DESIGN OF MODIFIED MARCH-C ALGORITHM AND BUILT-IN SELF-TEST ARCHITECTURE FOR MEMORIES

G. Karthy
B.E., M. Tech., Assistant Professor, ECE,
Kalasalingam Academy of research and education.
Krishnankoil, (India).
E-mail: g.karthy@klu.ac.in ORCID: https://orcid.org/0000-0003-0084-4185

P. Sivakumar
B.E., M.Tech., Ph.D., Professor, ECE,
Kalasalingam Academy of Research and Education.
Krishnankoil, (India).
E-mail: siva@klu.ac.in ORCID: https://orcid.org/0000-0003-1328-8093

Recepción: 05/12/2019 **Aceptación:** 30/12/2019 **Publicación:** 23/03/2020

Citación sugerida:
Karthy, G., y Sivakumar, P. (2020). Design of Modified March-C Algorithm and Built-in self-test architecture for Memories. *3C Tecnología. Glosas de innovación aplicadas a la pyme. Edición Especial, Marzo 2020*, 219-229. http://doi.org/10.17993/3ctecno.2020.specialissue4.219-229

Suggested citation:
Karthy, G., & Sivakumar, P. (2020). Design of Modified March-C Algorithm and Built-in self-test architecture for Memories. *3C Tecnología. Glosas de innovación aplicadas a la pyme. Edición Especial, Marzo 2020*, 219-229. http://doi.org/10.17993/3ctecno.2020.specialissue4.219-229

ABSTRACT

Semiconductor Memories is a pivotal aspect as its technology growth increases. RAM, ROM, DRAM, etc., are the different types of memory and it becomes difficult to test the memory because of the complexity of the design increases day by day. The testing of memory is very difficult as it's required many test patterns. In this paper, a new test architecture is designed using a response analyzer and checker to detect a fault on a chip, and the modified MARCH C algorithm is also proposed to check the fault in the memory in the shortest time.

KEYWORDS

RAM, SOC, Response analyzer, Checker, March algorithm.

 http://doi.org/10.17993/3ctecno.2020.specialissue4.219-229

1. INTRODUCTION

Testing the complete memory is a difficult task. Testing can be done with help of fault models in the Built-in self-test (BIST) architecture. Many fault models are available to test the memory. Here we have used traditional faults models to test the memory. Out of various available algorithms MARACH algorithms provides better fault detection and coverage. In this paper, we proposed a new architecture that consists of the checker, response analyzer, memory unit and a BIST controller with Modified March C algorithm. By using a checker, we can get more précised output.

2. PROBLEM STATEMENT

Memory testing is used to identify that the memory is capable of writing and reading the correct data or not. March based algorithms can identify and locating the fault types which can help to check the design and manufacturing errors. The quality of the test is strongly dependent on the fault model in terms of its fault coverage, its test length as well as the test time required.

In this paper Modified MARCH C- the algorithm is implemented to detects the maximum fault. In addition to that Response analyzer and Checker are included in this architecture to identifies more faults with high precision.

3. TYPES OF FAULT IN THE MEMORY

There are 3 types functional faults models involved in the memory:

1. Memory cell faults.
2. Address Decoder faults.
3. Dynamic faults.

3.1. MEMORY CELL FAULTS

This type of faults forces the contents from 0 to 1 or does not change the contents. Types are SAF- Stuck at fault, SOF-Stuck at open fault, TF – Transition fault, DRF-Data retention

 http://doi.org/10.17993/3ctecno.2020.specialissue4.219-229

fault, CF- Coupling fault, BF-Bridging fault BF, NPSF-Neighborhood Pattern Sensitive Fault, Active (Dynamic) NPSF, Passive NPSF, Static NPSF.

3.2. ADDRESS DECODER FAULTS (AFs)

It occurs in the address, it can be:

Cell not accessed by an address, many cells are accessed by an address, cell accessed by many addresses.

3.3. DYNAMIC FAULTS

1. Recovery faults: Part of the memory cannot recover fast enough from a previous state.
2. Disturb faults*:* victim cell forced to 1 or 0 when we read or write aggressor cell (maybe the same cell).
3. Data Retention faults: Because memory loses its content spontaneously, data cannot be retrieved.

4. MARCH ALGORITHMS

The fixed sequence of read/write operations is carried out to check whether the memory cell is good. The targeted fault model decides the real number of write/read operations and the order of the operations. March tests are the most commonly used memory test algorithms, in which there are fixed sequences of March elements. Then March element is applied to a cell in memory one by one. The operation can be in either descending or ascending address order. The notations of the March algorithm are summarized below:

↑: *address sequence changes in ascending order*

↓: *address 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)

 http://doi.org/10.17993/3ctecno.2020.specialissue4.219-229

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

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

The response will be 0 or 1 if the test algorithm reads a cell, and they are specified as R0 and R1, respectively. Similarly, writing a 1 into a cell is denoted as W1 and writing 0 as W0.

A March- based test algorithm is a fixed sequence of read and write operations called March element. It is specified by a number of reads and write operations and n address order. MATS, MATS+, March-C, March-Y, March-A, and March-B are different types of March- based tests. Because of its simplicity and high fault coverage in most contemporary memory BIST, March based test algorithm is implemented because of its high fault coverage and simplicity. The various March algorithms and Features are stated in the below table:

Table 1. Memory Algorithms and its features.

Sl. No	Type of the March algorithm	Features
1.	MARCHING 1/0 Test	It can detect Auxiliary faults (AF) and Stuck at faults (SAF) and Transition faults (TF).
2.	MATS Test	Modified Algorithmic Test Sequence, it can detect OR type technology.
3.	MATS+ Test	It can detect all Stuck at faults (SAF) and Auxiliary faults (AF).
4.	MATS++	It is like MAT+ additionally it covers transition faults (TF).
5.	MARCH X	It can detect all stuck at faults, auxiliary faults, transition faults and Coupling faults
6.	MARCH C	It can also detect all stuck at fault, auxiliary faults, transition faults and Coupling faults
7.	MARCH C-	Redundancy of MARCH C algorithm is removed.
8.	MARCH A	It can detect AF's, SAF's, linked Coupling Fault CFid's, TF's and certain CFin's linked with CFid's
9.	MARCH Y	Extended version of MARCH X
10.	MARCH B	Extended version of MARCH A

5. BUIILT -IN-SELF-TEST(BIST)

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 for external test equipment, if any. The BIST technique is a common technique to test memories (RAM and ROMs).

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

 http://doi.org/10.17993/3ctecno.2020.specialissue4.219-229

a. On-line BIST: It is implemented on the chip itself. It has area overhead but has the shortest test time.

b. Off-line BIST: It is implemented off the chip itself. It has no area overhead but has the longest test time.

6. TEST ARCHITECTURE

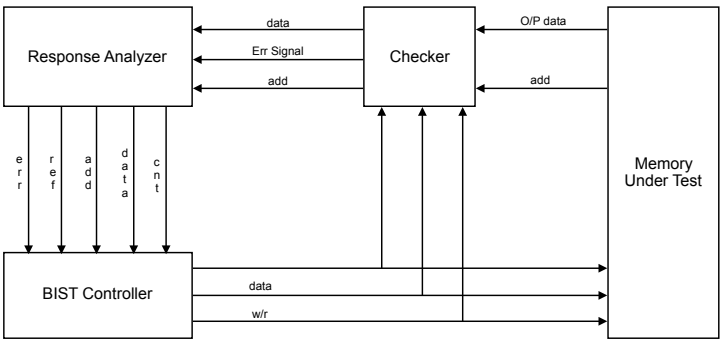

Figure 1. BIST Architecture.

7. EXPLANATION AND RESULTS

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 for external test equipment, if any. The BIST technique is a common technique to test memories (RAM and ROMs).

This architecture consists of BIST Controller, Memory Under Test (MUT), Checker and Response Analyzer. Clock signal becomes to enable the BIST controller to starts working. The BIST controller gives the control signal to the memory. Then the memory undergoes read or/write operation according to the March algorithm.

Then the output from the memory is given to the checker. The checker compares the output from the memory to the data stored inside it. Whenever the fault occurs the checker gives the error signal, the original data along with the address to the response analyzer.

 http://doi.org/10.17993/3ctecno.2020.specialissue4.219-229

The response analyzer is used to switch the controller from normal to repair mode. Whenever the repair mode becomes to enable the controller automatically enables the write signal to repair the fault according to the address and data given by the response analyzer. After the repair operation gets completed the ref signal becomes enable to indicate that fault is repaired. As the continue signal becomes to enable the controller switches to normal mode. This process is continued until the end of the operation. The export mask address signal is used to indicate whether the fault is repairable or not. In this way, this test architecture is used to test and repair the fault in memory with maximum accuracy.

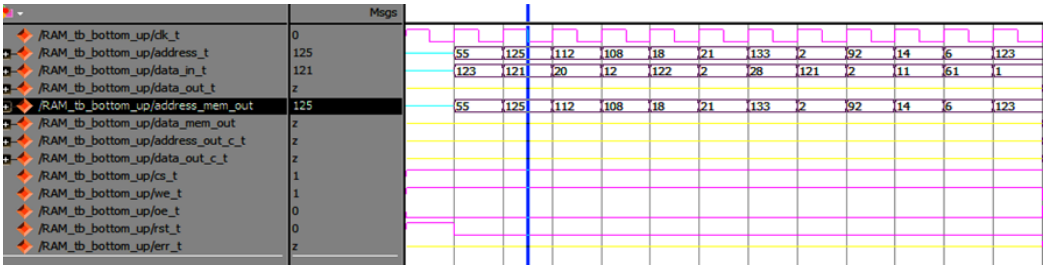

Figure 2. Data writing an operation into the memory and checker.

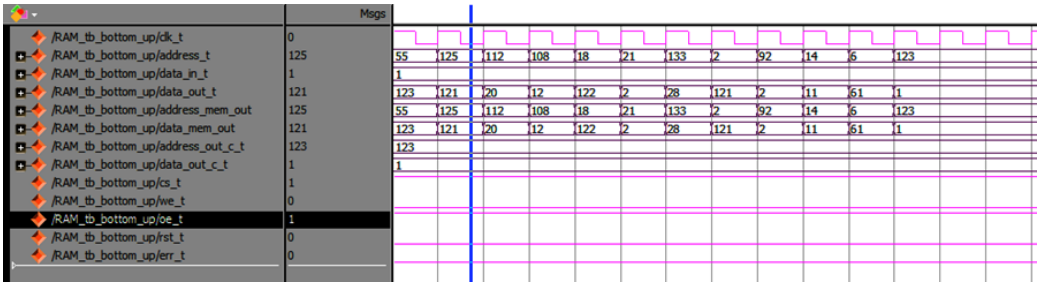

Figure 3. Data reading operation from the memory and it is compared inside the checker for error.

Figure 4. Reading, Writing, and comparison of the data inside the checker for detecting the error.

http://doi.org/10.17993/3ctecno.2020.specialissue4.219-229

8. CONCLUSION

MARCH tests are extensively being used today for functional testing of memory technologies. They are more efficient with better fault coverage than the older classical pattern. In this project modified MARCH C Algorithm with modified BIST architecture are proposed. With this simple BIST architecture and modified MARCH C, testing time can be reduced because two read/write operations carried out in a single clock time. So, Testing Speed can also be doubled. It also provides better fault coverage as MARCH C algorithm covers most of the faults in the memory.

ACKNOWLEDGMENT

We thank the ECE department of Kalasalingam Academy of Research and education, Krishnankoil for supporting this research by providing their center for VLSI lab facility. The Facility which is sponsored by Department of Science and technology (DST) under Fund for Improvement of S&T Infrastructure (FIST) Scheme.

 http://doi.org/10.17993/3ctecno.2020.specialissue4.219-229

REFERENCES

Adams, R. D. (2003). *High Performance Memory Testing: Design Principles, Fault Modeling and Self-Test*. Springer US. https://doi.org/10.1007/b101876

Begum, I., Khayyum, S., & Ravinder, K. (2013). A Built-In Self-Test By Enhanced Faults Coverage with Microcode Optimization for Embedded Memory. *International Journal of Advanced Research in Computer and Communication Engineering, 2*(7). https://ijarcce.com/upload/2013/july/42-h-Khayyum%20Shaik-A%20Built-In%20Self%20Test%20By%20Enhanced%20Faults%20Coverage%20with%20Microcode.pdf

Cockburn, B. F. (1994). Tutorial on Semiconductor Memory Testing. *Journal of Electronic Testing: Theory and Applications, 5*(4), 321-336. https://doi.org/10.1007/BF00972517

Hamdioui, S., Al-Ars, Z., & Van de Goor, A. J. (2002). Testing Static and Dynamic Faults in Random Access Memories. In *Proceedings of 20th IEEE VLSI Test Symposium, pp. 395-400, Monterey, CA, USA*. https://doi.org/10.1109/VTS.2002.1011170

Hamdioui, S., Van de Goor, A. J., & Rodgers, M. (2002). March SS: A Test for All Static Simple RAM Faults. In P*roceedings of 2002 IEEE International Workshop on Memory Technology, Design, and Testing, pp. 95-100, Bendor, France.* https://doi.org/10.1109/MTDT.2002.1029769

Hamdioui, S., Wadsworth, R., De los Reyes, J., & Van de Goor, A. J. (2003). Importance of Dynamic Faults for New SRAM Technologies. In *8th IEEE Proceedings Of European Test Workshop, pp. 29-34, Maastricht, Netherlands, Netherlands.* https://doi.org/10.1109/ETW.2003.1231665

Kapse, V., & Arif, M. (2012). Optimization of Microcode Built-In Self-Test by Enhanced Faults Coverage for Embedded Memory. *IEEE Students' Conference on Electrical, Electronics and Computer Science, Bhopal, India*. https://doi.org/10.1109/SCEECS.2012.6184782

Li, J. F., Cheng, K. L., Huang, C. T., & Wu, C. W. (2001). March-based RAM diagnostic algorithms for stuck-at and coupling faults. *Proceedings International Test Conference 2001 (Cat. No.01CH37260), Baltimore, MD, USA*, pp. 758-767. https://doi.org/10.1109/TEST.2001.966697

Parvathi, M., Vasantha, N., & Prasad, K. S. (2012). Modified March C - Algorithm for Embedded Memory Testing. *International Journal of Electrical and Computer Engineering (IJECE), 2*(5), 571-576. https://pdfs.semanticscholar.org/a697/29ca02295d535faf06141b7c9572bfc48e86.pdf

Sharma R. K., & Sood, A. (2010). Modeling and Simulation of Multi-operation Microcode-Based Built-In Self Test for Memory Fault Detection and Repair. *IEEE Computer Society Annual Symposium on VLSI.* https://doi.org/10.1109/ISVLSI.2010.88

Singh, S., & Chandrawad, U. (2013). Built-In-Self Test for Embedded Memories by Finite State Machine. *International Journal of Digital Application & Contemporary research, 2*(2). http://ijdacr.com/uploads/papers/Swati+singh_paper.pdf

Tanabe, A., Takeshima, T., Koike, H., Aimoto, Y., Takada, M., Ishijima, T., … Okuda, T. (1992). A 30-ns 64 Mb DRAM with built-in self-test and self-repair function. *IEEE Journal of Solid-State Circuits, 27*(11), 1525–1533. https://doi.org/10.1109/4.165332

Tehranipour, M. H., Nourani, M., Fakhraie, S. M., Movahedin, M. R., & Navabi, Z. (2011). *Embedded Test for Processor and Memory Cores in System-on-Chips.* https://wenku.baidu.com/view/260a9369561252d380eb6ee3.html

Van de Goor, A. J. (1998). *Testing Semiconductor Memories, Theory and Practice.* Netherlands: ComTex Publishing. https://doi.org/10.1002/qre.4680080216

Van de Goor, A. J., & Al-Ars, Z. (2000). Functional memory faults: a formal notation and a taxonomy. In *Proceedings of 18th IEEE VLSI Test Symposium, Montreal, Quebec, Canada, Canada.* https://doi.org/10.1109/VTEST.2000.843856

Wang, D., Hu, Y., Li, H. W., & Li, X. W. (2008). The Design- for-Testability Features and Test Implementation of a Giga Hertz General Purpose Microprocessor. *Journal of computer science and technology, 23*(6), 1037-1046. http://jcst.ict.ac.cn/EN/Y2008/V23/I6%20%20%20%20%20%20%20%20%20%20/1037

Zakaria, N. A., Hassan, W. Z. W., Halin, I. A., Sidek, R. M., & Xiaoqing, W. (2012). Fault detection with optimum march test algorithm. *Third International Conference on Intelligent Systems Modelling and Simulation*. https://doi.org/10.1109/ISMS.2012.88

 http://doi.org/10.17993/3ctecno.2020.specialissue4.219-229