ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 2, Issue 7, July 2013

A Built-In Self Test By Enhanced Faults Coverage with Microcode Optimization for Embedded Memory

Imthiazunnisa Begum M.Tech[1], Shaik Khayyum[2], Korani Ravinder M.Tech,[Ph.D][3]

H.O.D, ECE Department, VIFCET, JNTU Hyderabad[1]

M.Tech Student, ECE Department VIFCET, JNTU Hyderabad[2]

Asst. Professor, ECE Department, VIFCET, JNTU Hyderabad[3]

Abstract: System-on-chip (SoC) designs are moving from logic dominant chips to memory dominant chips to be able to meet future application requirements. Embedded memory density and area on-chip is increasing day by day. In order to achieve good memory yield, an at-speed test technique such as built-in self test (BIST) must be implemented to test these embedded memories. Memory Built-in Self Test (MBIST) is the popular approach to test embedded memories. MBIST usually use the deterministic pattern such as MARCH test algorithm to test memories. In MARCH test algorithm, the patterns are generated according to specified predetermined values. The existing March algorithms consist of as many as four or seven operations per March element. Therefore, it is essential to define new test algorithms which fulfill the need of detecting new faults. A new March BLC tests having number of operations per element according to the today's growing needs of embedded memory testing with enhanced fault using Verilog HDL as a primary language and used Modelsim SE 6.5 f as simulation tool.

Keywords: Memory Built-In Self Test (MBIST), Embedded memory fault, Hardware Descriptive Language (HDL)

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) [6] as shown in Fig.1.

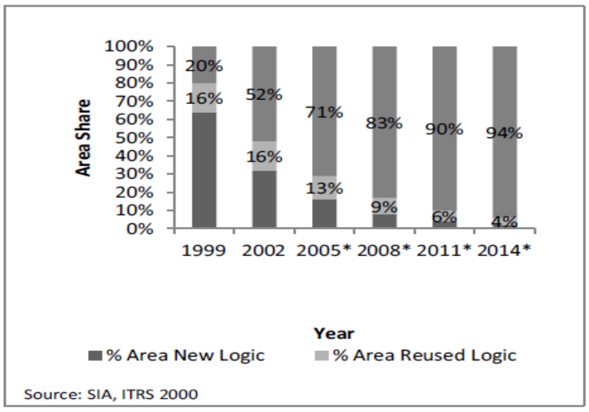

Fig.1 The future of embedded memory

Also, as the memories grow in size and speed, the signal lines, that is 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.

Moreover, 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 [1].

The following considerations for fault modeling for new technologies also have to be taken into account, for example:
• Transistor Short channel effect: lowering the threshold voltage may make the drain leakage contribution significant.
• Cross talk effect and noise from power lines.
• The impact of process variation

The above cited newer defects are a source of new fault models. The development of new, optimal, high coverage tests and diagnostic algorithms 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.

ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 2, Issue 7, July 2013

• BIST: The only solution that allows at-speed testing for embedded memories.

2. MICROCODE MBIST CONTROLLER

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

In addition, the shrinking technology will be a source of previously unknown defects/faults [1]. In the late 1990's, experimental results based on DPM screening of a large number of tests applied to a large number of memory chips indicated that many detected faults could not be explained with the well known fault models, suggesting the existence of additional faults. This stimulated the introduction of new fault models, based on defect injection and SPICE simulation: Read Destructive Fault (RDF), Write Disturb Fault (WDF), Transition Coupling Fault (Cft), Deceptive Read Disturb Coupling Fault (Cfdrd) etc. [1] 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. [4-5]

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 BLC

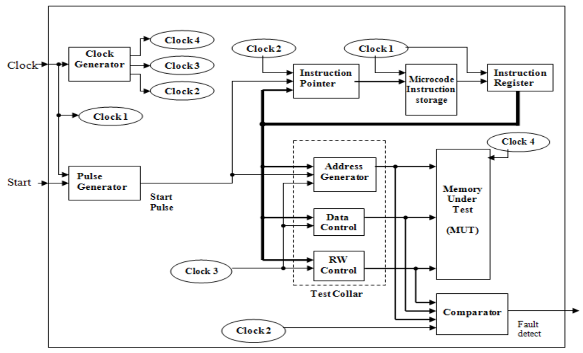

Fig. 2 Microcode MBIST Architecture

[2] and March RAW [4]. March BLC 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 [3]. 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.

In this paper we present March BLC [2], an optimized test that detects all static faults in the presence of BL coupling using only the required CBs, with a test time complexity of 46n. Compared to March m-MSS (108n) [16], which applies all possible CBs, the test time is significantly reduced by over 50%.

$$
\begin{array}{ll}
\text{March BLC} = \{\Updownarrow (w0); & \text{ME0} \\
\Uparrow (r0, r0, w0, r0, w1, w1, r1); & \text{ME1} \\
\Uparrow (r1, r1, w1, r1, w0, w1); & \text{ME2} \\
\Uparrow (r1, r1, w0, w0, r0); & \text{ME3} \\
\Uparrow (r0, r0, w0, r0, w1, w1, w0); & \text{ME4} \\
\Downarrow (r0, r0, w0, w1, w1, r1); & \text{ME5} \\
\Downarrow (r1, r1, w0, w1); & \text{ME6} \\
\Downarrow (r1, r1, w0, w0, r0); & \text{ME7} \\
\Downarrow (r0, r0, w1, w1, w0)\} & \text{ME8}
\end{array}
$$

This has been illustrated in the present work by implementing March BLC 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.

A) 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

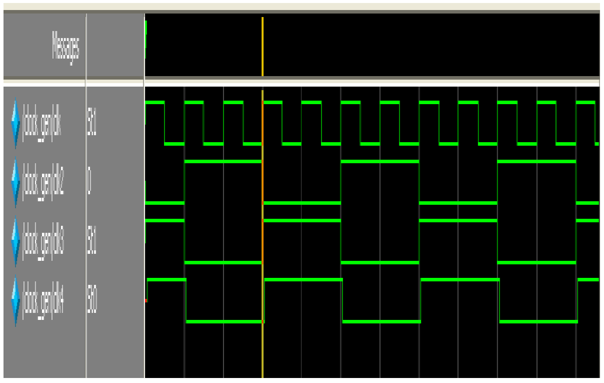

Fig 3. Simulated waveform of Clock generator Module

Pulse Generator generates a 'Start Pulse' at positive edge of the 'Start' signal which marks start of test cycle.

ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 2, Issue 7, July 2013

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.

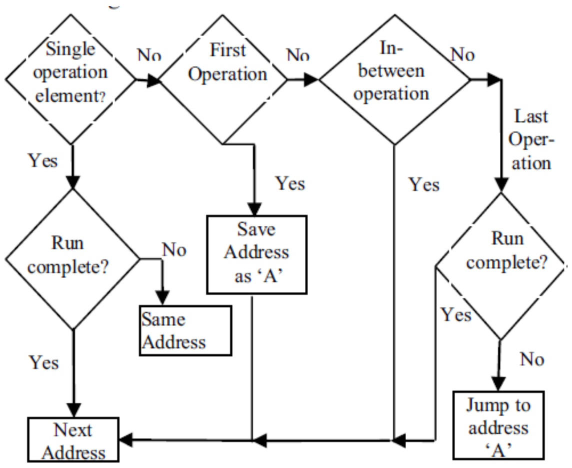

Fig. 4 Flowchart illustrating functional operation of Instruction Pointer

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[17].

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

Address Generator points to the next memory address to be applied the test algorithm according to the test pattern sequence. It can address the memory in either increasing or decreasing order.

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.

B) 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. There is no standard in developing a microcode MBIST instruction. 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. Table 1 shows the 7-bit microcode MBIST Instruction word and description of its various fields 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. A detailed description of how these three bits are interpreted is given in Table 2.

Table 1

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

Table 2

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

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 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 eight bits of all zeroes are generated.

The instruction word is so designed so as to represent any March algorithm. The contents of Instruction storage unit for March BLC algorithm are shown I Table 3. 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.

ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 2, Issue 7, July 2013

Table 3

	#1 Valid	#2 Fo	#3 Io	#4 Lo	#5 I/D(0/1)	#6 R/W (0/1)	#7 Data (0/1)
ME0: χ W0	1	0	0	0	0	1	0
ME1: ↑{ R0	1	1	0	0	0	0	0
R0	1	0	1	0	0	0	0
W0	1	0	1	0	0	1	0
R0	1	0	1	0	0	0	0
W1	1	0	1	0	0	1	1
W1	1	0	1	0	0	1	1
R1}	1	0	0	1	0	0	0
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	1	0	0	1	0
W1}	1	0	0	1	0	1	1
ME3: ↑{R1	1	1	0	0	0	0	1
R1	1	0	1	0	0	0	1
W0	1	0	1	0	0	1	0
W0	1	0	1	0	0	1	0
R0}	1	0	0	1	0	0	0
ME4: ↑{R0	1	1	0	0	0	0	0
R0	1	1	0	0	0	0	0
W0	1	0	1	0	0	1	0
R0	1	0	1	0	0	0	0
W1	1	0	1	0	0	1	1
W1	1	0	1	0	0	1	1
W0}	1	0	0	1	0	1	0
ME5: ↓{ 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	1	0	1	1	1
R1}	1	0	0	1	1	0	1
ME6: ↓{ R1	1	0	0	0	1	0	0
R1	1	0	1	0	1	0	1
W0	1	0	1	0	1	1	0
W1}	1	0	0	1	1	1	1
ME7: ↓{ R1	1	1	0	0	1	0	1
R1	1	0	1	0	1	0	1
W0	1	0	1	0	1	1	0
W0	1	0	1	0	1	1	0
R0}	1	0	0	1	1	0	0
ME8: ↓{ R0	1	1	0	0	1	0	0
R0	1	0	1	0	1	0	0
W1	1	0	1	0	1	1	1
W1	1	0	1	0	1	1	1
W0}	1	0	0	1	1	1	0
	0	X	X	X	X	X	X

C) Behavior Simulation

There are many simulator tools available to execute the simulation step prior to the actual construction. This project has used ModelSim 6.5f to simulate the. Verilog HDL code of the above architecture written and synthesized using Xilinx ISE 9.2i [9].

3. RESULTS

The simulation waveform of a fault-free SRAM is shown in Figure 5. Of the generated Clock signals, only Clock2 and Clock4 appear in the top module.

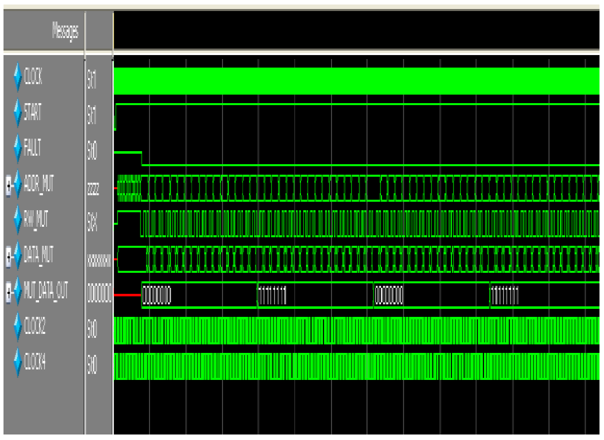

Figure 5. Simulated waveform of Fault-free SRAM

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 BLC algorithm is executed. As this is a write signal, no values are read out from the memory to be compared with expected or correct

ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 2, Issue 7, July 2013

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 Figure 6.

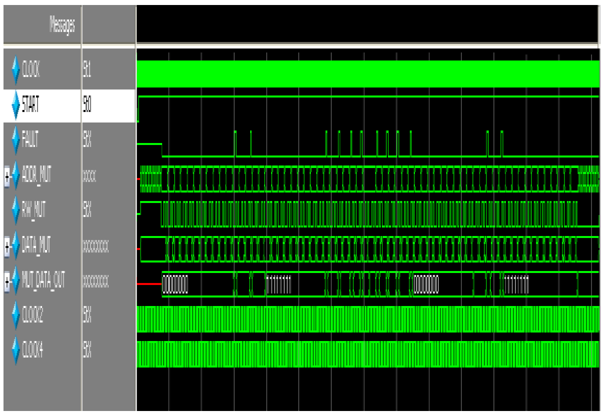

Figure 6. Simulated waveform of Faulty SRAM

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). The fault detect waveform shows 12 pulses due to the above faults in four locations, as the test elements march through MUT to uncover these defects.The above stated faults are the faults which cannot be detected by March C- algorithm but are detected by March BLC Algorithm implemented here.

4. CONCLUSION

The above waveforms have shown that the micro-code MBIST architecture above is an effective testing method to test embedded memories as it provides a flexible approach and better fault coverage. Just as March BLC ,any new march algorithm can be implemented using the same BIST hardware by replacing the microcode storage unit, without the need to redesign the entire circuitry.

REFERENCES

[1] 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.
[2] Sandra Irobi Zaid Al-Ars Said Hamdioui.Memory Test Optimization for Parasitic Bit LineCoupling in SRAMs. IEEE International Test Conference, 2010.
[3] 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.
[4] 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.
[5] S. Hamdioui, et. al, "Importance of Dynamic Faults for New SRAM Technologies", In IEEE Proc. Of European Test Workshop, pp. 29-34,2003.
[6] International SEMATECH, "International Technology Roadmap for Semiconductors (ITRS): Edition 2001"
[7] A.J. van de Goor, "Testing Semiconductor Memories, Theory and Practice" ComTex Publishing, Gouda, Netherlands, 1998.
[8] 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.
[9] "Xilinx ISE 6 Software Manuals and help – PDF Collection",http://toolbox.xilinx.com/docsan/xilinx7/ books/manuals .pdf
[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] I.S. Irobi, Z. Al-Ars, and S. Hamdioui. Detecting memory faults in the presence of bit line coupling in sram devices. IEEE International Test Conference, 2010
[17] Dr. R.K. Sharma Aditi Sood. "Modeling and Simulation of Microcode-based Built-In Self Test for Multi-Operation Memory Test Algorithms", IJCSI International Journal of Computer Science Issues, Vol. 7, Issue 3, No. 2, May 2010 pp.36-40