International Conference on Innovative Applications of Computational Intelligence on Power, Energy and Controls with their Impact on Humanity (CIPECH14) 28 & 29 November 2014

Memory Yield and Repair Rate Improvement Scheme Using Built in Self Repair Techniques

Jyotika
Centre for Development of Advanced Computing
Mohali, India.
jyotika23_1990@yahoo.in

Balwinder Singh
Centre for Development of Advanced Computing
Mohali, India.
balwinder@cdac.in

***Abstract*—Memory is an important part of every computing system. In SOC, 90 to 92% of the total chip area is covered by embedded memories (ITRS 2009) and that means memory density is higher than the logic density. Therefore testing and diagnosis of memories are important issues in the SOCs. Yield of memory is affected by the faults present in memory which also affects the yield of SOC. Built in self-repair techniques are used to repair the embedded memories. Built in self-repair techniques are used for the better yield of the system by using various techniques like 1-D Redundancy and 2-D Redundancy. Test, Redundancy analysis, Repair delivery are the three basic steps for the memory repair. A built in redundancy algorithms (BIRA) are used to implement built in self-repair (BISR).**

***Keywords—Built in redundancy analysis (BIRA), built in self-repair (BISR), built in self-test (BIST), and embedded memories*.**

I. INTRODUCTION

The system knowledge and expertise brings hardware and software components together on a single chip in a way to give complete solution and is termed as system on chip (SOC).In general, SOC design may incorporates Intellectual Property (IP) components such as programmable processor, on chip memory, audio video controllers, modems, graphics controller etc. implemented in hardware. The increasing demand of the application world puts an immense thrust on SOC technology so that more and more functionality can be integrated on single chip. The memory cells are more likely to faults and defects due to the increasing density of memory cells. Thus, embedded memories usually prevail the yield of SOC chips. Only detecting the faults in SOC is not enough, therefore for the memory repair, both repair and diagnosis algorithms are in challenge. Built-in Redundancy Analysis (BIRA) is used for the improvement of yield of embedded memories. Erstwhile, Automatic Test Equipment is used for testing and repairing the memories. During the testing and repairing process, the ATE would perform the test algorithm and observe the response of memory chip to perform the Redundancy Analysis (RA) for the replacement of faulty cells. After this process, the replacement information is burned into fuses which are used to reconfigure the defected memory as a fault free. Combination of BIST and BIRA as Built-in Self-Repair (BISR) is the trend of yield improvement of embedded memory in SOC system. As the memory increases, the number of spare lines is increased to guarantee the desired yield of memory devices. For this repair rate should be optimized. Built in self-repair is used to detect the faulty memory locations in the memory and repair those memory locations. The embedded memories dominate the yield of the SOC chips so by repairing the memory from faults, yield of memory can be enhanced. Redundancy repair is the efficient method of repairing memory. The ways to improve memories are as follows:

Redundant rows or columns [10]: By using this approach, spare rows or columns are added into the memory array. One of the redundant rows/columns is used to replace the defected row/column. It can be implemented easily.

Redundant rows and redundant columns [10]: By using this approach, both redundant rows and columns are added into the memory array. When a faulty cell is detected, use a redundant row or a redundant column to replace the defected cell. This approach can work efficiently than above approach. The main drawback of this approach is that the optimal redundancy allocation problem becomes NP-complete [7]. Although many heuristic algorithms have been proposed to solve this problem, it is still difficult to develop on-chip implementations for these algorithms.

II. BUILT IN SELF REPAIR (BISR)

Embedded random access memories are repaired by using built in self-repair techniques. The important portion of the chip is occupied by the memory cores. The memory cores predominate the manufacturing yield of the chip. BISR approach consists of BIST module, a BIRA module and a MUX module. The block diagram of BISR is as shown in figure 2.

Input Multiplexer is used to select signals depending on even if it is working in normal/test mode. Output of multiplexer is given to the memory.

Memory is used as circuit under test. At particular memory location data is written on the address location.

BIST Controller is used generate test patterns for RAMs which is under test. Fault in a RAM is detected by the BIST circuit and then the faulty information is given to the BIRA (Built in Redundancy Analysis) circuit March Tests can be used for the detection of faults. The main goal of testing

algorithm is to get the maximum fault coverage and minimum testing time. A March test consists of a fixed series of March elements. A March element is a finite series of operations applied to each cell in the memory before going on to the next cell, according to a specific address order; i.e., ascending, descending, or irrelevant.

March LR Algorithm: To verify the whether the given memory is faulty or not, it is essential to carry out a series of read write operations to the memory cell. March LR algorithm is superior to all others faults in terms of fault coverage and test time. It is one of the efficient algorithms in terms of fault coverage and minimum test time. This algorithm covers all the faults like stuck at faults, transition faults, and address decoder faults along with more coupling faults**.**

The March LR algorithm is as given below**:**

{↕(w0); ↓(r0,w1); ↑(r1,w0,r0,w1); ↑(r1,w0); ↑(r0,w1,r1,w0); ↑(r0)}

Six March (M1, M2, M3, M4, M5, and M6) elements are used. Finite state Machine is used for design controller. FSM is as shown in fig. 1.

The March LR algorithm consists of the following steps:

1. Write 0s to all locations initial in upward direction or downward direction.

2. Read 0 and write1 in downward direction.

3. Read 1, write 0, read 0 write 1 in upward direction.

4. Read 1, write 0 in upward direction

5. Read 0, write 1, read 0, write 1 in upward direction.

6. Read 0 in upward direction.

March LR is more superior to the existing March tests because it is able to detect linked faults consisting of an arbitrary number of simple faults. It is one of the efficient algorithms in terms of fault coverage and minimum test time.

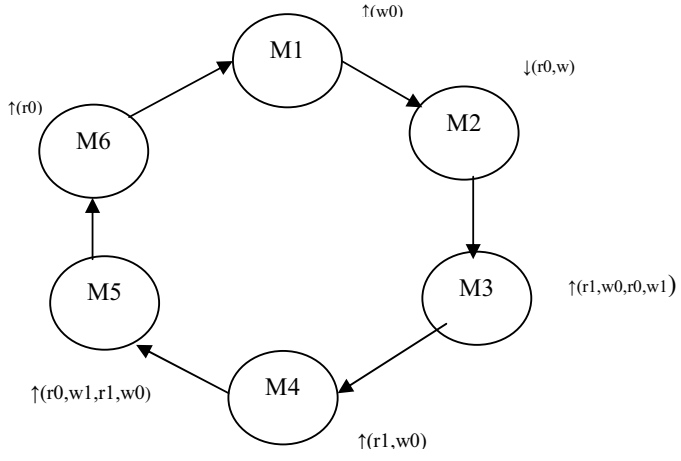

Fig.1. Finite State Machine for March LR Algorithm

Redundant Logic Array acts as a spare memory. The existing faulty locations are compared with the normal input address in the normal mode and it uses spare memory locations for read and writes operations. If there is no match then for read write operations it will use original memory.

Output multiplexer the value from the redundant memory is selected by the output multiplexer based on whether the memory is faulty or not.

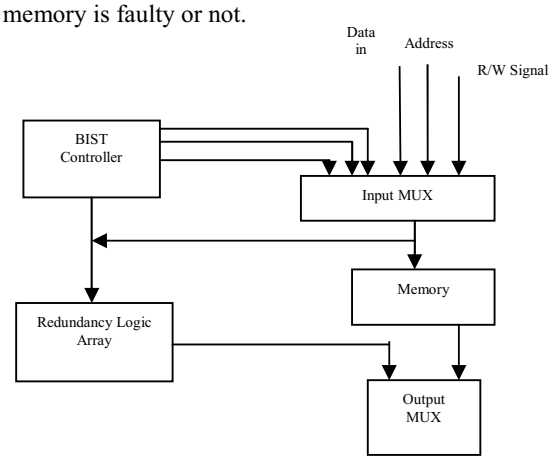

Fig.2. Block Diagram of BISR

III. BISR ARCHITECTURE

It consists of BIST, BIRA and memory Wrapper .To test the memory BIST is used and to allocate redundancy BIRA is used. To detect the faults in the main memory and spare memory BIST is used. The Proposed RA algorithm is used for redundancy allocation; this is performed by the BIRA circuit. To switch between normal mode and test/repair mode, test wrapper is used. Fig. 3 shows the architecture of the BISR.

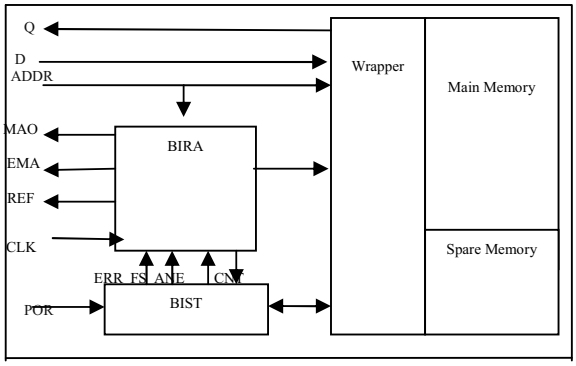

Fig.3. Architecture of BISR Scheme

In the test/repair mode, the memory is accessed by the BIST module while in normal mode; there is a comparison of address field of spare words with the incoming address. In this mode output selection is done from the main or spare memory. If there is a match, the data field of the redundant word is used along with the faulty memory location for reading and writing

data. The output multiplexer of Redundant Array Logic then ensures that in case of a match, the redundant word data field is selected over the data read out of the faulty location in case of a read signal.

The BISR procedure is shown in Fig.4. BISR technique is on power BISR. When power is on, the BIST tests the memory. When the fault is detected, BIST gives all the information to BIRA (Built in Redundancy Analysis) and allocation of spare rows and columns is done. The ERR (error) signal and FS (fault syndrome) signal are used by the BIRA to identify or mark the defective row and defective column. After allocating redundancy the main memory test is done.

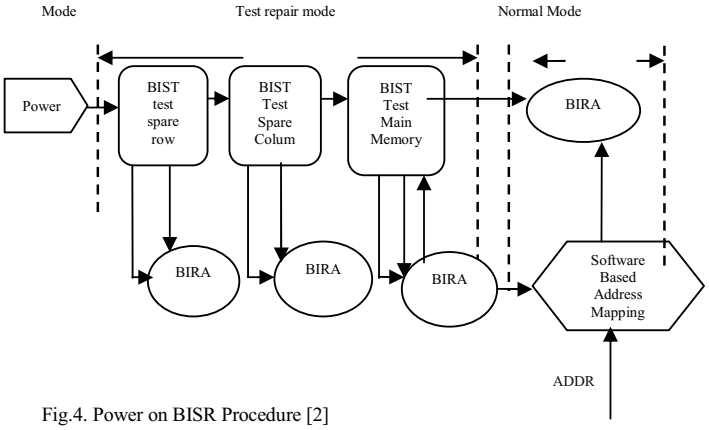

Fig.4. Power on BISR Procedure [2]

At the time of redundancy allocation, if there is a fault in a row and if there is no spare for allocation then Built in Redundancy analysis module transfers the faulty address through the EMA(export mask address) and MAO(mask address output) signals.

When the redundancy allocation is completed, the REF (repair end flag) signal takes into account and then BIRA will operate in normal mode.

IV REDUNDANCY ANALYSIS

To avoid yield loss, spare rows and columns of storage cells are added to repair the faulty cells to get improved yield of memory. The redundancy algorithm can be 1-D redundancy algorithm or 2-D redundancy algorithm. In this analysis to recover the faulty cells, choose minimum number of spare rows and columns.

Repair Rules

- The first rule is that if there are multiple faults in a row; repair the faulty row by spare row.
- The second rule is that, if there are multiple faults with the same column address and different row addresses within a segment, the last detected faulty sub word should be repaired with an available spare row The faulty sub word will be repaired with an available spare row. In this case, when the memory accesses cells in the overlapped region in the functional mode, the address remapper gives priority to the spare row.

Redundancy Analysis Algorithm

RA algorithm consists of two phases: Must repair phase and final repair phase.

Must Repair Phase: It identifies the faulty lines or faults that must be repaired by spare row and/or spare column.

Final Repair Phase: In this phase, row first, column first, fault driven algorithm are used.

Repair Algorithm:

In this algorithm, after the memory completes, then the bitmap of faulty addresses are compared with the original addresses to find the row/column faults.

1) BIST detect the faults, if the fault is detected, faulty addresses are stored and it will continue until the end of MARCH LR Algorithm completes.
2) BIRA compares the faulty addresses that are store in bitmap.
3) If the spare row/column is available, then the entire faulty row/column is replaced by spare row/column.
4) Check whether the faulty addresses are repaired.

V REPAIR RATE ANALYSIS

The repair rate is defined as the ratio of the number of repaired memories to the number of defective memories. Repair rate depends on the number of redundancies available on the chip, which is the fixed value. More redundancy means higher repair rate [5] [6].

$$\text{Repair Rate} = \frac{\text{Number of repaired memories}}{\text{Number of faulty memories}}$$

For the improvement of memory yield, redundancy techniques is used, which is further increases the repair rate. For achieving higher repair rate three techniques are there which are as follows:

(i) Conversion: When a single cell fault is repaired by a spare row (column), if a newly detected fault shares a column (row) address with the single cell fault and there is an available spare column (row), both faults are repaired by a spare column (row) instead of a spare row (column) [7].

(ii) Swap: When a fault is detected, if it is not repaired by pre-allocated spare lines and there is no available spare line, it can be repaired by swapping a spare row and a spare column [7].

(iii) Inclusion: If all faults repaired by a spare line are covered by other spare lines, the spare line is no longer required for fault repair [7].

These three techniques are used for achieving higher repair rate. In this, 15 faults are injected deliberately and only one fault is not detected and repaired by the techniques. The repair rate comes is 91%. For the optimization or enhancement of repair rate, other linked faults are injected in memory. Then according to conversion, swap and inclusion techniques repair rate can be optimized and it is about 93%.

VI SIMULATION RESULTS

The memory designed is of size 256x8 bits, the ram array has 256 elements each having 8 bits. In this the faults injected is bitwise and the number of faults injected in memory are 15. Synthesis process has performed using Xilinx 12.4 project navigator tools for synthesis the complied HDL design codes into gate level schematics and the technology mapping has chosen in this paper was spartran3E. The repair rate after repairing the memory using formula then it comes about 91% and by using different techniques the repair rate is enhanced by 93%. Simulation result of injection faults in memory and detecting faults using March LR Algorithm and repairing the memory faults using spare rows and columns are as shown in this section. The device utilization report of modules is shown in Table I.

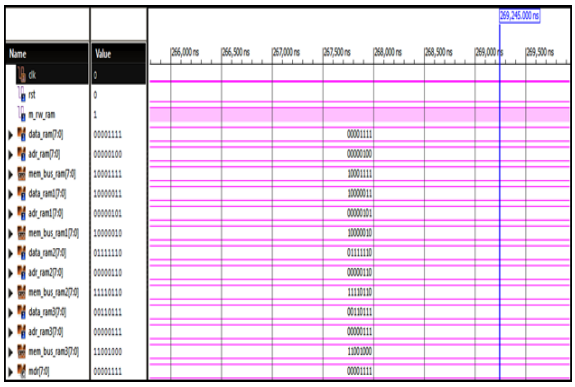

Fig.5. Simulation waveform of injecting multiple faults at memory locations.

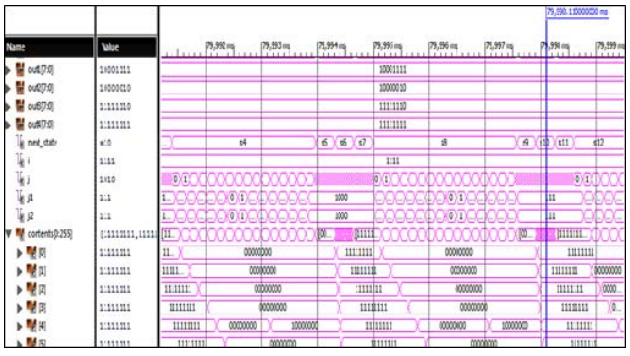

Fig.6. Simulation waveform of detecting multiple faults using March LR Algorithm

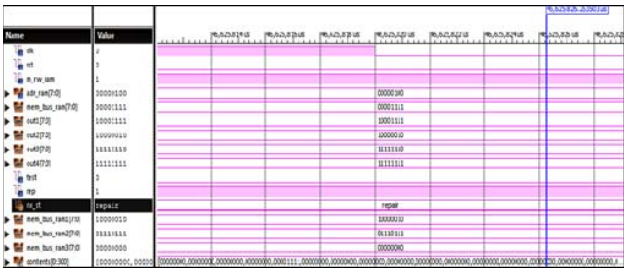

Fig.7. Simulation waveform of repairing multiple faults using redundancy algorithms

TABLE I. DEVICE UTILIZATION REPORT OF MODULES

Components	***Injecting Faults***	***Detecting Faults(BIST)***	***Repairing of Faults(BISR)***
Number of Slices	7078(47%)	6577(44%)	6851 (46%)
Number of slice Flip Flop	2202(7%)	2270(7%)	2286(7%)
Number of LUT	13700(46%)	12812(43%)	1398(45%)
Number of IOBs	99	100	117

In previous paper the faults injected are not more than 10.The repair rate obtained is about 83.3% by using three spare rows and columns. In this paper the number of faults injected is bitwise and are more than 10 in number. The repair rate calculated is 91%. After optimizing the repair rate by using three techniques, the repair rate obtained is about 93%.

VII CONCLUSION

Built in Self Repair is used to enhance the yield of embedded memories. Yield can be increased by using redundancy techniques. 2-D redundancy technique produce optimal repair rate and cannot guarantee a high repair rate. So repair rate can be optimized. In this, number of faults can be injected into the memory and can be detected by the March LR Algorithm and repaired by spare row and column that is assigned in memory. Conversion, swap, inclusion are the three techniques that are used to increase the repair rate. Repair rate is optimized to 93%. By optimizing the repair rate memory yield can be improved. This paper deals with the injection and testing of faults and repairing memory from the faults.

REFERENCES

[1] International Technology Roadmap for Semiconductors (ITRS), 2011. [Online]. Available: http://public.itrs.net

[2] Jin-Fu Li, Jen-Chieh Yeh, Rei-Fu Huang, and Cheng-Wen Wu, “A Built-In Self-Repair Design for RAMs with 2-D

Redundancy", *IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 13, no. 6,* June 2005.

[3] W. K. Huang, Y. H. Shen, and F. Lombrardi, "New approaches for the repairs of memories with redundancy by row/column deletion for yield enhancement," Computer-Aided Design of Integrated Circuits and Systems, *IEEE Transactions on 9.3 (1990): 323-328.*

[4] S.-K. Lu, Y.-C. Tsai, C.-H. Hsu, K.-H. Wang and C.-W. Wu, "Efficient Built-In Redundancy Analysis for Embedded Memories with 2-D Redundancy", *IEEE Trans. VLSI Systems,* vol. 14, no. 1, pp. 34-42, 2006.

[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. VLSI Systems*, vol. 17, no. 12, pp. 1665-1678, 2009.

[6] "High-efficiency BIRA for embedded memories with a high repair rate and low area overhead", *Journal of Semiconductor Technology and Science*, vol.12, no.3, September, 2012.

[7] S.-Y. Kuo and. K. Fuchs, "Efficient spare allocation in reconfigurable arrays," *IEEE Design Test*, vol. 4, pp. 24–31, 1987.

[8] K.Bala Souri, K.Hima Bindu, K.V.Ramana Rao, "A Built-in-Self-Repair Scheme for Random Access Memories with 2-D Redundancy", *International Journal of Soft Computing and Engineering (IJSCE),* vol.1, November 2011.

[9] Tsu-Wei Tseng, Jin-Fu Li and Chih-Chiang Hsu, "ReBISR: A Reconfigurable Built-In Self-Repair Scheme for Random Access Memories in SOCs", *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 18, no. 6, June 2010.

[10] Benso, Alfredo,"An on-line BIST RAM architecture with self-repair capabilities." *Reliability, IEEE Transactions on 51.1* (2002): 123-128.

[11] S.-K. Lu, Y.-C. Tsai, C.-H. Hsu, K.-H. Wang and C.-W. Wu, "Efficient Built-In Redundancy Analysis for Embedded Memories with 2-D Redundancy", *IEEE Transaction VLSI Systems*, vol. 14, no. 1, Pp. 34-42, 2006.

[12] P.Ravinder, N.Uma Rani, "Design and Implementation of Built in Self-Test and Repair", *International Journal of Engineering Research and Applications,* vol.1, pp.778-785

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

[14] L.-T. Wang, C.-W. Wu, and X. Wen, VLSI Test Principles and Architectures, Morgan Kaufmann, 2006.

[15] Joohwan Lee, Kihyun Park, Sungho Kang"High-efficiency BIRA for embedded memories with a high repair rate and low area overhead", *Journal of Semiconductor Technology And Science,* vol.12, no.3, September, 2012.

[16] 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. VLSI Systems*, vol. 17, no. 12, pp.1665-1678, 2009.

[17] R.Rajsuman, "Design and test of large embedded memories: an overview*," IEEE Design Test Computer.*, vol. 18, no. 3, pp. 16-27, May 2001.

[18] Shekar Babu M, Sumanth Kumar Reddy, S V V Sateesh, "Built In Self Repair for Multiple Rams with Different Redundancies in a SOC," *International Journal of Computer Applications*, vol. 24, no. 8, June 2011.

[19] Singh Balwinder, Sukhleen Bindra Narang, and Arun Khosla. "Modeling and Simulation of Efficient March Algorithm for Memory Testing", *Contemporary Computing. Springer Berlin Heidelberg,* pp-96-107, 2010.