International Journal of Advanced Research in Electronics and Communication Engineering (IJARECE)
Volume 3, Issue 9, September 2014

Efficient Built-in Self Repair Analyzer for Embedded word oriented SRAM and DRAM Memories with selectable redundancy

P. Neelima
Student, M.Tech, ECE Department,
Geethanjali College of Engineering & Technology
JNTUH, India

EVLN. RangaCharyulu,
Professor, M.E, ECE Department
Geethanjali College of Engineering & Technology
JNTUH, India

Abstract- **This paper proposes Built-In Self-Repair Analyzer (BISR) strategy with Redundancy which is an effective yield-enhancement strategy for embedded memories. It consists of a Built-In Self-Test (BIST) module, a Built-In Address-Analysis (BIAA) module and a Multiplexer (MUX) module. The BISR is designed flexible so that it can provide four operation modes to SRAM users. The feature of the proposed BISR strategy is that it can save each fault address for only once.To achieve a high repair speedin BIAA module, fault addresses and redundant ones form a one-to-one mapping. Besides, instead of adding spare words, rows, columns or blocks in the SRAMs, users can select normal words as redundancy. This paper proposes BISR strategy for DRAM also. Advantage of Proposed Work is that the BISR can perform bit oriented and word oriented memory analyzing and also it supports self-repair strategy with selectable redundancy. In the existing paper BISR is implemented for only SRAM, whereas the proposed paper extended it to DRAM. We design word oriented SRAM and DRAM and implement BISR analyzer with selectable redundancy. The selectable redundancy will provide the advantage of low area, low complexity and flexible to compiler designs.**

Keywords- SRAM; DRAM; Built-In Self-Repair (BISR); Built-In Self-Test (BIST); Built-In Address-Analysis (BIAA); Multiplexer (MUX); compiler.

I. INTRODUCTION

These days, the area occupied by embedded memories in System-on-Chip (SoC) is over 90%, and it is expected to rise up to 94%. Hence, the performance and yield of embedded memories will dominate that of SoCs. However, the yield of memory fabrication is limited largely by random defects, random leakage defects, random oxide pinholes, specific processing faults,gross processing and assembly faults, misalignments, gross photo defects and other faults and defects. In order to increase the reliability and yield of embedded memories,we have many redundancy mechanisms, both redundant rows and columns are incorporated into the memory array, spare words, rows, and columns are added into the word-oriented memory cores as redundancy. All these redundancy mechanisms result in the increase of area and complexity to embedded memories design. Let us consider that a compiler is used to configure SRAM for different needs, theBISR had better don't change other modules in SRAM. To solve this problem, a new redundancy scheme is proposed in this paper. We can selectsome normal words in embedded memoriesas redundancy, instead of adding spare words, spare rows, spare columns or spare blocks.

Testing the memory is necessary before using redundancy to repair. Design for test (DFT) techniques which are proposed in 1970 improves the testability by including additional circuitry. The DFT circuitry which is controlled through a BIST circuitry is more time-saving and efficient compared to that controlled by the external tester (ATE). However, memory BIST does not address the loss of parts due to manufacturing defects but it takes care of only the screening aspects of the manufactured parts. The aim of BISR techniques is to test embedded memories, save the fault addresses and replace them with redundancy. A new memory BISR strategy is applying two serial redundancy analysis (RA) stages. All the previous BISR techniques can repair memories, but they failed to tell us how to avoid storing fault address more than once. This paper proposes an efficient BISR strategy which can store each fault address only once.

II. BISR STRATEGY:

A. Redundancy Architecture

The proposed SRAM BISR strategy is known for its flexibility. The SRAM users can decide to choose whether to use it by setting a signal. So the SRAM redundancy is designed to be selectable. In another words, some normal words in SRAM can be selected as redundancy if the SRAM needs to repair itself. We call these words as Normal-Redundant words in order to distinguish them from the real normal ones. We take a 64 × 4 SRAM for example, as shown in Figure 1. There are 60 normal words and 4 Normal-Redundant words. When we use the BISR, the Normal-Redundant words are accessed as normal ones. Otherwise,onlywhen there are faults in normal wordsthe Normal-Redundant words can be accessed.In this case, the SRAM can only offer capacity of 60 words to users.

 All Rights Reserved © 2014 IJARECE

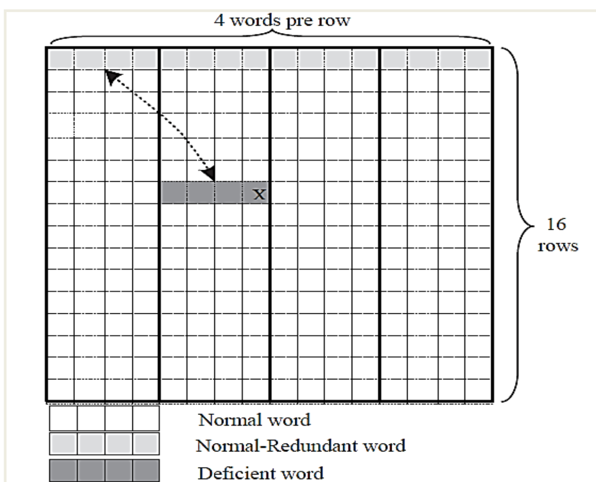


Figure1. Architecture of redundancy in SRAM

This kind of selectable redundancy architecture can save area and increase efficiency. After applying BISR, other modules in SRAM can remain unchanged. Thus the selectable redundancy does not create any problem to SRAM compiler.

B. Overall BISR Architecture:

The architecture of the proposed BISR strategy is shown in Figure 2. It consists of three parts namely BIST module, BIAA module and MUX module. We call the SRAM with BISR asa system. To test the addresses of the normal words in SRAM, the BIST module uses March C- test. It detects SRAM failures with a comparator that compares actual memory data with expected data. If there is a failure (compare_Q = 1), the current address is considered as a faulty address. The BIAA module can store faulty addresses in a memory named Fault_A_Mem. There is a counter in BIAA that counts the number of faulty addresses. When BISR is used (bisr_h = 1), the faulty addresses can be replaced with redundant addresses to repair the SRAM. The inputs of SRAM in different operation modes are controlled by the MUX module. In test mode (bist_h = 1), the inputs of SRAM are generated in BISR while they are equal to system inputs in access mode (bist_h = 0).

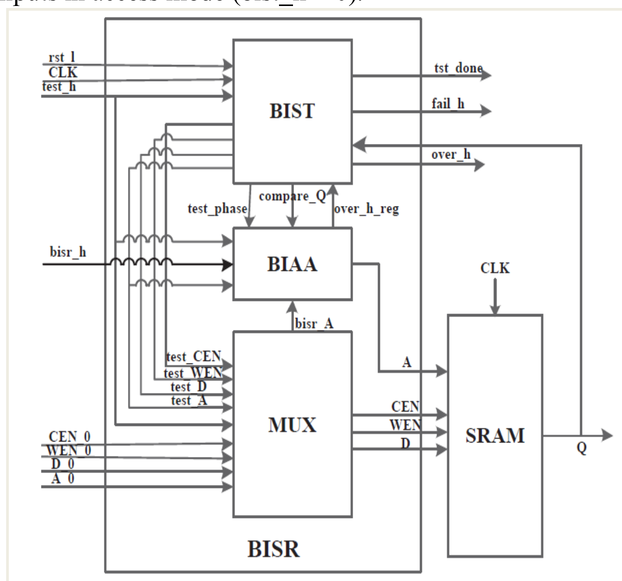


Figure2:Proposed Architecture

Algorithms	***Test length***	***Complexity***	***Fault coverage***
March C	11n	O(n)	AF, SAF, TF, CFin, CFid, and CFst
March C-	10n	O(n)	AF, SAF, TF, CFin, CFid, and CFst
March C+	14n	O(n)	AF, SAF, TF, SOF, CFin, and CFid
March 3	10n	O(n)	AF, SAF, SOF, and TF

Comparison of March Tests

1	Up	- write 0
2	Up	- read 0, write1
3	Up	- read 1, write0
4	Down	- read 0, write1
5	Down	- read 1, write0
6	Down	- read 0

March C- Steps

C. BISR procedure:

Figure 3 shows the proposed BISR flow diagram. The BISR starts its flow by resetting the system (rst_l = 0). After that if the system starts to work in test mode, it directly goes into TEST phase. During this phase, the BIAA module and BIST module work in parallel. The BIST uses March C- to test the normal addresses of SRAM. As long as any fault is detected by the BIST module, the faulty address will be sent to the BIAA module. Then the BIAA module checks if the faulty address has been already stored in Fault-A-Mem. If the obtained faulty address has not been stored, then the BIAA stores it and the faulty address counter adds 1. Otherwise, the faulty address can be ignored. When the test is finished, there will be two conditions. One of them is if there is no fault or there are too many faults that overflow the redundancy capacity, BISR goes into COMPLETE phase. Next if there are faults in SRAM but without overflows, then the system goes into REPAIR&TEST phase. In the same way as TEST phase, the BIST module and BIAA module work at the same time in REPAIR&TEST phase. The BIAA module will replace the faulty addresses stored in Fault-A-Mem with redundant ones and the BIST module will test the SRAM again. There will be two outputs- repair fail or repair pass. By using the BISR, the users can choose the SRAMs that can be repaired with redundancy or the ones without fault.

 All Rights Reserved © 2014 IJARECE

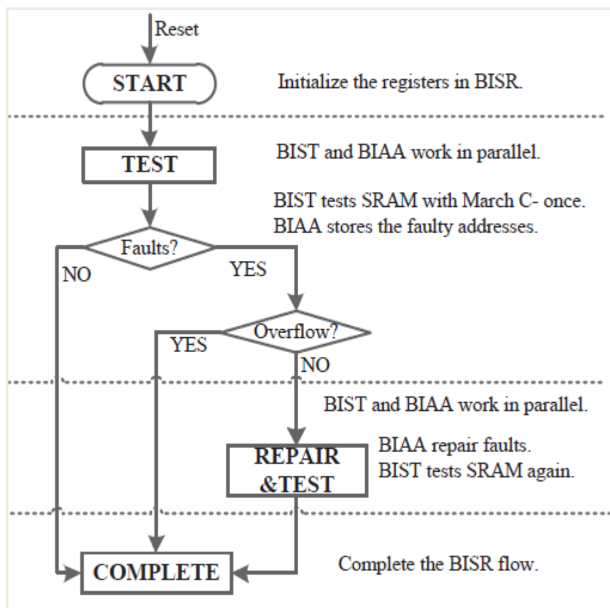


Figure3: Proposed BISR flow diagram

D. Features of the BISR

Firstly, the BISR strategy is flexible. Tablebelow listed the operation modes of SRAM. In access mode, SRAM users can decide whether the BISR is used based on their needs. If the BISR is required, the Normal-Redundant words will be taken as redundancy to repair fault. If BISR is not required, they can be accessed as normal words.

Modes	*Repair selection*	*Operation*
Test mode (*test_h*=1)	Default: repair (*bisr_h*=1)	Access normal words. Repair faults and test.
	Don't repair (*bisr_h*=0)	Access normal words. Test only.
Access mode (*test_h*=0)	Repair (*bisr_h*=1)	Access normal words. Repair faults and write/read SRAM.
	Don't repair (*bisr_h*=0)	Access Normal- Redundant and normal Words. Write/read SRAM only.

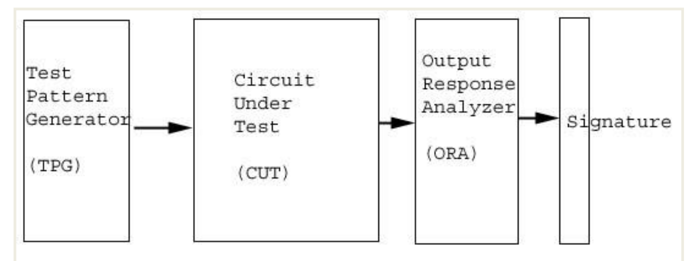
Operation modes of SRAM

The second feature is that the BISR strategy is efficient. On one side, the efficiency reflects on the selectable redundancy which is described as flexible above. It does not matter whether the BISR is applied or not, the Normal-Redundant words are used in the SRAM. It saves area and it has high utilization. On the other side, each fault address can be stored only once into Fault-A-Mem. The March C- has 6 steps. In another words, the addresses will be read 5 times in one test. Some faulty addresses can be detected in more than one step. For example take Stuckat- 0 fault, it can be detected in both 3rd and 5th steps. But the fault address should not be stored twice. So we propose an efficient method to solve this problem in BIAA module. BIST detects whether the current address is faulty address or not. If it is, then BIAA checks whether the Fault-A-Mem overflows. If not, the current fault address should be compared with those that are already stored in Fault-A-Mem. Only if the faulty address is not equal to any address in Fault-A-Mem, it can be stored. To simplify the comparison, write a redundant address into Fault-A-Mem as background. In this case, the fault address can be compared with all the data stored in Fault-A-Mem no question how many fault addresses have been stored previously.

III. BIST FOR DIGITAL CIRCUITS:

Testing of integrated circuits (ICs) is most important to ensure a high level of quality in functionality of product in both commercially and privately produced products. The impact of testing will have its affects on areas of manufacturing as well as those involved in design. If we give this range of design involvement, it is a major concernhow to go about best achieving a high level of confidence in IC operation. This desire to attain a highlevel of quality must be tempered along with the cost and time involved in this process. These two design considerations are at constant probabilities. It is with two goals in mind - effectiveness vs. cost/time, that Built-In-Self Test (BIST) has become a major design consideration in Design-For-Testability (DFT) methods.

BIST is beneficial in number of ways. Firstly, it can reduce dependency on external Automatic Test Equipment (ATE). This aspect will have impact on the cost/time constraint because the ATE will be used less by the current design and can be used somewhere else or on other devices. In addition, the BIST can provide high speed, in system testing of the Circuit-Under-Test (CUT). This is very important to the quality component of testing. Along with that, BIST can overcome pin limitations due to packaging, and make efficient use of available extra chip area, and provide more detailed information about the faults present. All these advantages are plentiful motivations for BIST.



Basic BIST Architecture Block Diagram

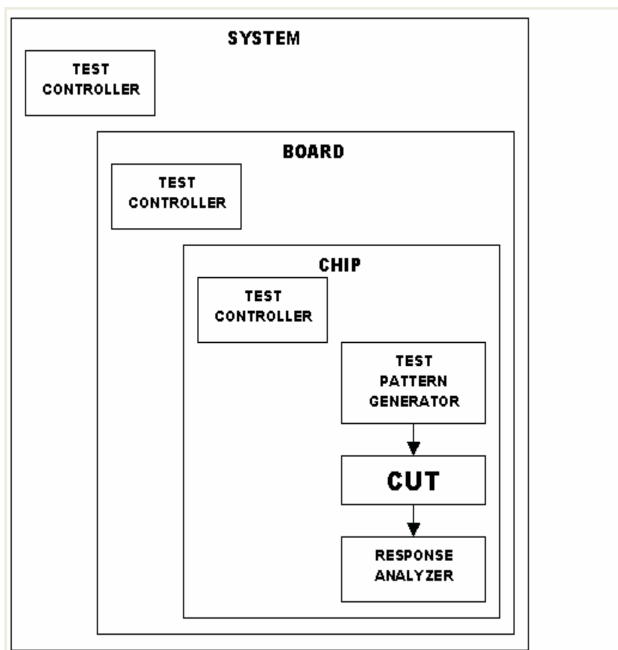
What is BIST?

The basic concept of BIST involves: design of test circuitry around a system that tests the systemautomatically by applying certain test stimulus and observing the response of corresponding system. As the test framework is embedded directly into the system hardware, the testing process has the potential of being faster and more cost effective than using an external test setup. One of the first definitions of BIST was given as: "The ability of logic to verify a failure-free status automatically, without the need for externally applied test stimuli (other than power and clock), and without the need for the logic to be part of a running system."

 All Rights Reserved © 2014 IJARECE

Basic BIST Hierarchy:

Figure below shows the basic BIST hierarchy. The test controller at the system level can activate self-test on all boardssimultaneously. In succession, the test controller on each and every board activates self-test on each chip on that specific board. For the circuit under test (CUT) the pattern generator produces a sequence of test vectors, while the response analyzer compares the output response of the CUT with its fault-free response.



Basic BIST Hierarchy

IV. IMPLEMENTATION

The overall Architecture will be designed using HDL language and simulation, synthesis and FPGA implementation (Translation, Mapping, Placing and Routing) will be done using various FPGA based EDA Tools.

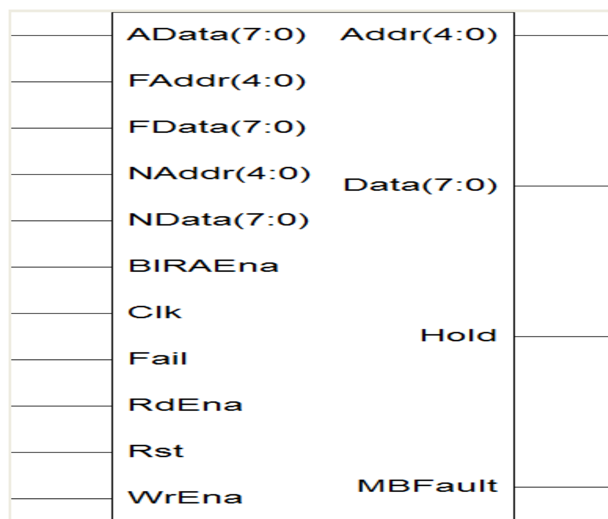
VLSI EDATools:

Xilinx ISE: Integrated Software Environment (ISE) enables to quickly Design, Simulation of HDL source, Synthesis of HDL based RTL design and FPGA Implementation (Placing, routing ,mapping) and Bit Stream generation. *Languages (HDL):*Verilog/VHDL

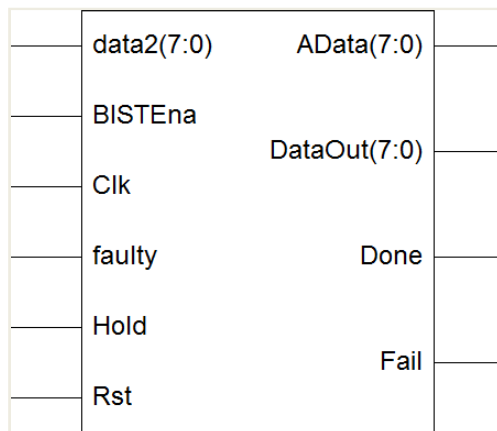
V. EXPERIMENTAL RESULTS

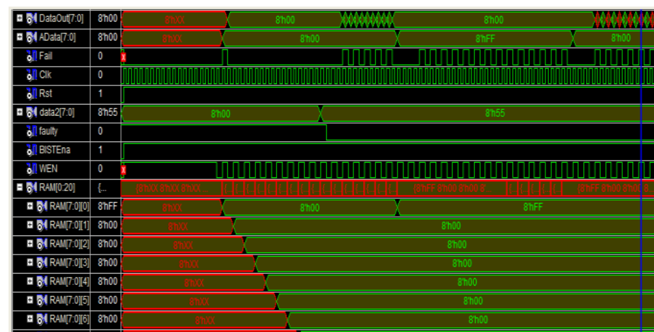
The simulation results obtained using VLSI EDATools(Xilinx ISE, Languages (HDL)) are shown below.

BIRA:



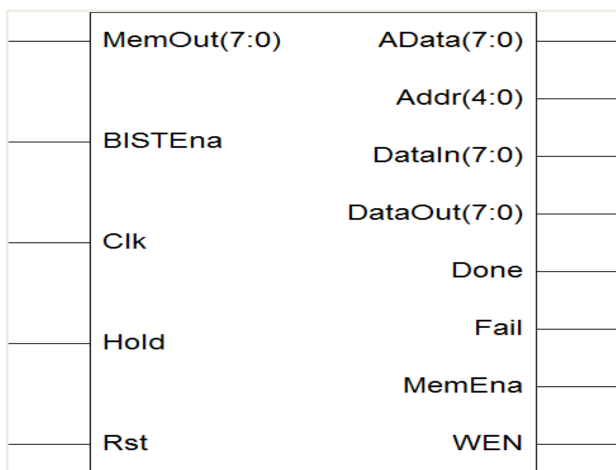
BIST with MEMORY:

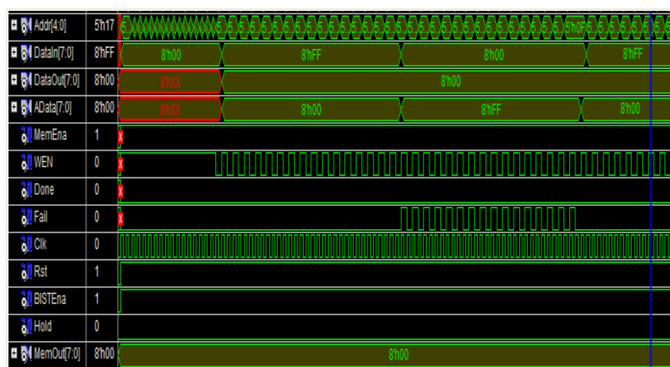




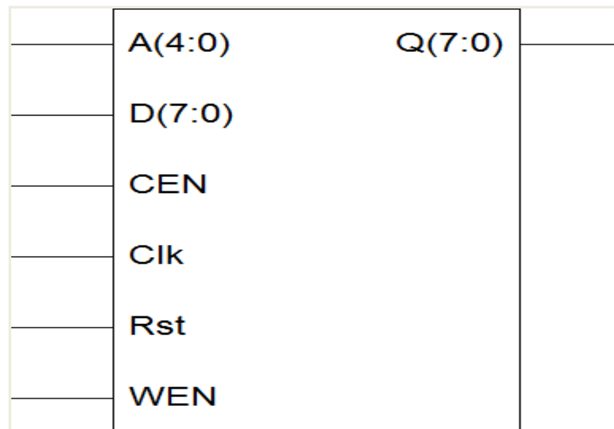
 All Rights Reserved © 2014 IJARECE

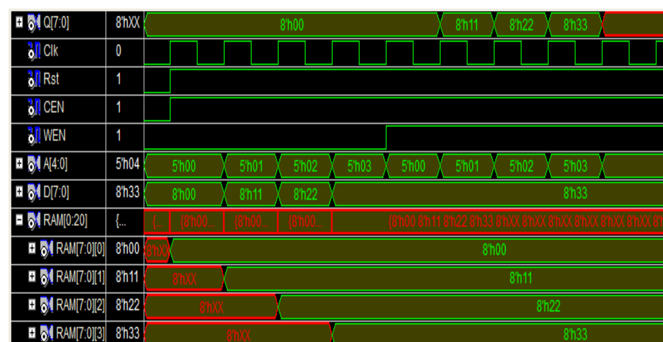
BIST ONLY:



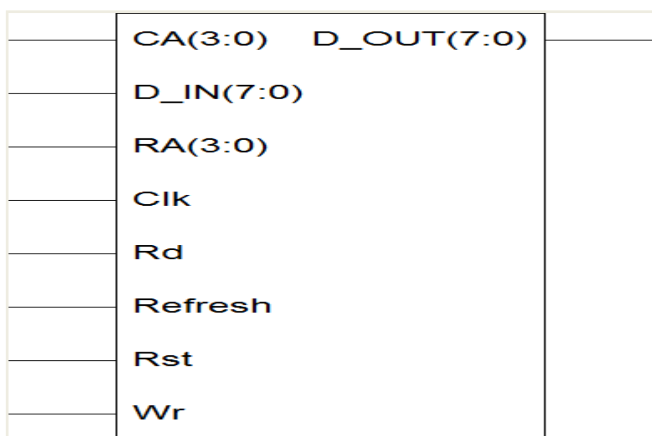


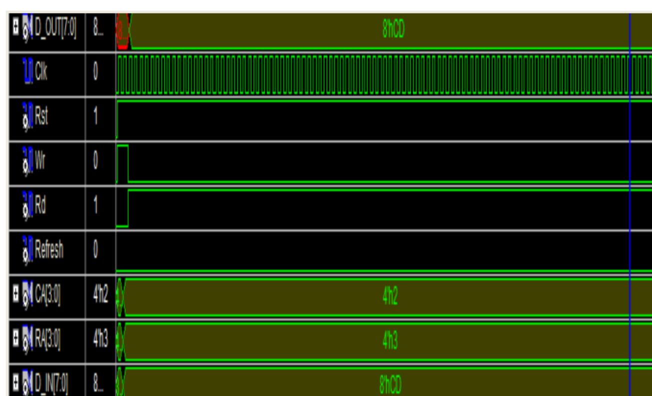
SRAM:



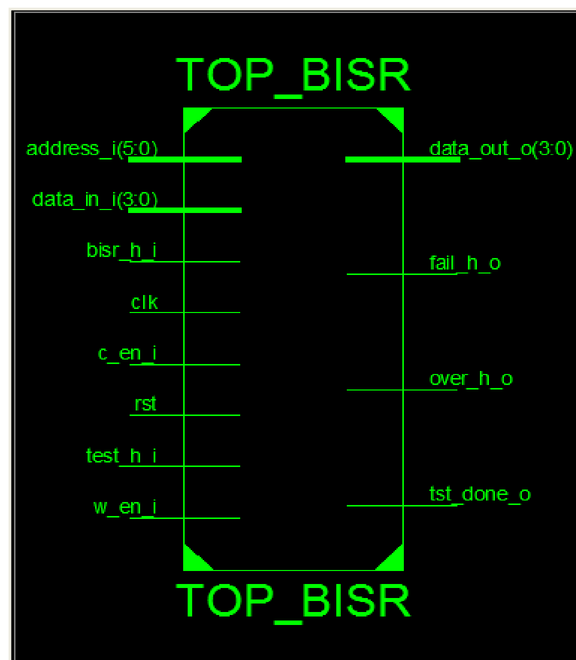


DRAM



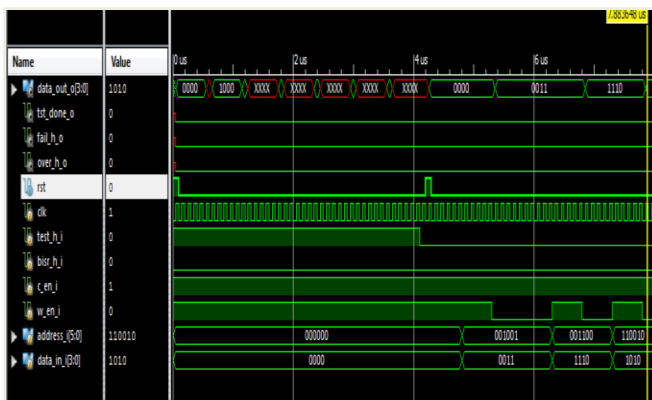


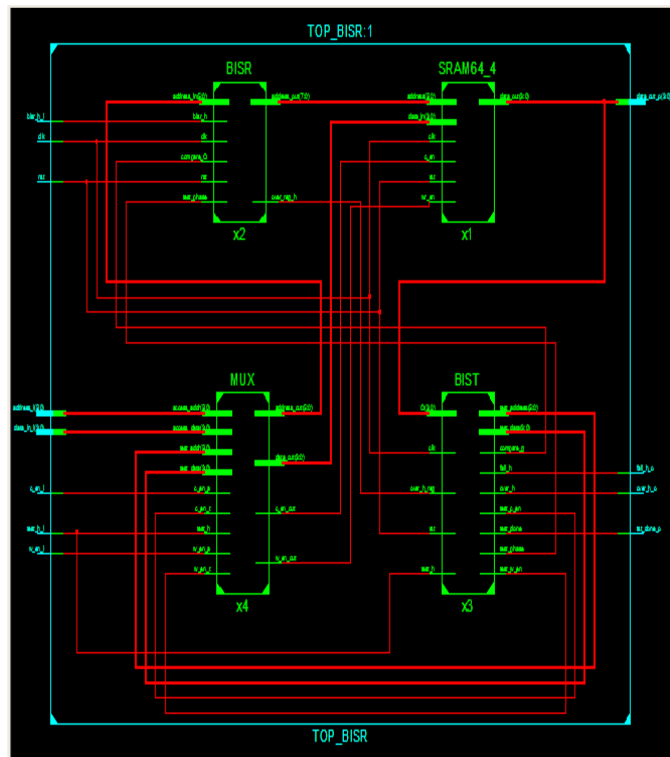
TOP MODULE:



 All Rights Reserved © 2014 IJARECE

International Journal of Advanced Research in Electronics and Communication Engineering (IJARECE)
Volume 3, Issue 9, September 2014





V. CONCLUSION

Memory testing is very important but challenging. Memory BIST is considered as the best solution for its various engineering and economic reasons. March tests are the most popular algorithmsthat are presently implemented in BIST hardware.Using Defect Coverage instead of Fault Coverage as ourmeasure for test quality is revolutionary.Integrating diagnostic capabilities into BIST improves overall system robustness and chip yield. Automatic generation eases design efforts for test integration and help satisfying time-to-market requirements.Self-reparability is the key to fault-tolerant and reliable circuit. We have designed word oriented SRAM and DRAM memories and implemented BISR analyzer with selectable redundancy.In conclusion, the future Memory BIST designs should be fast, small, efficient, robust, and flexible.

REFERENCES

1. P. Ohler, S. Hellebrand, and H.-J. Wunderlich, “An integrated built-in self-test and repair approach for memories with 2D redundancy,” in Proc. IEEE Eur. Test Symp. (ETS), Freiburg, May 2007, pp. 91–99.
2. Semiconductor Industry Association, "International technology roadmap for semiconductors (ITRS), 2003 edition," Hsinchu, Taiwan, Dec.2003.
3. C. Stapper, A. Mclaren, and M. Dreckman, "Yield model for Productivity Optimization of VLSI Memory Chips with redundancy and Partially good Product," IBM Journal of Research and Development, Vol. 24, No. 3, pp. 398-409, May 1980.
4. W. K. Huang, Y. H. shen, and F. lombrardi, "New approaches for repairs of memories with redundancy by row/column deletion for yield enhancement," IEEE Transactions on Computer-Aided Design, vol. 9, No. 3, pp. 323-328, Mar. 1990.
5. P. Mazumder and Y. S. Jih, "A new built-in self-repair approach to VLSI memory yield enhancement by using neuraltype circuits," IEEE transactions on Computer Aided Design, vol. 12, No. 1, Jan, 1993.
6. H. C. Kim, D. S. Yi, J. Y. Park, and C. H. Cho, "A BISR (built-in self-repair) circuit for embedded memory with multiple redundancies," VLSIand CAD 6th International Conference, pp. 602-605, Oct. 1999.
7. Shyue-Kung Lu, Chun-Lin Yang, and Han-Wen Lin, "Efficient BISR Techniques for Word-Oriented Embedded Memories with HierarchicalRedundancy," IEEE ICIS-COMSAR, pp. 355-360, 2006.
8. C. Stroud, A Designer's Guide to Built-In Self-Test, Kluwer Academic Publishers, 2002.
9. Karunaratne. M and Oomann. B, "Yield gain with memory BISR-a casestudy," IEEE MWSCAS, pp. 699-702, 2009.
10. I. Kang, W. Jeong, and S. Kang, " High-efficiency memory BISR with two serial RA stages using spare memories," IET Electron. Lett., vol. 44, no. 8, pp. 515-517, Apr. 2008.
11. Heon-cheol Kim, Dong-soon Yi, Jin-young Park, and Chang-hyun Cho, "A BISR (Built-In Self-Repair) circuit for embedded memory with multiple redundancies," in Proc. Int. Conf. VLSI CAD, Oct. 1999, pp.602-605.
12. M. Sachdev, V. Zieren, and P. Janssen, " Defect detection with transient current testing and its potential for deep submicron CMOS ICs," IEEE International Test Conference, pp. 204-213, Oct. 1998.
13. Mentor Graohics, MBISTArchitect Process Guide, Software Version 8.2009 3, Aug 2009, pp. 113-116.
14. Pavlov. Andrei and Sachdev. Manoj, CMOS SRAM Circuit Design and Parametric Test in Nano-Scaled Technologies, CA: Springer, 2008, pp.85-86.

 All Rights Reserved © 2014 IJARECE

First Author

P.Neelima
M.Tech Scholar in VLSI System Design,
ECE Department,
Geethanjali College of Engineering and Technology
JNTUH, India.

Second Author

EVLN.RangaCharyulu
B.Tech, M.E, Professor,
ECE Department,
Geethanjali College of Engineering and Technology
JNTUH, India.

,