Modeling and Simulation of Microcode-based Built-In Self Test for Multi-Operation Memory Test Algorithms

Dr. R.K. Sharma Aditi Sood
Department of Electronics and Communications Engineering
National Institute of Technology, Kurukshetra

Abstract
As embedded memory area on-chip is increasing and memory density is growing, newer test algorithms like March SS are defined to detect newly developing faults. These new March algorithms contain multiple operations per March element. This paper presents a microcoded BIST architecture which can implement these new March tests having number of operations per element according to the growing needs of embedded memory testing. This is shown by implementing March SS Test and testing for new faults including Write Disturb Fault (WDF), Transition Coupling Fault (Cft), Deceptive Read Disturb Coupling Fault (Cfdrd), which established tests like March C- are not capable of detecting. Verilog HDL code of this architecture is written and synthesized using Xilinx ISE 8.2i. Verification of the architecture is done by testing Mentor's ModelSim.

Keywords- *Defect-Per Million (DPM;, Built-In Self Test (BIST); Memory Built-in Self Test (MBIST;, Microcoded MBIST; MUT (Memory Under Test.)*

1. Introduction

According to the 2001 ITRS, today's system on chips (SoCs) are moving from logic dominant chips to memory dominant chips in order to deal with today's and future application requirements. The dominating logic (about 64% in 1999) is changing to dominating memory (approaching 90% by 2011) [1] as shown in Fig.1.

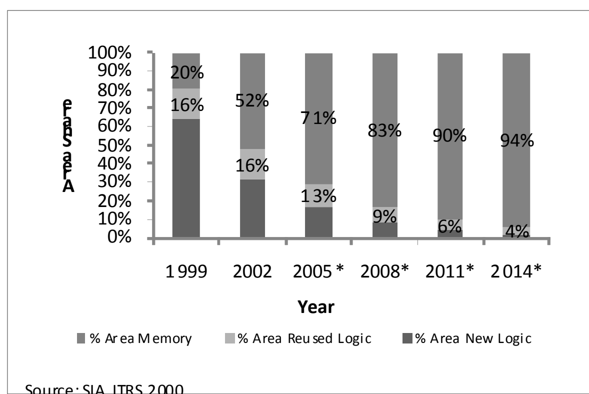

Fig.1 The future of embedded memory

As the memories grow in size and speed, the bit lines, word lines and address decoder pre-select lines will have high parasitic capacitance in addition to a high load. This increases their sensitivity for delay and timing related faults. Also, the significance of the resistive opens is considered to increase in current and future technologies. Since the partial resistive opens behave as delay and time related faults, these faults will become more important in the deep-submicron technologies [2]. Moreover, transistor short channel effect, cross talk effects, impact of process variation have to be necessarily taken into account for developing fault models for embedded memories based on newer technologies.

These factors help in the development of new, optimal, high coverage tests and diagnostic algorithms that allow for dealing with the new defects. The greater the fault detection and localization coverage, the higher the repair efficiency; hence higher the obtained yield.

Thus, the new trends in Memory testing will be driven by the following items:
• Fault modeling: New fault models should be established in order to deal with the new defects introduced by current and future (deep-submicron) technologies.
• Test algorithm design: Optimal test/diagnosis algorithms to guarantee high defect coverage for the new memory technologies and reduce the DPM level.
• BIST: The only solution that allows at-speed testing for embedded memories.

A new microcoded BIST architecture is presented here which is capable of employing new test algorithms like March SS [5] and March RAW [3] that have been developed for coverage of some recently developed static and dynamic fault models.

2. Microcode MBIST Controller

As shown in the previous section, the importance of developing new fault models increases with the new memory technologies.

The well-known fault models, developed before late 1990's could not explain the occurrence of many faults that were detected using experimental results based on DPM screening of a large number of tests applied to a large number of memory chips that were performed at that time, suggesting the existence of additional faults. This implied that new memory technologies involving high density of shrinking devices lead to newer faults and stimulated the introduction of new fault models, based on defect injection

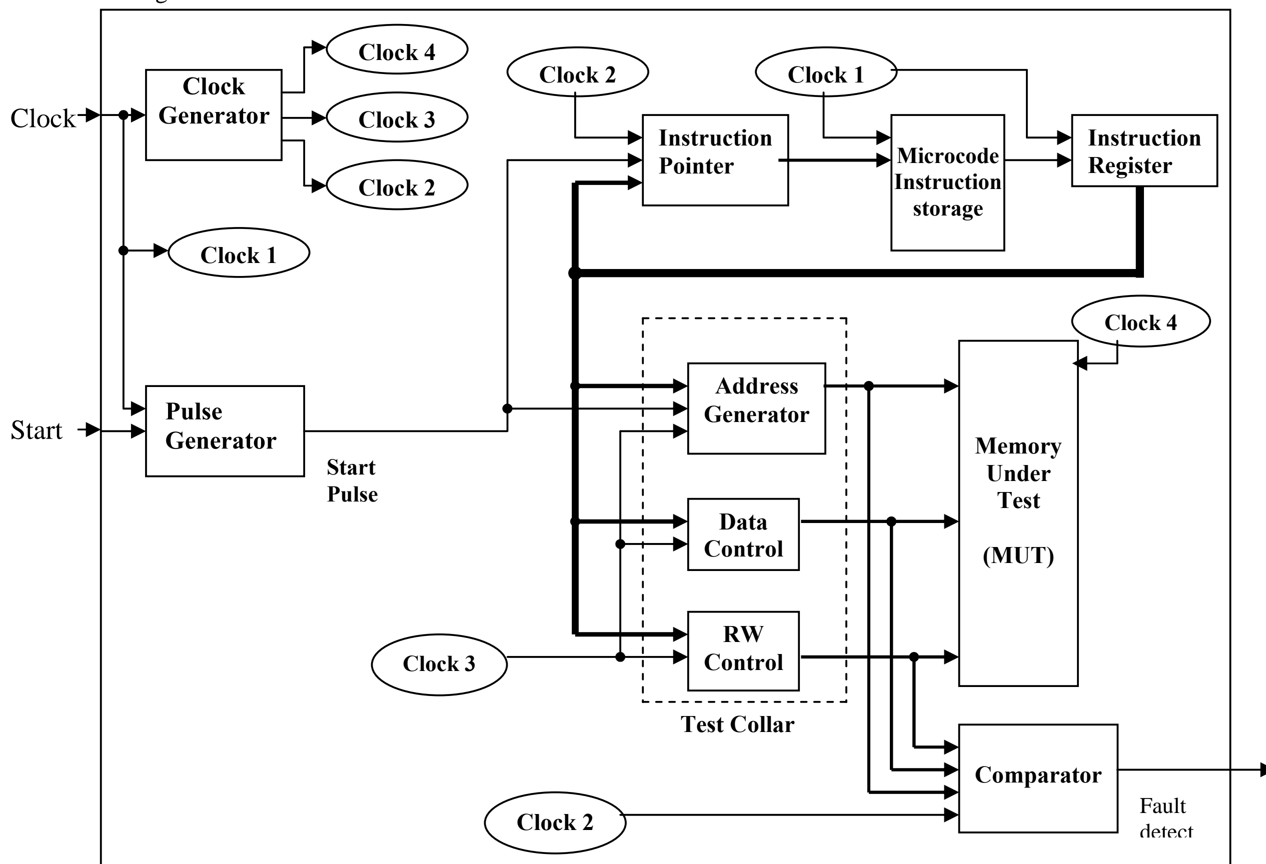

Fig. 2 Microcode MBIST Architecture

and SPICE simulation. Write Disturb Fault (WDF), Transition Coupling Fault (Cft), Deceptive Read Disturb Coupling Fault (Cfdrd) etc. are examples of some such newly defined fault models [2]. Another class of faults called Dynamic faults which require more than one operation to be performed sequentially in time in order to be sensitized have also been defined. [3-4]

Traditional tests, like March C-, are thus becoming insufficient/inadequate for today's and the future high speed memories. Therefore, more appropriate test algorithms have been developed to deal with these new fault models. Examples of such tests are March SS [5] and March RAW [3]. March SS covers some of the new fault models like Deceptive Read Destructive fault, Write disturb fault, etc., whereas March RAW covers some of the Dynamic faults.

These new test algorithms have as many as six or seven operations per march element, and thus some of the recently modeled and simulated architectures are inadequate to implement these test algorithms, as they have been developed to make space for only up to two test operations per March element [6]. This architecture is capable of implementing the newly developed March algorithms, because of its ability to execute algorithms with unlimited number of operations per March element. Thus many of the recently developed March algorithms can be applied using this architecture.

This has been illustrated in the present work by implementing March SS algorithm. However, the same hardware can be used to implement other new March algorithms also by just changing the Instruction storage unit, or the instruction codes and sequence inside the instruction storage unit. The instruction storage unit is used to store predetermined test pattern.

2.1 Methodology

The block diagram of the architecture is shown in Fig 2. The BIST Control Circuitry consists of Clock Generator, Pulse Generator, Instruction Pointer, Microcode Instruction storage unit, Instruction Register. The Test Collar circuitry consists of Address Generator, RW Control, Data Control.
Clock Generator generates simulated clock waveforms Clock2, Clock3, Clock4, for the rest of the circuitry based on the input clock (named Clock1) as shown in Fig. 3

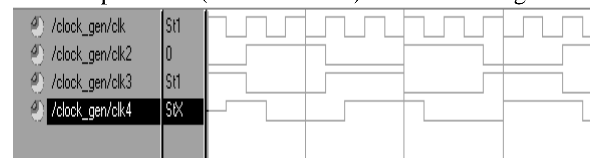

Fig 3. Simulated waveform of Clock generator Module

Pulse Generator generates a 'Start Pulse' at positive edge of the 'Start' signal which marks start of test cycle. **Instruction Pointer** points to the next microword, that is

the next march operation to be applied to the memory under test (MUT). Depending on the test algorithm, it is able to i) point at the same address, ii) point to the next address, or iii) jump back to a previous address.

The flowchart in Fig. 4 precisely describes the functioning of the Instruction Pointer. Here, 'Run complete' indicates that a particular march test operation has marched through the entire address space of MUT in increasing or decreasing order as dictated by the microcode.

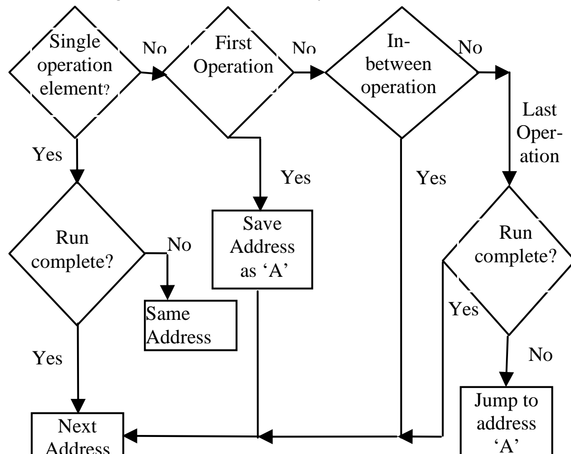

Fig. 4 Flowchart illustrating functional operation of Instruction Pointer

Instruction Register (IR) holds the microword (containing the test operation to be applied) pointed at by the Instruction Pointer. The various relevant bits of microword are sent to other blocks from IR.

Address Generator points to the next memory address in MUT, according to the test pattern sequence. It can address the memory in forwards as well as backwards direction.
RW Control generates read or write control signal for MUT, depending on relevant microword bits..
Data Control generates data to be written to or expected to be read out from the memory location being pointed at by the Address Generator

The Address Generator, RW Control and Data Control together constitute the Memory *Test Collar Comparator* gives the fault waveform which consists of positive pulses whenever the value being read out of the memory does not match the expected value as given by Test Collar.

2.2. Microcode Instruction specification.

The microcode is a binary code that consists of a fixed number of bits, each bit specifying a particular data or operation value. As there is no standard in developing a microcode MBIST instruction [7], the microcode instruction fields can be structured by the designer depending on the test pattern algorithm to be used.
The microcode instruction developed in this work is coded to denote one operation in a single microword. Thus a five operation March element is made up by five micro-code words. The format of 7-bit microcode MBIST instruction word is as shown in Fig. 5. Its various fields are explained as follows: Bit #1 (=1) indicates a valid microcode instruction, otherwise, it indicates the end of test for BIST Controller. Bits #2, #3 and #4 stand for first operation, in-between operation and last operation of a multi-operation March element, interpreted as shown in Fig. 5.
Bit #5 (=1) notifies that the memory under test (MUT) is to be addressed in decreasing order; else it is accessed in increasing order. Bit #6 (=1) indicates that the test pattern data is to be written into the MUT; else, it is retrieved

#1	#2	#3	#4	#5	#6	#7
Valid	**Fo**	**Io**	**Lo**	**I/D**	**R/W**	**Data**

Fo	Io	Lo	Description
0	0	0	A single operation element
1	0	0	First operation of a Multi-operation element
0	1	0	In-between Operation of a Multi-operation element
0	0	1	Last Operation of a Multi-operation element

Fig. 5 Format of Microcode Instruction word

from the memory under test. Bit #7 (=1) signifies that a byte of 1s is to be generated (written to MUT or expected to be read out from the MUT); else byte containing all zeroes are generated.

Table 1 Content of Instruction Storage Unit

	#1 Valid	#2 Fo	#3 Io	#4 Lo	#5 I/D (0/1)	#6 R/W (0/1)	#7 Data (0/1)
M0: χ W0	1	0	0	0	0	1	0
M1: ↑{ R0	1	1	0	0	0	0	0
R0	1	0	1	0	0	0	0
W0	1	0	1	0	0	1	0
R1	1	0	1	0	0	0	0
W1}	1	0	0	1	0	1	1
M2: ↑ {R1	1	1	0	0	0	0	1
R1	1	0	1	0	0	0	1
W1	1	0	1	0	0	1	1
R1	1	0	1	0	0	0	1
W0	1	0	0	1	0	1	0
M3: ↓{R0	1	1	0	0	1	0	0
R0	1	0	1	0	1	0	0
W0	1	0	1	0	1	1	0
R0	1	0	1	0	1	0	0
W1}	1	0	0	1	1	1	1
M4: ↓{ R1	1	1	0	0	1	0	1
R1	1	0	1	0	1	0	1
W1	1	0	1	0	1	1	1
R1	1	0	1	0	1	0	1
W0}	1	0	0	1	1	1	0
M5: χ R0	1	0	0	0	1	0	0
	0	X	X	X	X	X	X

The instruction word is so designed so as to represent any March algorithm. The contents of Instruction storage unit for March SS algorithm are shown in Table 1.

The first march element M0 is a single operation element, which writes zero to all memory cells in any order. Similarly, the second march element M1 is a multi-operation element, which consists of five operations: i) R0, ii) R0, iii) W0, iv) R1 and v) W1. MUT is addressed in increasing order as each of these five operations is performed on each memory location before moving on to the next location.

2.3 Behavior Simulation

Mentor's ModelSim has been used to verify the functionality and timing constraints of BIST module.
Verilog HDL code of the above architecture written and synthesized using Xilinx ISE 8.2i [8].

3. RESULTS

The simulation waveform of a fault-free SRAM is shown in Fig. 6. Of the generated Clock signals, only Clock2 and Clock4 appear in the top module

The top module shows the interfacing of BIST Controller (including test collar), MUT and Comparator. As the START signal goes high, indicating the start of test, the first March element M0 of March SS algorithm is executed. As this is a write signal, no values are read out from the memory to be compared with expected or correct values and hence the output FAULT waveform of comparator is high impedance. As read operation starts at the beginning of execution of M1 element, the values from MUT are read out and compared with the expected values. The FAULT waveform shows a 'low' level throughout the test for a fault-free SRAM

The SRAM model is also amended to be in defective state by inserting faults. The simulated waveform is shown in Fig. 7.

The inserted faults are Deceptive Read Disturb fault (DRDF) at location 11, Write Disturb Fault (WDF) at location 13, Deceptive Read Disturb Coupling fault (CFdrd) at location 9 (victim) due to location 10 (aggressor), Write Disturb Coupling Fault (CFwd) at location 14 (victim) due to location 15 (aggressor) [9].
The fault detect waveform shows 12 pulses due to the above faults in given four locations, as the test elements march through MUT to uncover these defects.
The above stated faults cannot be detected by March C-algorithm but are easily detected by March SS Algorithm which can be implemented by the architecture presented in this work.

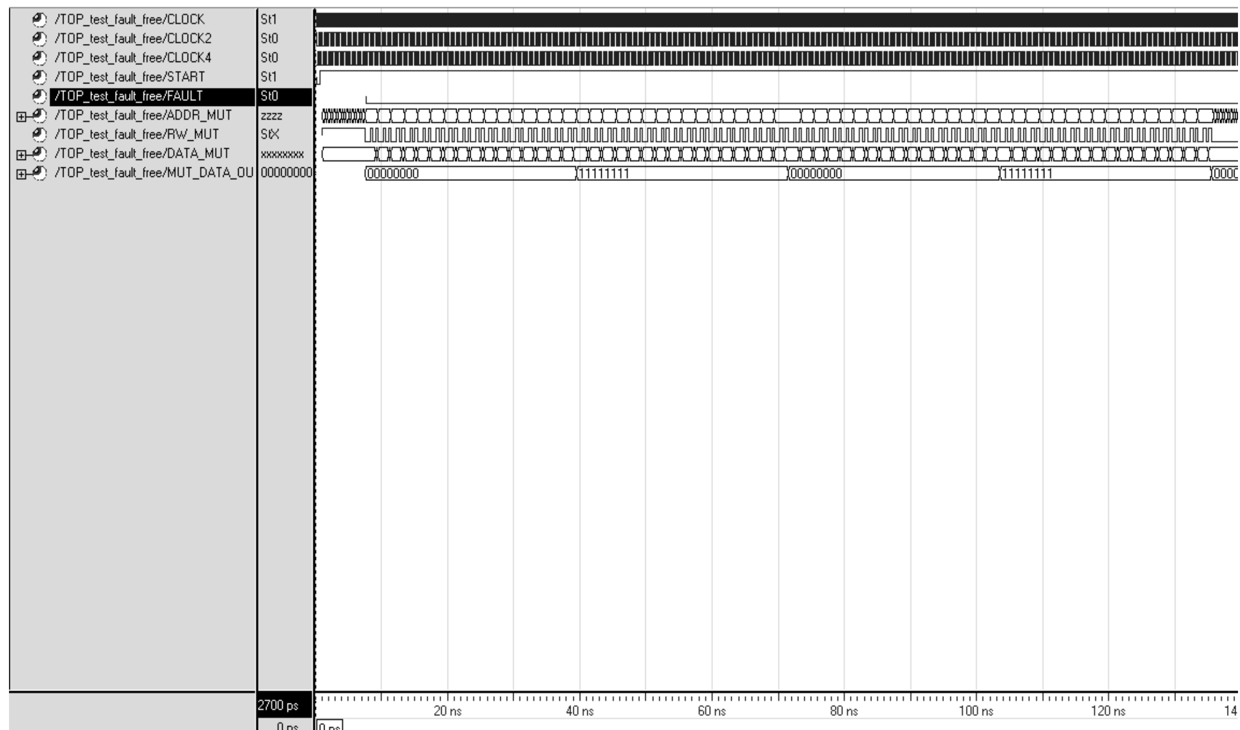

Fig. 6 Simulated waveform of fault-free SRAM

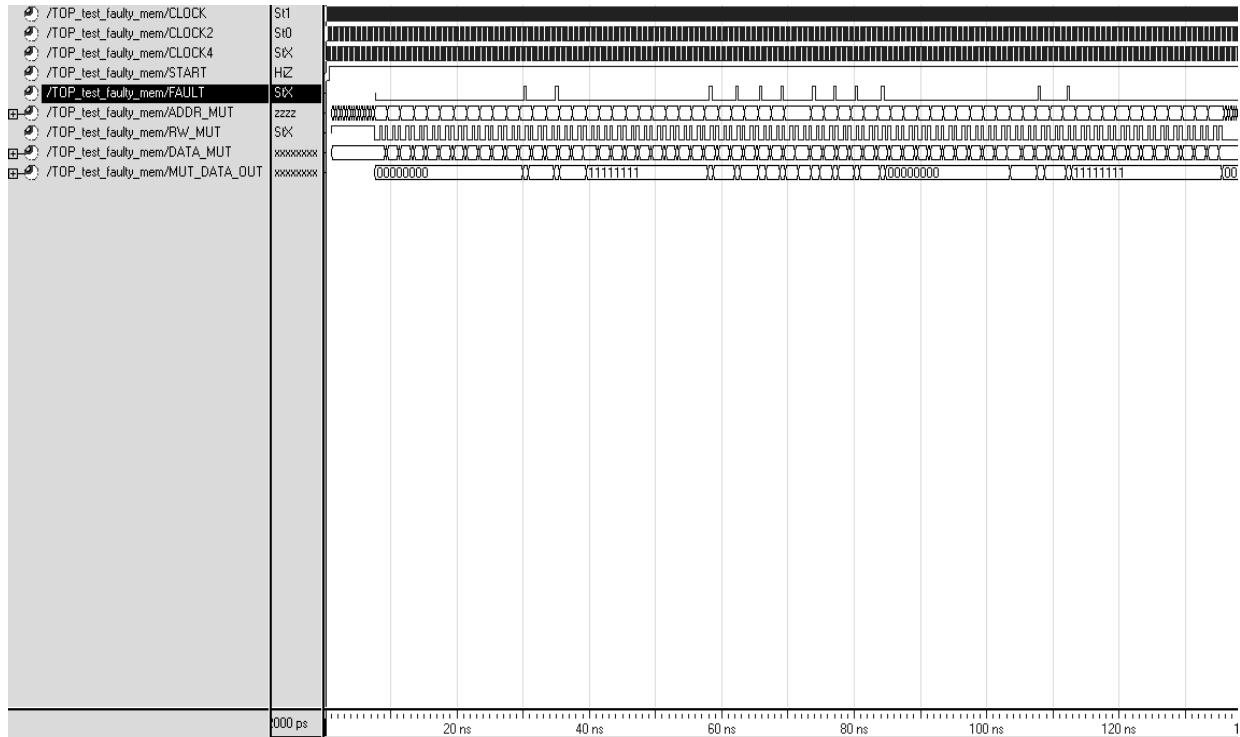

Fig. 7 Simulated waveform of faulty SRAM

4. Conclusion

The simulation results have shown that the micro-coded MBIST architecture described here is an effective testing method to test embedded memories as it provides a flexible approach and better fault coverage. Just as March SS, any new march algorithm can be implemented using the same BIST hardware by changing the instructions in the microcode storage unit, without the need to redesign the entire circuitry.

References

[1] International SEMATECH, “International Technology Roadmap for Semiconductors (ITRS): Edition 2001”

[2] S. Hamdioui, G.N. Gaydadjiev, A.J .van de Goor, “State-of-art and Future Trends in Testing Embedded Memories”, *International Workshop on Memory Technology, Design and Testing (MTDT’04),* 2004.

[3] S. Hamdioui, Z. Al-Ars, A.J. van de Goor, “Testing Static and Dynamic Faults in Random Access Memories”, *In Proc. of IEEE VLSI Test Symposium*, pp. 395-400, 2002.

[4] S. Hamdioui, et. al, “Importance of Dynamic Faults for New SRAM Technologies”, *In IEEE Proc. Of European Test Workshop*, pp. 29-34, 2003.

[5] S. Hamdioui, A.J. van de Goor and M. Rodgers, ”March SS: A Test for All Static Simple RAM Faults”, *In Proc. of IEEE International Workshop on Memory Technology, Design, and Testing*, pp. 95-100, Bendor, France, 2002.

[6] N. Z. Haron, S.A.M. Junos, A.S.A. Aziz, “Modelling and Simulation of Microcode Built-In Self test Architecture for Embedded Memories”, *In Proc. of IEEE International Symposium on Communications and Information Technologies* pp. 136-139, 2007.

[7] R. Dean Adams, “High Performance Memory Testing: Design Principles, Fault Modeling and Self-Test”, Springer US, 2003.

[8] “Xilinx ISE 6 Software Manuals and help – PDF Collection”,http://toolbox.xilinx.com/docsan/xilinx7/books/ manuals .pdf

[9] A.J. van de Goor and Z. Al-Ars, “Functional Fault Models: A Formal Notation and Taxonomy”, *In Proc. of IEEE VLSI Test Symposium*, pp. 281-289, 2000.

[10] Zarrineh, K. and Upadhyaya, S.J., “On Programmable memory built-in self test architectures,” *Design, Automation and Test in Europe Conference and Exhibition 1999*. Proceedings , 1999, pp. 708 -713

[11] Sungju Park et al, “Microcode-Based Memory BIST Implementing Modified March Algorithms”, *Journal of the Korean Physical Society,* Vol. 40, No. 4, April 2002, pp. 749-753

[12] A.J. van de Goor, “Using March tests to test SRAMs”, *Design & Test of Computers, IEEE,* Volume: 10, Issue: 1, March 1993 Pages: 8-14.

[13] R. Dekker, F. Beenker and L. Thijssen, “*Fault Modeling and Test Algorithm Development for Static Random Access Memories*”,

[14] R.Dekker, F. Beenker, L. Thijssen. “A realistic fault model and test algorithm for static random access memories”. *IEEE Transactions on CAD*, Vol. 9(6), pp 567-572, June 1990.

[15] B. F. Cockburn: “Tutorial on Semiconductor Memory Testing” *Journal of Electronic Testing: Theory and Applications,* 5, pp 321-336 1994 Kluwer Academic Publishers, Boston.

[16] A.J. van de Goor, “Testing Semiconductor Memories, Theory and Practice” ComTex Publishing, Gouda, Netherlands, 1998.