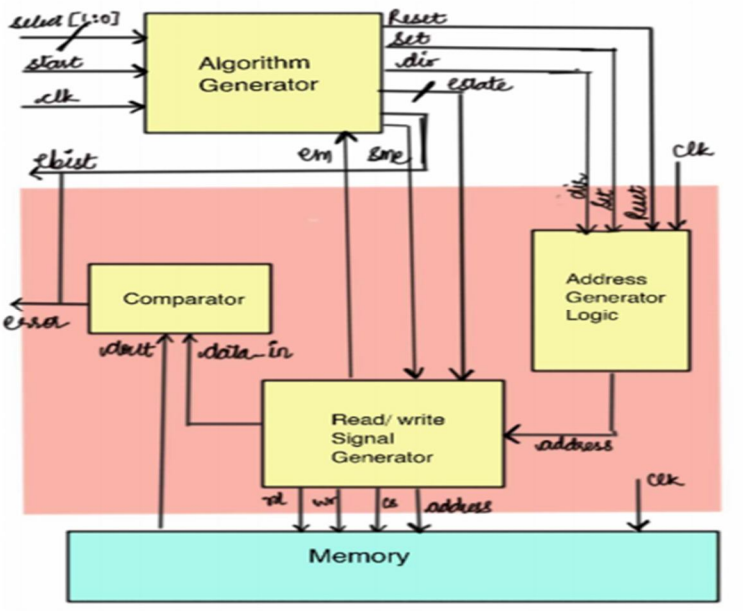


Figure 1: Programmable Memory Buit-In Self Test Architecture

A. Algorithm Generator

A FSM of the Algorithm Generator also known as Sequence Generator is operated by 'START' which is a 1-bit input signal. At the point when we set value of select signal, a 2 bit input signal specifies which testing algorithm is to be configured. Further, it sends a MarchElement code i.e. ESTATE[2:0] to the TestController Unit. At that point the Sequence Generator holds up a signal 'EM' meaning end of MarchElement. The Sequence Generator sends the next MarchElement code after its recieves a signal 'EM' from the Test Controller. In the event that all of MarchElement codes of an algorithm are finished executing in their respective order, FSM raises a signal 'ebist' meaning finish of BIST.

B. Test Controller

Test Controller has three components :

1) *Address Generator Logic:* Address Generator logic generates an address to access memory according to 'Dir' meaning Direction signal . When 'Dir=1' then , it will behave as up counter and for 'Dir=0' it will behave as down counter.
2) *Read/Write Signal Generator:* The Read/Write Signal Generator is a FSM which executes read write operation on the memory. It is initialized by signal 'SME' i.e. start of march element and performs read write operations according to the march element(estate).For example if March element is M1 then according to the below mentioned Table-2 it will perform WriteZero (w0) operation on all the memory location in the respective sequence provided by address generator logic. Further, it will generate the 'EM' signal when all the rd/wr operations on the memory is performed. This 'EM' signal determines the end of March element and then wait for the next march element that is to be executed.
3) *Comparator:* Comparator is a combinational block which compares a read data from the memory and an original data from March element and if the comparison fails it will generate the 'error' signal which will shut the further processing and raise the EBIST signal.

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 9 Issue VI Jun 2021- Available at www.ijraset.com

IV. RESULTS

The top level view of Programmable Memory BIST and its RTL schematic are performed using Xilinx ISE and is shown in Figures 2 and 3.

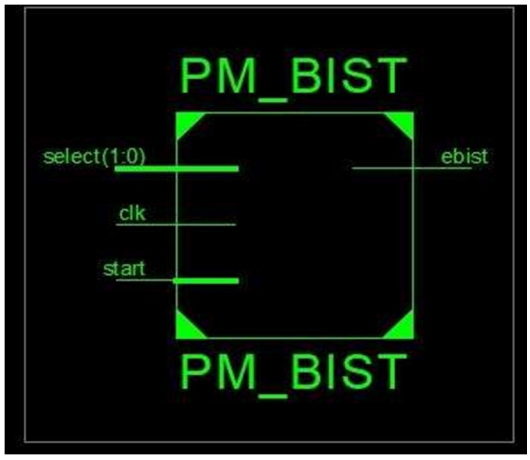


Figure 2: Top level view of PMBIST

Figure 3: RTL Schematic of PMBIST

The simulation outputs are performed using Icarus Verilog as it helps in the analysis of each signal that is generated throughout the code. Now in order to check the working of Programmable Memory BIST let us introduce some faults. Whenever it encounters a fault the 'ebist' signal has to be activated by the FSM and Programmable memory BIST should stop. Let us add Stuck at 0 and Stuck at 1 at address location 10 and their detection is verified in the Figures 4 and 5.

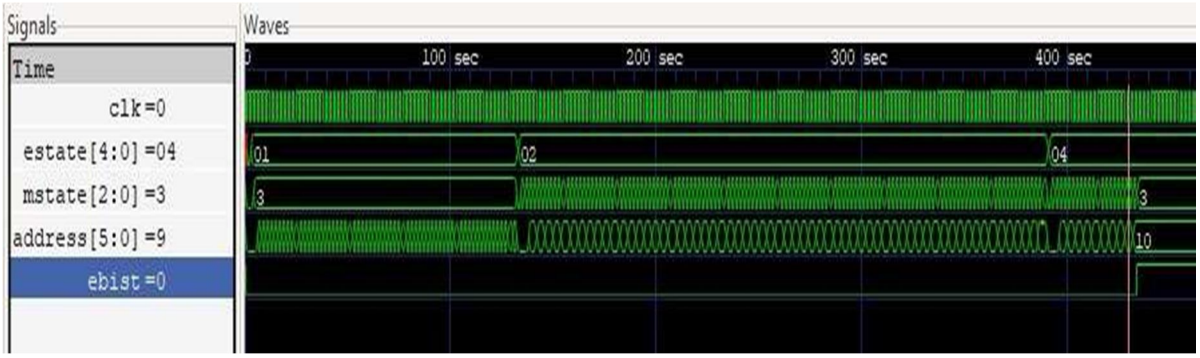


Figure 4: Detection of Stuck at 0 Fault

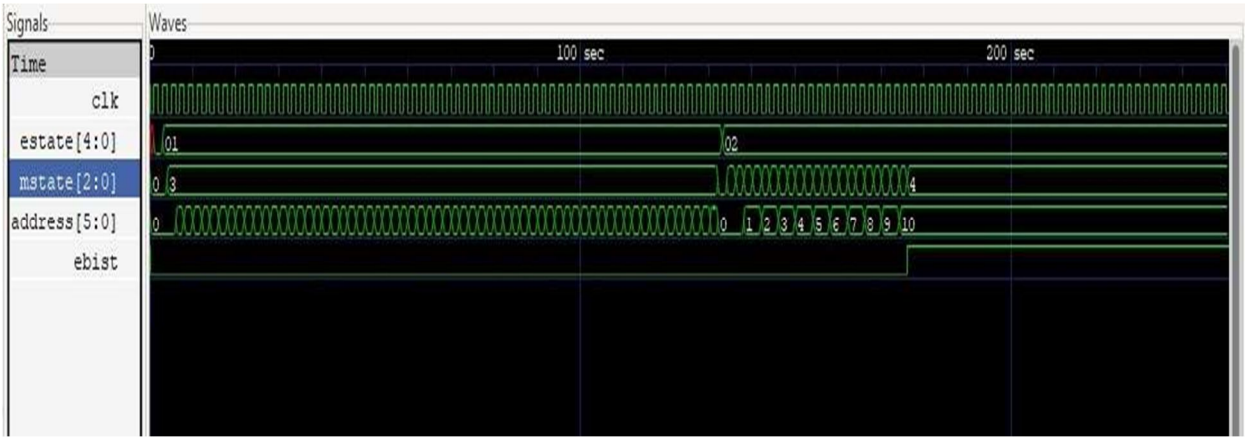


Figure 5: Detection of Stuck at 1 Fault

In Inversion Coupling fault the address location 10 acts as aggressor cell and address location 11 acts as victim cell and its detection using March X algorithm is verified in Figure 6.

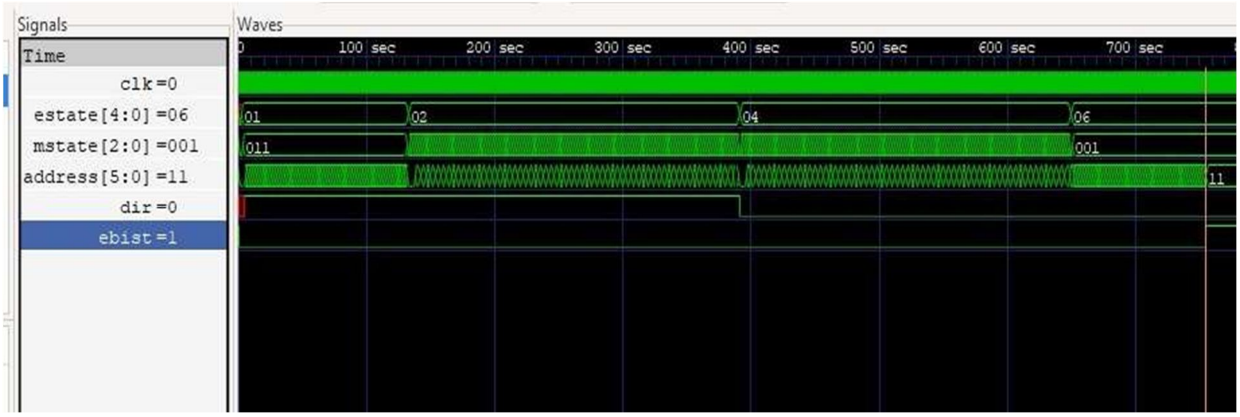


Figure 6: Detection of Falling Inversion Coupling Fault using MARCH X algorithm

The Table 1 summarises all the faults detected by the various algorithms of Programmable Memory BIST\

S.No.	Test Algorithms	Stuck at Faults		Transition Faults		Coupling Faults	
		Stuck at 0	Stuck at 1	Rising Transition Fault	Falling Transition Fault	Inversion Coupling Fault (CF_{in})	Idempotent Coupling Fault (CF_{id})
1	March C-	Yes	Yes	Yes	Yes	Yes	Yes
2	MATS+	Yes	Yes	Yes	No	No	No
3	March X	Yes	Yes	Yes	Yes	Yes	No

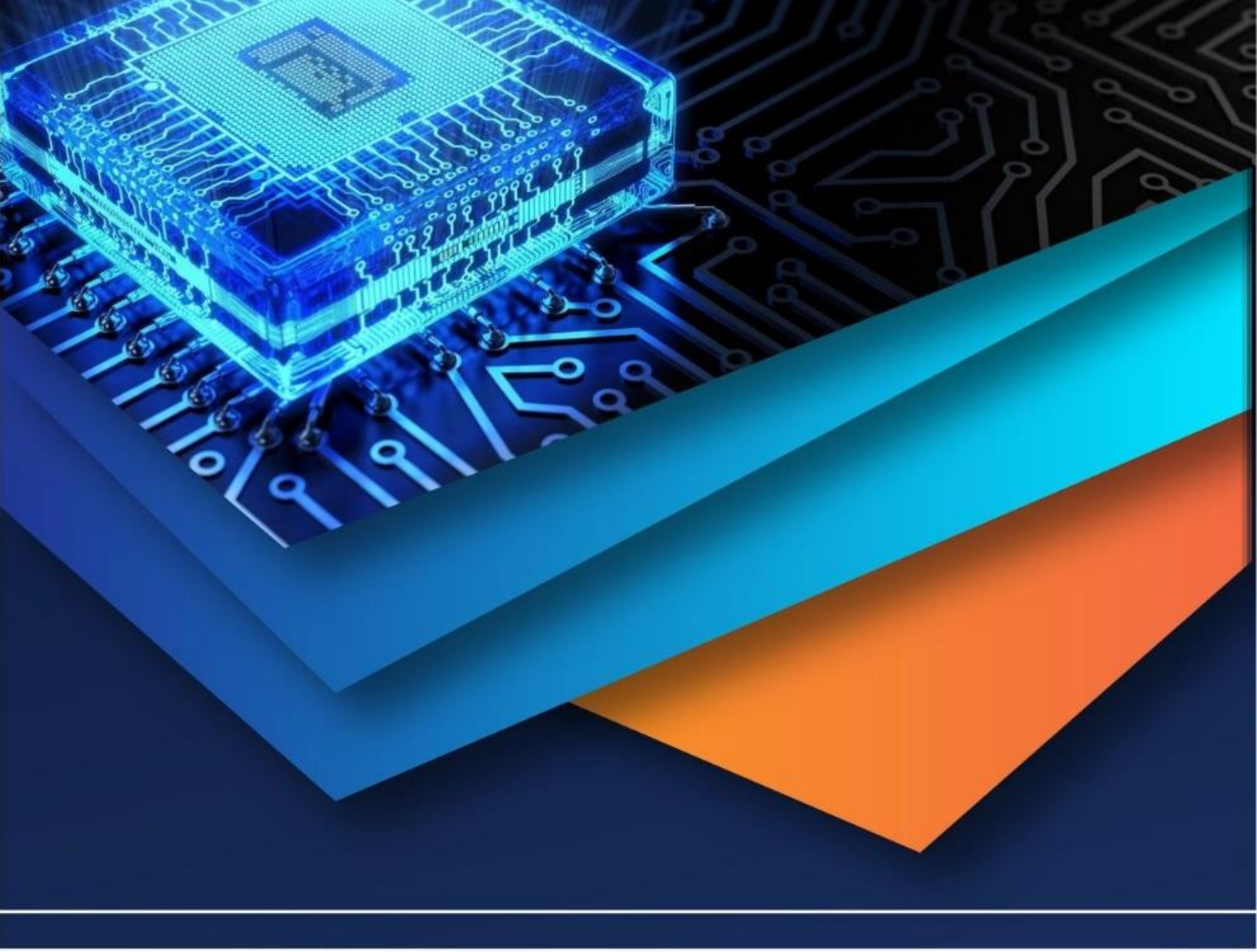
Table 1: Test Algorithms and their fault Coverage

V. CONCLUSION

A FSM based programmable memory BIST controller is put forward in the present study. This plan grants us adaptability in choosing the test algorithms with a select input. In contrast with individual testing plans, our strategy accomplishes the greater flexibility as we can test memory module using suitable testing algorithm at given time. The method which we had put forth will be valuable in SoC testing also, as various memory cores might be utilized in SoC which require distinctive testing algorithms. The presented architecture works on the reconfiguring process if another testing algorithm is chosen, and along with this the overall testing time is decreased. The upsides of the aforementioned configuration includes decreased testing data and testing time, re-programmability and ease in controlling the testing process.

REFERENCES

[1] Laung-Terng Wang, Cheng-Wen Wu and Xiaoqing Wen, “Design for Testability: VLSI Test Principle and Architecture” .
[2] A. J. van de Goor and A. Offerman, “Towards a uniform notation for memory tests,” in Proc. European Design and Test Conf., 1996, pp. 420-427.
[3] V. 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.
[4] Computer Architecture by Morris Mano
[5] WonGi Hong, JungDai Choi, & Hoon Chang. (2008). A programmable memory BIST for embedded memory. 2008 International SoC Design Conference. doi:10.1109/socdc.2008.4815717
[6] Po-Chang Tsai, Sying-Jyan Wang and Feng-Ming Chang department of Computer Science, National Chung-Hsing University Taichung 402, Taiwan, ROC
[7] https://www.edn.com/memory-fault-models-and-testing/
[8] https://www.ncbi.nlm.nih.gov/books/NBK321721/figure/oin_tutorial.F3/
[9] Y MIN, “Introduction”, VLSI Test Principles and Architectures, 2006
[10] www.researchgate.net
[11] Natraj Maneshinde, Prateek Hegade, Rajiv Mittal, Namita Palecha, M S Suna, “Programmable FSM based built-in-self-test for memory”,2016 IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT), 2016
[12] Ze-Wang Chen, Jian-Hua Su, You-Ren Wang, “An Effective Test Algorithm and Diagnostic Implementation for Embedded Static Random Access Memories”, Journal of Circuits, Systems and Computers, 2012.
[13] Joseph, P Elsa, an P Rony Antony, “VLSI design and comparative analysis of memory BIST controllers”, 2014 First International Conference on Computational Systems and Communications(ICCSC), 2014.
[14] Wen-Ben Jone, Der-Chen Huang, S.R. Das, “An efficient BIST method for non-traditional faults of embedded memory arrays”, IEEE Transactions on Instrumentation and Meassurement, 2003



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY