ISSN:0975-9646

Shaik Moulali et al, / (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 3 (1) , 2012, 3264 - 3268

Sram Cell Static Faults Detection and Repair Using Memory Bist

Shaik Moulali*, Dr. Fazal Noor Bhasha, B.Srinivas, S.Dayasagar chowdary , P.Srinivas, K. Hari Kishore

***Abstract*—Memories are one of the most universal cores. On average embedded RAMs occupy 90% area in system-on-chip (SOC), so embedded memory test design has become an essential part of the SOC development infrastructure. Here we designed reusable memory built in self test (MBIST) engine for memory test, which also gives useful information for fault diagnosis. A simple architecture for built in self repair is implemented. Integrating BISR in MBIST improves the chip yield. SOC consists of many memory models. Like, SRAM, FLASH, ROM etc, we consider here only SRAM type of memory core.Here we will see what the functional model of SRAM is and what types of Functional Faults, Fault Models and Defects exist in SRAM cores due to process variation and manufacturing. xilinx spartan3E tool used for synthesise and simulation.**

***Keywords*—Mbist, Bist, Bisr, Redundancy Logic.**

I.INTRODUCTION

Embedded memory test design has become an essential part of the system-on-chip (SOC) development infrastructure. According to the recent ITRS report, the memory cores will occupy more than 90% of the chip area in less than ten years. The yield of on-chip memories thus will dominate the chip yield. Go/no-go testing is no longer enough for embedded memories in the SOC era. Memory diagnostics is quickly becoming a critical issue, so far as manufacturing yield and time-to-volume of SOC products are concerned. Effective memory diagnostics and failure analysis (FA) methodologies will help improve the yield of SOC products, especially with rapid revolution in new product development and advanced process technologies.

Built-in Self Test, or BIST, is the technique of designing additional hardware and software features into integrated circuits to allow them to perform self-testing, i.e., testing of their own operation (functionally, parametrically, or both) using their own circuits, thereby reducing dependence on an external automated test equipment (ATE) BIST is a Design-for-Testability (DFT) technique, because it makes the electrical testing of a chip easier, faster, more efficient, and less costly. The concept of BIST is applicable to just about any kind of circuit, so its implementation can vary as widely as the product diversity that it caters to. The main drivers for the widespread development of BIST techniques are the fast-rising costs of ATE testing and the growing complexity of integrated circuits. It is now common to see complex devices that have functionally diverse blocks built on different technologies inside them. Such complex devices require high-end mixed-signal testers that possess special digital and analog testing capabilities. BIST can be used to perform these special tests with additional on-chip test circuits, eliminating the need to acquire such high-end testers.

Memory testing is a more and more important issue because RAMs are key components for electronic systems and Memories represent about 30% of the semiconductor market, embedded memories are dominating the chip yield. Memory testing is more and more difficult due to Growing density, capacity, and speed, Emerging new architectures and technologies, embedded memories: access, diagnostics & repair, heterogeneity, custom design, power & noise, scheduling, compression, etc.

Cost drives the need for more efficient test methodologies for fault modeling and simulation, test algorithm development and evaluation, diagnostics, DFT, BIST, BIRA, BISR, etc

II.MEMORIESANDMEMORYFAULTANALYSIS

A.BASIC FUNCTIONAL MODEL OF SRAM CORE

Basic functional model of SRAM is shown below. Many sub functional models are there, fault can be any where in the sub modules. Faults can be exist in address decoder logic, cell array, write driver etc. In section B we will see how different faults exist and what the fault models respectively are.

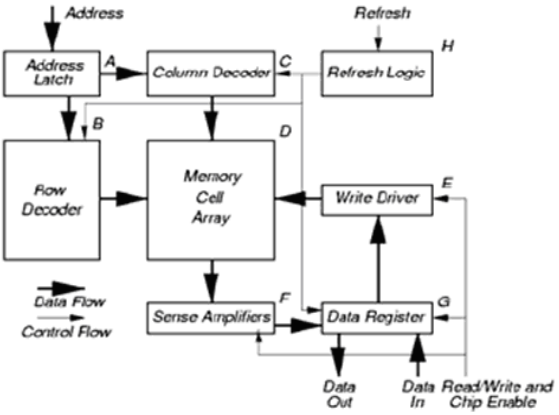
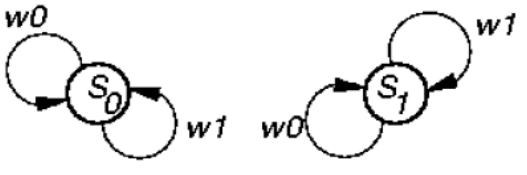


Fig. 1 BASIC FUNCTIONAL MODEL OF SRAM CORE

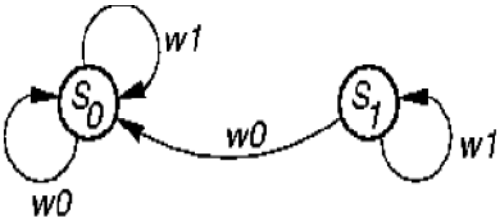
B.FAULT ANALYSIS

Stuck-at-Fault (SF): Either a cell or a line is stuck to logical`0'or`1'.



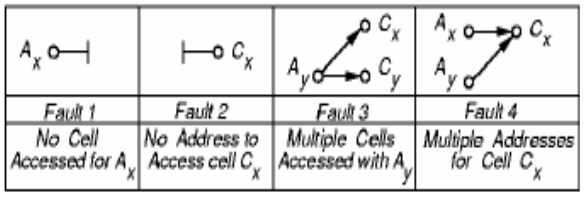
ST0 **ST1**

Transition Fault (TF): The 0!1 (or 1!0) transition is impossible on a cell or a line.



*Coupling Fault (CF)***:** When a cell is written to 0!1 (or 1!0), the content of the other cell is changed. CF is generalized to a k-coupling fault when k-1 cells are changed and is classified into Inversion or Idempotent coupling faults depending upon what content changed.

Address Decoder Fault (ADF): No cell will be accessed with a certain address or multiple cells are accessed simultaneously or a certain cell can be accessed with multiple addresses.



Retention Faults (RF): A cell fails to retain its logic value after some time. This fault is caused by a broken pull-up resistor.

Address Decoder Open Faults (ADOF): CMOS address decoder open faults are caused by open defects in the CMOS logic gates of the memory address decoders, and due to their sequential behavior, cannot be mapped to faults of the memory array itself.

Neighborhood Pattern Sensitive Fault (NPSF): a typical neighborhood pattern sensitive faults preventing the base cell from being transited to a certain value is called 'static'NPSF, and an NPSF is called `dynamic' when a transition on the neighborhood cells triggers a transition on the base cell.

C.RAM TEST ALGORITHM

A test algorithm (or simply test) is a finite sequence of test elements. A test element contains a number of memory operations (access commands). Data pattern (background) specified for the Read operation. Address (sequence) specified for the Read and Write operations. A March test algorithm is a finite sequence of March elements.There now exists a variety of industry-standard MBIST algorithms, such as the "March" algorithm, the checkerboard algorithm, and the varied pattern background algorithm.

III. MBIST (memory built in self test)

MBIST, as its name implies, is used specifically for testing memories. It typically consists of test circuits that apply, read, and compare test patterns designed to expose defects in the memory device. There now exists a variety of industry-standard MBIST algorithms, such as the "March" algorithm, the checkerboard algorithm, and the varied pattern background algorithm.

A.VARIOUS MBIST ALGORITHMS:

Classical Test Algorithms

Classical test algorithms are either (1) simple, fast but have poor fault coverage, such as Zero-one, Checkerboard; or (2) have good fault coverage but complex and slow, such as Walking, GALPAT, Sliding Diagonal, Butterfly, MOVI, and etc.. Due to these imbalanced conflicting traits, the popularity of these algorithms is decreasing.

March-based Test Algorithms

A March-based test algorithm is a finite sequence of March elements. A March Element is specified by an address order and a number of reads and writes. Examples of some March-based tests are 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.

B. IDEAL IMPLEMENTTATION OF MEMORY BIST(MBIST)

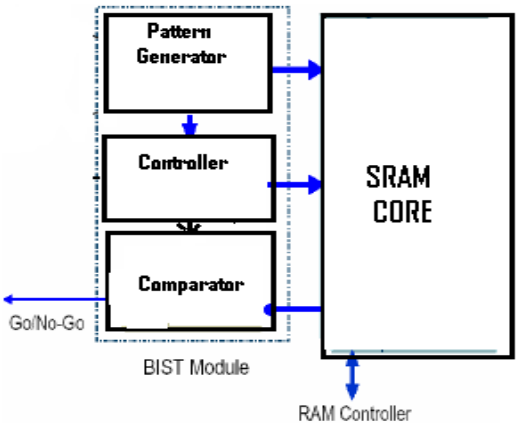


Fig.2 Block Diagram Of Memory Bist

BIST Controller

BIST controllers are mainly used to control pattern generators and comparators, and to strictly allow the controller to perform write and read operations in compliance with the Memory BIST Protocol.

Pattern Generator

Pattern generators are mainly used to generate different patterns required for a Memory BIST operation. These Patterns can be Marching ones and zeros, Alternations ones and zeros, random numbers and fixed patterns.

Comparator

Comparators are mainly used to compare the Memory Read data and compare against the expected values, the values generated by the pattern generator. Any failing comparisons will be flagged as BIST fail by the BIST Controller.

C.MBISR (memory built in self repair)

Today's deep submicron technologies allow the implementation of multiple memories on a single chip Due to their high density memories is more prone to faults. These faults impact the total chip Yield. One way to solve this problem is to enhance the memory by redundant memory locations.

The memory is tested by external test hardware or by on chip dedicated hardware (memory BIST). The second testing strategy is the preferred method for embedded memories.The present BIST concept prefers a redundancy logic that is placed in parallel to a memory without spare rows and spare columns. There will be no additional delay for the word redundancy logic on top of a memory in the data path of the memory. The memory is repaired during testing by storing faulty addresses in registers. Furthermore, the application can be started immediately after the memory BIST passes. The redundancy logic calculation will not increase the test time of the memory BIST.

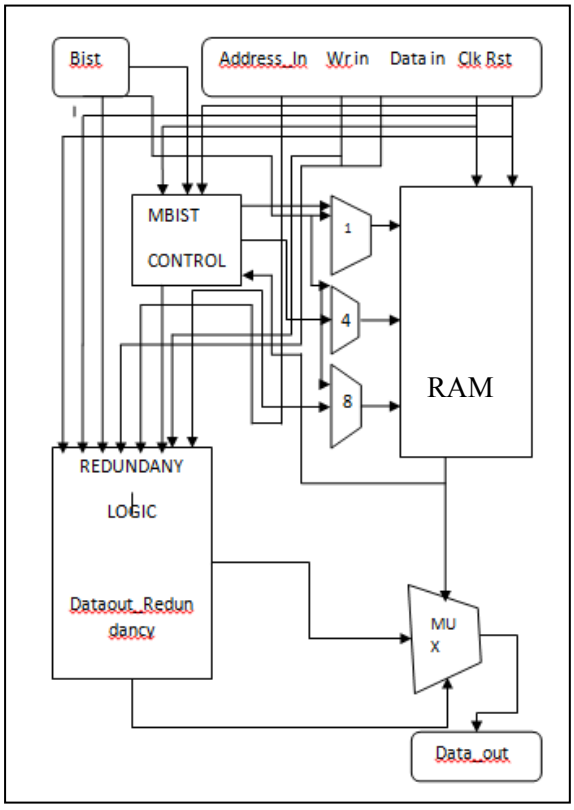


Fig.3 Architecture Diagrem For Mbist

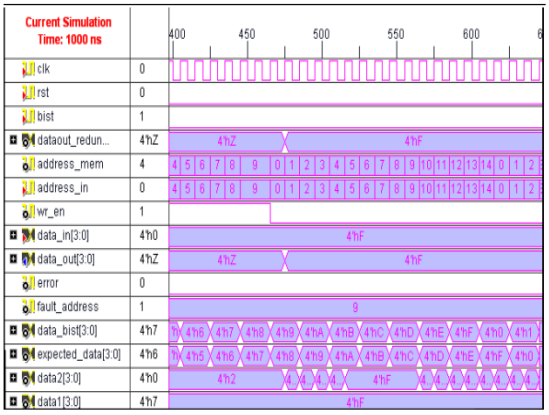
In the above architecture 1 represents 1-bit Mux, 4 represents 4-bit Mux, 8 represents 8-bit Mux.

Description

- Bist controller writes data to every memory location of the ram.
- Bist controller reads data from every memory location and compares with the actual written data.
- If both are same that memory location is functioning correctly.
- If the read data is different from actual written data in that particular memory location fault is there with the corresponding memory location.
- Bist controller sends all the fault address to the redundancy logic and those addresses are saved in redundancy logic register.
- Whenever external peripherals accessing memory if the operation to be performed is write then the input address is compares against all the fault address stored.
- If input address matches with any of the fault address the data to be written will be saved in the redundancy logic data register also along with memory.
- If the external operation is to be performed is read then the address in is compared against all the fault address in the redundancy logic.
- If the address matches with any of the fault address and also if fill bit is 1 then the output data will be taken from corresponding redundancy logic data register.
- If the input address is not matched with the fault address the output data will be read from ram.
- If The Input Address Is Not Matched With The Fault Address The Output Data Will Be Read From Ram.

IV.SIMULATION RESULTS

SIMULATION RESULT FOR ERROR DETECTION



SIMULATION RESULT FOR ERROR REPAIRING

V. SYNTHESIS RESULTS

Device utilization summary

Selected Device: 3s100evq100-5

Number of Slices:	68 out of	960	7%
Number of Slice Flip Flops:	81 out of	1920	4%
Number of 4 input LUTs:	96 out of	1920	5%
Number used as logic:	88		
Number used as Shift registers:	4		
Number used as RAMs:	4		
Number of IOs:	16		
Number of bonded IOBs	16 out of	66	24%
Number of GCLKs:	1 out of	24	4%

Timing Summary

Minimum period: 4.988ns

Maximum Frequency: 200.487MHz

HDL Synthesis Report

Macro Statistics

# RAMs	: 1
16x4-bit single-port RAM	: 1
# Counter	: 3
3-bit up counter	: 1
4-bit up counter	: 2
# Registers	: 22
1-bit register	: 4
4-bit register	: 18
# Comparators	: 9
4-bit comparator equal	: 5
4-bit comparator not equal	: 4
# Tristates	: 6
4-bit tristate buffer	: 6

Final Register Report

Macro Statistics

# Registers	: 77
Flip-Flops	: 77
# Shift Registers	: 4
2-bit shift register	: 4

Design Statistics

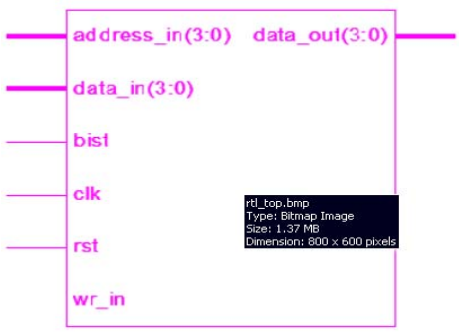
# IOs	: 16

Cell Usage

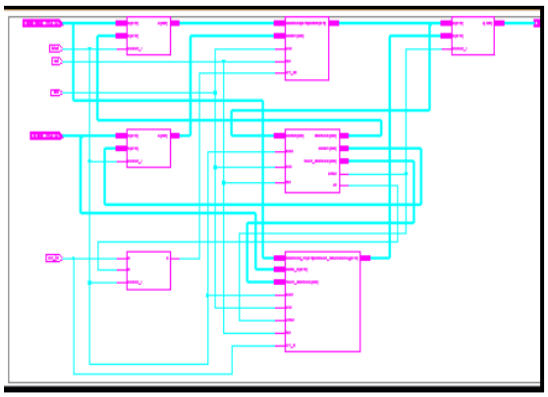
# BELS	: 97
# GND	: 1
# INV	: 3
# LUT2	: 9
# LUT2_L	: 1
# LUT3	: 20
# LUT4	: 45
# LUT4_D	: 6
# LUT4_L	: 4
# MUXF5	: 8
# FlipFlops/Latches	: 81
# FDC	: 12
# FDCE	: 16
# FDE	: 52
# FDP	: 1
# RAMS	: 4
# RAM16X1S	: 4
# Shift Registers	: 4
# SRL16E	: 4
# Clock Buffers	: 1
# BUFGP	: 1
# IO Buffers	: 15
# IBUF	: 11
# OBUF	: 4

RTL SCHEMATICS

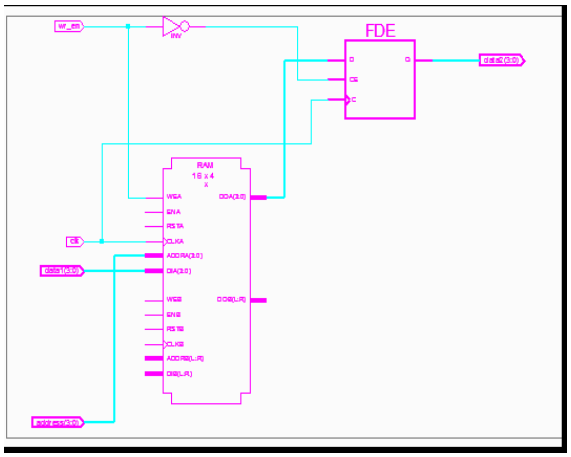
RTL_TOP



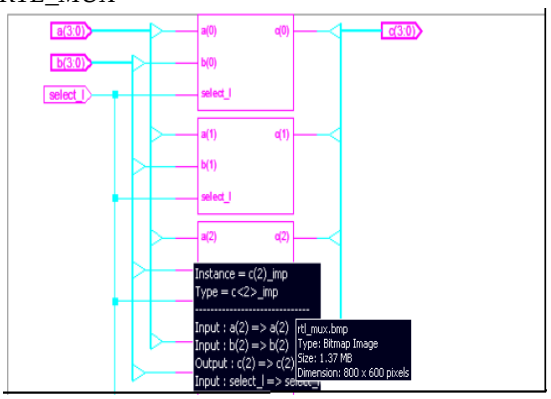
RTL_BIST CONTROLLER



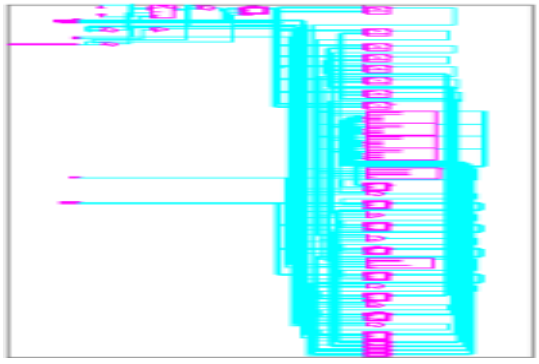
RTL_MEMORY



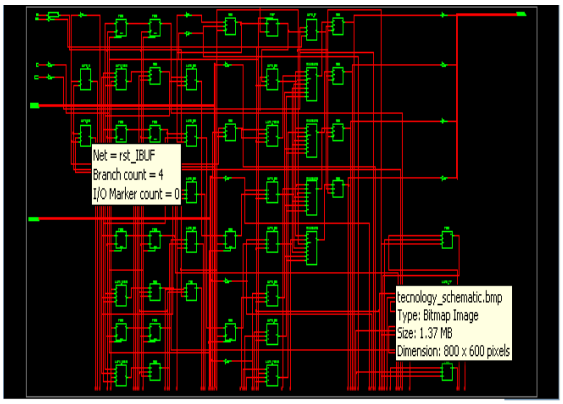
RTL_MUX



RTL_REDUNDANCY LOGIC



TECHNOLOGY SCHEMATIC



VI.CONCLUSION

Memories are the most universal component today. Almost all system chips contain some type of embedded memory, such as ROM, SRAM, DRAM, and flash. Memory testing is very important but challenging. Memory BIST is considered the best solution due to various engineering and economic reasons. March tests are the most popular algorithms currently implemented in BIST hardware. Various implementation schemes for memory BISTs are presented and their trade-offs are discussed. Integrating self repair feature improves the chip yield.

REFERENCES

[1].Y. Zorian and S. Shoukourian, "Embedded-memory test and repair: Infrastructure IP for SoC yield," *IEEE Design Test Computers*, vol. 20, pp. 58–66, May–June 2003.

[2] Y.-Y. Hsiao, C.-H. Chen, and C.-W.Wu, "A built-in self-repair scheme for NOR-type flash memory," in *Proc. IEEE VLSI Test Symp. (VTS)*, Berkeley, Apr. 2006, pp. 114–119.

[3] S. K. Thakur, R. A. Parekhji, and A. N. Chandorkar, "On-chip test and repair of memories for static and dynamic faults," in *Proc. Int. Test Conf. (ITC)*, Santa Clara, Oct. 2006, pp. 1–10, Paper 30.1.

[4] C.-D. Huang, J.-F. Li, and T.-W. Tseng, "ProTaR: An infrastructure IP for repairing RAMs in SOCs," *IEEE Trans. Very Large Scale Integr.(VLSI) Systems*, vol. 15, no. 10, pp. 1135–1143, Oct. 2007.

[5] T.-W. Tseng, J.-F. Li, C.-C. Hsu, A. Pao, K. Chiu, and E. Chen, "A reconfigurable built-in self-repair scheme for multiple repairable RAMs in SOCs," in *Proc. Int. Test Conf. (ITC)*, Oct. 2006, pp. 1–8, Paper 30.2.

[6]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.

[7]C.-T. Huang, C.-F. Wu, J.-F. Li, and C.-W. Wu, "Built-in redundancy analysis for memory yield improvement," IEEE Trans. on Reliability, vol. 52, no. 4, pp. 386-399, Dec. 2003.

[8]I. Kim, Y. Zorian, G. Komoriya, H. Pham, F. P. Higgins, and J. L. Lweandowski, ―Built in self repair for embedded high density SRAM□, in *Proc. Int. Test Conf. (ITC)*, Oct. 1998, pp. 1112–1119.

Shaik Moulali received the B.Tech. degree in Electronics & Communications Engineering from JNTU, Hyderabad in 2010 and pursuing M.Tech (VLSI) in K L University. His research interests include Analog VLSI Design, Digital VLSI Design and Low Power Memory Design and Fault Diagnosis.

Dr.Fazal Noorbasha was born on 29th April 1982. He received the B.Sc. degree in Electronics Science from BCAS College, Bapatla, Guntur, A.P., Affiliated to the Acharya Nagarjuna University, Guntur, A.P., India, in 2003, M.Sc. degree in Electronics from the Dr. HariSingh Gour Central University, Sagar, M.P., India, in 2006, M.Tech. Degree in VLSI technology, with specialization in Embedded systems from the North Maharashtra University, Jalgaon, M.S., INDIA in 2008 And Ph.D. from Department Of Physics and Electronics, Dr. HariSingh Gour Central University, Sagar, M.P., India, in 2011. Presently he is Assistant Professor, Department of ECE, KL University, where he has been engaged in the research and development of low-power, high-speed CMOS VLSI technology, Memory Processors LSI's, Digital Image Processing, Embedded Systems and Nanotechnology. He has published over 16 technical papers in international and National reputed journals and conferences.