ISSN: 2393-8528

Contents lists available at www.ijicse.in

International Journal of Innovative Computer Science & Engineering

Volume 1 Issue 2; Page No. 28-34

Design of an Area efficient BISR scheme For Multiple Memories

[1] Perika Kalanwesh, [2] Vamshi Kanth Reddy

[1]Student at Sri Indu College of Engineering and Technology, Hyderabad, India.

[2]Assistant Professor at CMRIT, Hyderabad, India.

ARTICLE INFO

Received 19 Oct. 2014
Accepted 20 Nov. 2014

Corresponding Author:

Perika Kalanwesh

Student at Sri Indu College of Engineering and Technology, Hyderabad, India.

ABSTRACT

The Proposed Built-in self-repair (BISR) scheme for multiple embedded memories to find optimum point of the performance of BISR for multiple embedded memories. All memories are concurrently tested by the small dedicated built-in self-test to figure out the faulty memories, the number of faults, and irreparability. After all memories are tested, only faulty memories are serially tested and repaired by the shared built-in redundancy analysis according to the sizes of memories in descending order. Thus, the fast test and repair are performed with low area overhead. To accomplish an optimal repair rate and a fast analysis speed, an exhaustive search for all combinations of spare rows and columns is proposed based on the optimized fault collection. In proposed BISR includes BIST and BIRA, The BIST is design by using of March X algorithm.

By using this algorithm the delay and area is reduced. This proposed design is programmed using Verilog - HDL using Xilinx ISE 14.7. The FPGA implementation is done on Spartan3E.

Index Terms— Built-in redundancy analysis (BIRA), built-in self-repair (BISR), built-in self-test (BIST), March X algorithm.

©2014, IJICSE, All Right Reserved.

I. INTRODUCTION

The capacity and density of semiconductor memories have increased, the probability of memory faults has increased. This causes decreases in yield and quality degradation. Because embedded memories in a systemon-chip (SoC) occupy most of the chip area, the yield of embedded memories heavily influences the yield of the SoCs. In addition, embedded memories will occupy up to 69% of the SoC chip area by 2017 [1]. Improving SoC memory yield is commonly accomplished with built-in self-repair (BISR). There are many BISR schemes, including built-in self-test (BIST) and built-in redundancy analysis (BIRA), to test and repair the embedded memories [2]–[17]. CRESTA focuses on the optimal repair rate, which is the ratio of memories repaired by BIRA to all repairable memories of the total tested memories, and fast analysis speed [2]. It simultaneously analyzes the entire cases of possible solutions with several parallel sub analyzers. Its area overhead increases with an increasing number of redundancies. Local repair most and essential spare pivoting mainly focus on reducing the area overhead in terms of storage requirements with a simple redundancy analysis (RA) algorithm [3]. Although their repair rates are not optimal, these two BIRAs have inspired the creation of other algorithms due to their simplicity and small area overhead. Intelligent Solve (IS) and Intelligent Solve First (ISF) algorithms achieve both a low area overhead and an optimal repair rate [4]. However, the two algorithms take a lot of time to complete the RA in the cases with complex fault distributions. Selected fail count comparison (SFCC) [5] achieves a low area overhead and an optimal repair rate, and it also reduces the search space by building a search tree based on the line faults. The BRANCH [6] analyzer analyzes all nodes concurrently within a branch for combinations of 2-D spares. There are various BISR schemes for multiple memories. Because routing, timing, and power consumption are realistically considered, the memories are partitioned into memory groups [18]. Most BISR schemes perform test and repair for the memory groups. Processor-based BISR schemes using infrastructure intelligent property have been proposed in [8]–[11]. A BISR scheme based on a self-test and repair processor can perform multiple-time memory repair [9]. To perform the RA algorithm, the BISR scheme uses specific instructions [11]. However,

because the processor-based BISR schemes use specific instructions to construct the RA algorithm and the RA algorithm may need multiple instructions to be implemented, the test and repair time can be long. In [14], reconfiguration BISR schemes based on a serial test and repair are proposed to reduce the area overhead. Although they can reduce area overhead through hardware sharing, the test and repair time can be long because of the memory repairs are completed serially. To reduce the test and repair time, a BISR scheme based on a parallel procedure is proposed [16]. However, the area overhead required to implement these BISR schemes is very high, because they use many BIST and BIRA modules. In [17], parallel test and serial repair BISR schemes are proposed. These BISR repair memories in serial after all memories have simultaneously been tested. The area overhead, and test and repair time are placed in the middle of the parallel test and repair and the serial test and repair. In this paper, an optimized BISR scheme is proposed for multiple embedded memories in the SoCs. All memories are concurrently tested by the small dedicated BIST to figure out the faulty, the number of faults, and irreparability. After all memories are tested, only faulty memories are serially tested and repaired by the shared BIRA.

II. BACKGROUND

A. Performance Criteria of BISR

The performance of BISR is evaluated with BIST and BIRA performance criteria. BIST performance criteria, such as test time, fault coverage, and area overhead, depend on the test algorithm. There are three main performance criteria for BIRA: 1) repair rate; 2) repair time; and 3) area overhead. The repair rate is defined as the ratio of the number of repaired memories to the number of faulty memories. Because the numbers of faults and spare memories are much smaller than the number of memory cells, BIST time dominates the test and repair time.

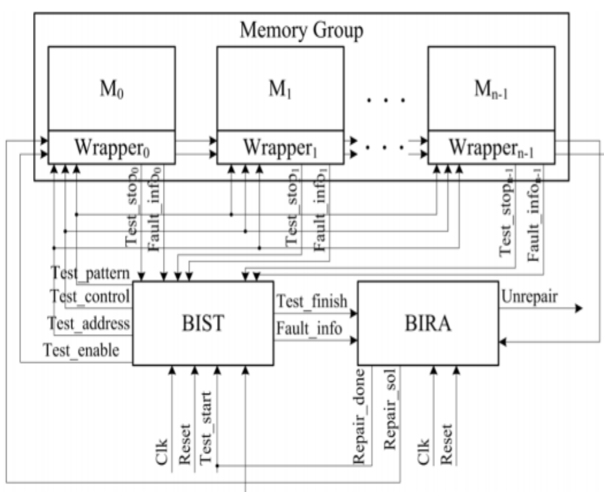


Fig. 1: Block diagram of the proposed BISR architecture

Although BIST time dominates the test and repair time, many studies have attempted to reduce the testing time using an efficient March c algorithm. A BIRA module requires more area than a BIST module. Therefore, the area overhead for BISR should be decreased by reducing the storage elements. However, all fault information is stored to achieve an optimal repair rate using an exhaustive search method. Because there is a tradeoff between these features, it is difficult to improve all three features at the same time.

B. Test and Repair Methods for Multiple Memories

There are three methods for multiple embedded memories: 1) a parallel test and parallel repair method; 2) a parallel test and serial repair method; and 3) a serial test and serial repair method. The parallel test and parallel repair method requires n BIST modules and n BIRA modules where there are n memories. The BIST and BIRA modules are proportional to the number of embedded memories, because each memory has dedicated BIST and BIRA modules. All the memories are concurrently tested and repaired by each BISR module. The area overhead is the largest among the BISR methods. However, this method can offer the shortest test and repair time among the BISR methods because of the concurrent test and repair process. The parallel test and serial repair method requires n BIST modules and one BIRA module. After all the memories are concurrently tested, the repair is serially performed. Although there is only one BIRA module for the serial repair approach, the number of fault storage elements is required to be the same as the number of memories. Because the faults can be concurrently detected from other memories, they must be stored separately. If the fault information is mixed without distinguishing the memory identification, the repair solutions cannot be exactly derived. To avoid this error, the faults are stored separately for each memory. The memories are serially tested and repaired by the serial test and serial repair method. It requires minimal area overhead, because there is one BISR for multiple memories. The test and repair time are long due to the serial procedures. Because the BIST time is almost as long as the test and repair time, it is necessary to reduce BIST time. To reduce area overhead, and test and repair time, the proposed BISR scheme is based on two stages of a test and repair method. Only faulty memories are serially tested and repaired after all the memories are tested at the same time.

III. PROPOSED BISR SCHEME

I. Proposed BISR Scheme

Because there are many memories in the SoC, the memories are partitioned into groups considering the routing, timing, and power consumption. Each group has the proposed BISR circuit. In other words, the

number of required BISR circuits is equal to the number of memory groups in the SoC. Then, the proposed BISR targets multiple memories in each group. All the memories are tested and classified as faulty memory or fault-free memory by parallel test procedure. The faultfree memories are excluded from the test procedure, because it is not necessary to repair the fault-free memories. Only faulty memories are selected and serially tested and repaired according to the size of memories in descending order. Fig. 1 shows a block diagram of the proposed BISR architecture. It mainly consists of BIST and BIRA modules. The main purpose is to classify the memories as faulty or not, and the number of faults in each memory is stored in the dedicated wrapper. When the fault is detected by BIST, the fault information is sent to BIRA through the port Fault_info. After the test is completed or stopped, the signal Test_ finish is activated and BIRA executes the RA process to find repair solutions. In the repair procedure, RA based on the exhaustive search for all combinations of spare rows and columns is performed. If the memory under test cannot be repaired, the signal Unrepair is activated, and the test and repair are finished and the SoC is determined as a faulty chip due to the irreparable memory. If the memory under test can be repaired, the repair is performed by the repair solution.

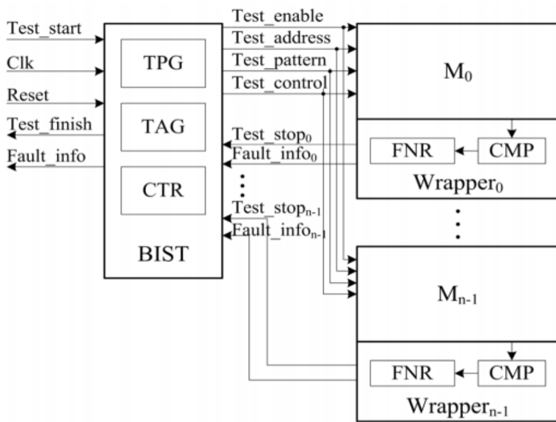


Fig. 2: Block diagram of the proposed BIST and wrapper modules

Fig. 2 shows a conceptual block diagram of the proposed BIST and wrapper modules. Assume that the number of memories is n. The BIST module consists of a test pattern generator (TPG), a test address generator (TAG), and a controller (CTR). The wrapper, which is dedicated to the memory, consists of a comparator (CMP) and a fault number register (FNR). When the signal Test_start is asserted, TAG and TPG generate test patterns (Test_pattern) and test addresses (Test_address) for executing the adopted March test algorithm. The overall test procedure is controlled by the signal Test_control. The clock and reset signals are provided through Clk and Reset. The CMP compares the results from the memory and expected responses to detect faults. If a fault is detected on the largest memory, the fault information is sent to BIST through the port Fault_infofirst. Because memory can be repaired by BIRA sequentially, if the largest memory is faulty, the fault information is sent to BIRA during the parallel test. After the parallel test and repair for the faulty memory are done, the rest of the memories are classified as faulty or not according to the value of the each FNR. The test order is predetermined according the sizes of the memories.

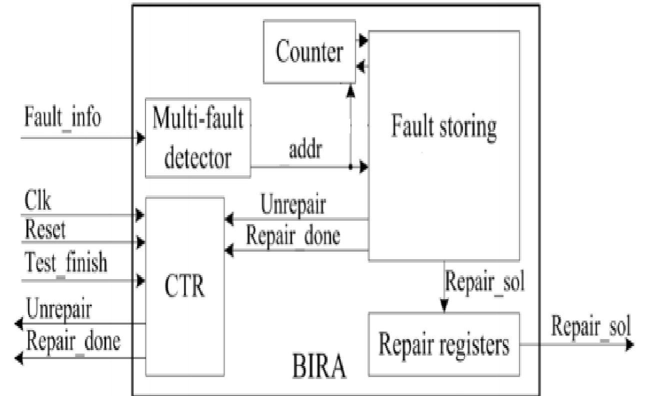


Fig. 3: Block diagram of the proposed BIRA architecture.

Fig. 3 shows a conceptual block diagram of the proposed BIRA architecture. It consists of a multi fault detector, a counter, fault storing, repair registers, and a CTR. If a fault is detected, it is sent to the multi fault detector through the port Fault_info. There can be multiple faults in a word within a word-oriented memory. If there are multiple faults in a word, the faults are stored serially in the fault storing. The fault address is sent to the counter to verify that the fault satisfies the must-repair condition [27]. When the signal Test_ finish is high, the RA algorithm based on the exhaustive search for all combinations of spare rows and columns is executed according to the information in the fault storing. After the repair process is done, the signal Repair_done is activated. If the memory under test is not repairable, the signal Unrepair becomes high and the test and repair procedure is finished. If the memory under test is repairable, the repair solutions can be found and they are stored in the Repair registers.

II. PROPOSED BIST SCHEME

BIST consists of *n* march elements, denoted by *M*i, with ($0 \leq i < n$). Each march element comprises zero (or more) write operations, denoted by *w*0/*w*1, meaning that 0/1 is written to the Memory cell, and zero (or more) read operations denoted by *r*0/*r*1, meaning that 0/1 is expected to be read from the memory cell. The March X algorithm [Fig. 3(a)] consists of four March elements denoted by *M*0 to *M*3. In Fig. 1, ⇑ denotes an increasing addressing order (which can be any arbitrary

addressing order) and ⇓ denotes a decreasing addressing order (which is the inverse addressing order of ⇑).

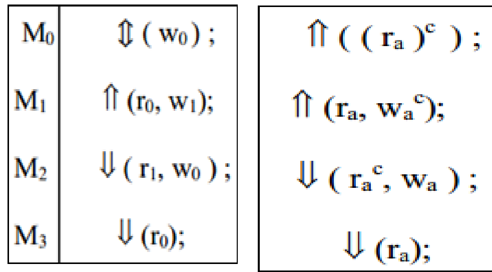
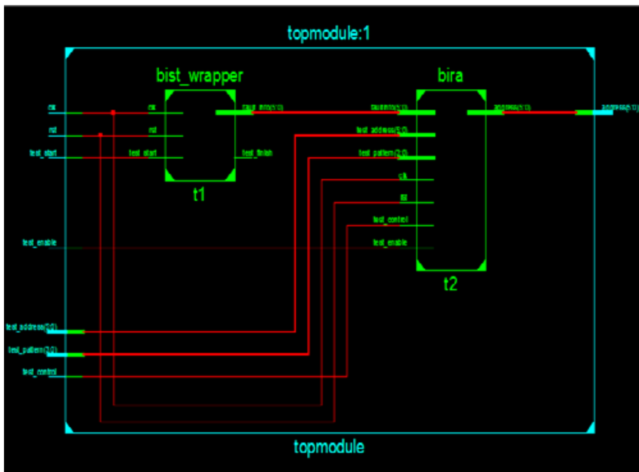


Figure 4: March X algorithm used in BIST

III. RTL AND SCHEMATRIC DIAGRAMS

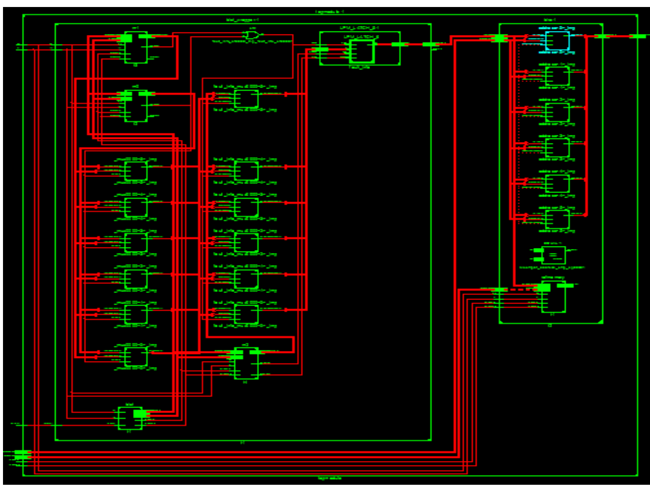
RTL Diagram:

The RTL diagram for BISR is shown below, in that includes BIST wappers and also BIRA modules



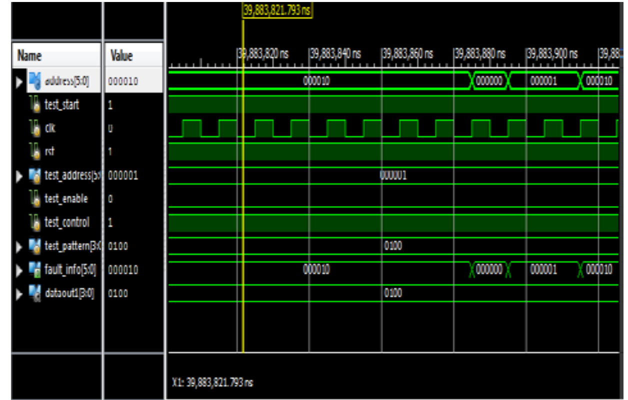
SCHEMATRIC diagram for BISR:-

The Sehematric diagram for BISR is shown below



IV. SIMULATION AND SYNTHESIZE RESULS

Simulation wave forms:-



Synthesize results:-

Area:-

	required	Available
NO. Of slices	35	4656
NO. Of 4 input LUTs	74	9312

Delay:-

Delay	5.299 **ns**

V. CONCLUSION

VI. A BISR technique for multiple embedded memories is proposed. To find optimum point of the performance of BISR for multiple embedded memories, the proposed BISR scheme is proposed. All memories are concurrently tested by the small dedicated BIST to figure out the faulty, the number of faults, and irreparability. After all memories are tested, only faulty memories are serially tested and repaired by the global BIRA according to the sizes of memories in descending order. The proposed BISR scheme finds the optimum point between the test and repair time.

REFERENCES

1. Semico Res. Corp., Phoenix, AZ, USA, ASIC IP Rep., 2007. [Online]. Available: http://www.semico.com/content/semico-systemschip-%E2%80%93-braver-new-world
2. T. Kawagoe, J. Ohtani, M. Niiro, T. Ooishi, M. Hamada, and H. Hidaka, "A built-in self-repair analyzer (CRESTA) for embedded DRAMs," in Proc. Int. Test Conf., Oct. 2000, pp. 567–574.
3. C.-T. Huang, C.-F. Wu, J.-F. Li, and C.-W. Wu, "Built-in redundancy analysis for memory yield improvement," IEEE Trans. Rel., vol. 52, no. 4, pp. 386–399, Dec. 2003.
4. P. Öhler, S. Hellebrand, and H.-J. Wunderlich, "An integrated built-in test and repair approach for memories with 2D redundancy," in Proc. 12th IEEE Eur. Test Symp., May 2007, pp. 91–96.
5. W. Jeong, I. Kang, K. Jin, and S. Kang, "A fast built-in redundancy analysis for memories with optimal

repair rate using a line-based search tree," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 17, no. 12, pp. 1665–1678, Dec. 2009.

6. W. Jeong, J. Lee, T. Han, K. Lee, and S. Kang, "An advanced BIRA for memories with an optimal repair rate and fast analysis speed by using a branch analyzer," IEEE Trans. Comput.-Aided Design Integr. Circuits Syst., vol. 29, no. 12, pp. 2014–2026, Dec. 2010.
7. Tanabe et al., "A 30-ns 64 Mb DRAM with built-in self-test and self-repair function," IEEE J. Solid-State Circuits, vol. 27, no. 11, pp. 1525–1533, Nov. 1992.
8. Y. Zorian, "Embedded memory test and repair: Infrastructure IP for SOC yield," in Proc. Int. Test Conf., Oct. 2002, pp. 340–349. [9] Y. Zorian and S. Shoukourian, "Embedded-memory test and repair: Infrastructure IP for SoC yield," IEEE Des. Test Comput., vol. 20, no. 3, pp. 58–66, May/Jun. 2003.
9. C.-D. Huang, T.-W. Tseng, and J.-F. Li, "An infrastructure IP for repairing multiple RAMs in SOCs," in Proc. Int. Symp. VLSI Design, Autom. Test, Apr. 2006, pp. 1–4.
10. C.-D. Huang, J.-F. Li, and T.-W. Tseng, "ProTaR: An infrastructure IP for repairing RAMs in system-on-chips," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 15, no. 10, pp. 1135–1143, Oct. 2007.
11. C.-H. Su, R.-F. Huang, and C.-W. Wu, "A processor-based built-in selfrepair design for embedded memories," in Proc. 12th Asian Test Symp., Nov. 2003, pp. 366–371.
12. 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. IEEE Int. Test Conf., Oct. 2006, pp. 1–9.
13. T.-W. Tseng, J.-F. Li, and C.-C. Hsu, "ReBISR: A reconfigurable built-in self-repair scheme for random access memories in SOCs," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 18, no. 6, pp. 921–932, Jun. 2010.
14. T.-W. Tseng and J.-F. Li, "A shared parallel built-in self-repair scheme for random access memories in SOCs," in Proc. IEEE Int. Test Conf., Oct. 2008, pp. 1–9.
15. T.-W. Tseng, J.-F. Li, and C.-S. Hou, "A built-in method to repair SoC RAMs in parallel," IEEE Des. Test Comput., vol. 27, no. 6, pp. 46–57, Nov./Dec. 2010.
16. S.-K. Lu, Z.-Y. Wang, Y.-M. Tsai, and J.-L. Chen, "Efficient built-in selfrepair techniques for multiple repairable embedded RAMs," IEEE Trans. Comput.-Aided Design Integr. Circuits Syst., vol. 31, no. 4, pp. 620–629, Apr. 2012.
17. M. Miyazaki, T. Yoneda, and H. Fujiwara, "A memory grouping method for sharing memory BIST logic," in Proc. Asia South Pacific Design Autom. Conf., Jan. 2006, pp. 671–676.
18. S.-Y. Kuo and W. K. Fuchs, "Efficient spare allocation for reconfigurable arrays," IEEE Des. Test Comput., vol. 4, no. 1, pp. 24–31, Feb. 1987.
19. K. Pagiamtzis and A. Sheikholeslami, "Content-addressable memory (CAM) circuits and architectures: A tutorial and survey," IEEE J. Solid-State Circuits, vol. 41, no. 3, pp. 712–727, Mar. 2006.
20. H.-Y. Lin, F.-M. Yeh, and S.-Y. Kuo, "An efficient algorithm for spare allocation problems," IEEE Trans. Rel., vol. 55, no. 2, pp. 369–378, Jun. 2006.
21. C.-L. Wey and F. Lombardi, "On the repair of redundant RAM's," IEEE Trans. Comput.-Aided Design Integr. Circuits Syst., vol. 6, no. 2, pp. 222–231, Mar. 1987.
22. W. K. Huang, Y.-N. Shen, and F. Lombardi, "New approaches for the repairs of memories with redundancy by row/column deletion for yield enhancement," IEEE Trans. Comput.-Aided Design Integr. Circuits Syst., vol. 9, no. 3, pp. 323–328, Mar. 1990.