Built-In Self-Repair for Multiple RAMs with Different Redundancies in a SOC

Shekar Babu M
VIT University
Vellore, India

Sumanth Kumar Reddy
VIT University
Vellore, India

S V V Sateesh
VIT University
Vellore, India

ABSTRACT

As RAM is major component in present day SOC, by improving the yield of RAM improves the yield of SOC. So the repairable memories play a vital role in improving the yield of chip. This paper presents the efficient Reconfigurable Built-in Self Repair (Re BISR) circuit along with 2D redundancies (spare row/column) and spare cells. Since most of faults are single cell fault, the area of spare is effectively utilized by replacing defected cell with spare cell. This in turn increases repair rate. The proposed repair circuit is Reconfigurable for less area, used to repair multiple memories with different in size and redundancy. The experimental results show that proposed ReBISR circuit reduces the area and increases the yield of the memory.

Keywords

ReBISR, BIRA, BISR, MBIST

1. INTRODUCTION

The advancement in IC technology increases integration of memories. As SOC size is shrinking, the major area on SOC is occupied by embedded memories. Thus memories in chip will decide the yield of the SOC. Increase in yield of memories in turn increase the yield of SOC. In [1], SOC yield increases from 2% to 10% by improving the memory yield from 5% to 10%. The techniques used for yield improvements in memories are Built In Self Test (BIST) and BISR. Many algorithms are proposed for spare allocation for defected memories. The redundancy is of 1D (only spare row or column) [3-4] or 2D (spare row and spare column) [5-7]. The redundancy analysis (RA) algorithm for 1D is simple compare to 2D.

In [2], a reconfigurable MBIST architecture is proposed by adding some data processing unit and address processing unit. Where the data width and address width of MBIST is variable according to the RAM which is being tested. In [3], this scheme uses multiple bank cache like memory for repairs. Remapping scheme and redirecting read/write request from faulty part to spare elements. In [4], Single spare column is used to replace multiple single cell faults by selectively decoding row address bits and sending control signals for multiplexers. This scheme is efficient only for single defects in single.

The repair rate and area cost of the Built In Redundancy Analysis (BIRA) is mainly depends on the redundancy organization. The redundancy organization memory is divided into various segments [5]. In which spare row and columns are used differently. Spare rows are used to replace entire row in the memory and the columns are divided in several spare column groups. Here the access time and area cost is induced due to additional multiplexers.

However different redundancy organization will lead to different area cost and repair rate. Since most of the memory faults are single cells, spare words [6] are very efficient in reducing area. In this, the spare rows and spare columns are virtually divided into spare row blocks and spare column group blocks.

SOC contains different RAMs with different address and data widths. If every RAM in SOC contains different repair circuit, area of the repair circuit increases. In order to reduce the area cost of repair circuit, reconfigurable repair circuit is used in which a single repair circuit is used for repairing multiple memories with different size and redundancies [7].

The below paper is organized as follows: Section 2 describes redundancy organization of the memory and block diagram of BISR, ReBISR and its working. Section 3 describes the proposed repair algorithm. Section 4 summaries the experimental results. Section 5 contains conclusion.

2. OVERVIEW OF PROPOSED SCHEME

2.1 Redundancy organization

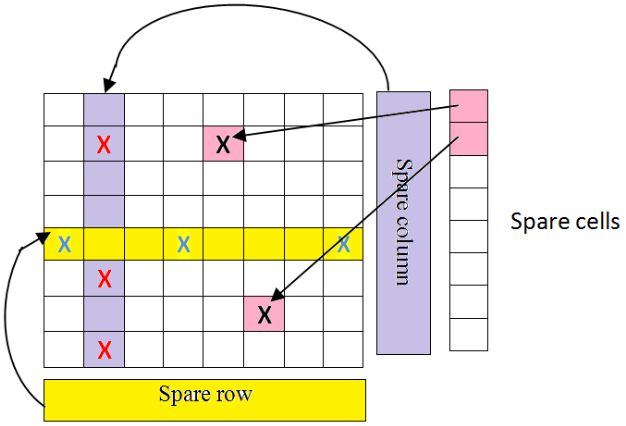


Fig 1: The proposed Redundancy organization

The fig 1 shows an example of 8x8 main memory modules along with 1 spare row, 1 spare column and spare cells. Since most of the memory faults are single cell defects, here spare cells are used for better utilization of spare elements. The row/column having multiple defects is remapped with corresponding spare row/column. The single defects in the main memory are remapped with spare cells. By this redundancy organization the area of spare is efficiently utilized.

2.2 Architecture BISR circuit

The BISR circuit mainly consists of MBIST and BIRA. Main memory contains multiplexers and repair registers. Multiplexers are used to switch between test/repair mode and normal read/write mode, where as repair registers are used to store faulty address and remapped address. MBIST is used to test the main memory as well as spare by March C- algorithm. The algorithm is implemented using FSM. The faulty information from MBIST is transferred to the bitmap, which is used to store faulty addresses.

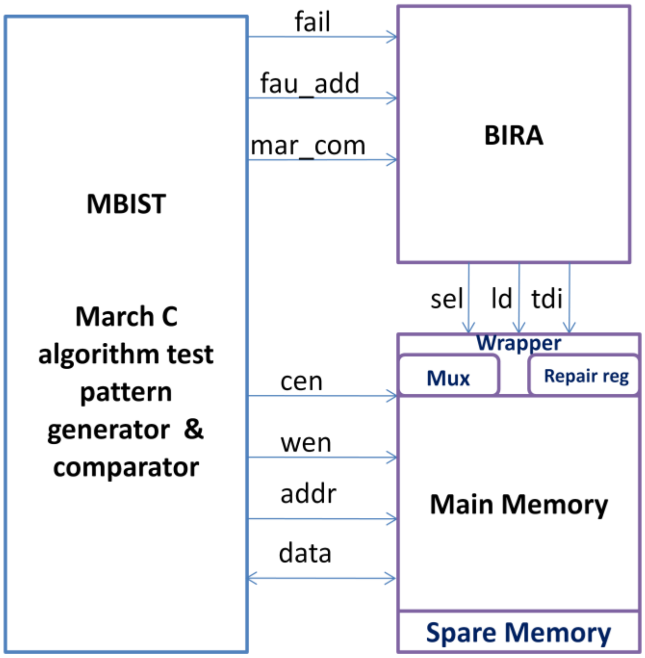


Fig 2: Block diagram of proposed BISR

The BIRA uses the proposed algorithm to remap the defected with spare elements. This fault signature is stored in the repair register of the RAM. In normal read/write mode of main memory the faulty address are bypassed by the new remapped address.

2.3 Architecture of ReBISR

The reconfigurable ReBISR module consists of both the reconfigurable MBIST and BIRA. The MBIST used here is having variable address width and variable data width. Depending on the RAM is being tested, the MBIST is reconfigured corresponding RAM. In the similar manner the BIRA is remapped to corresponding RAM.

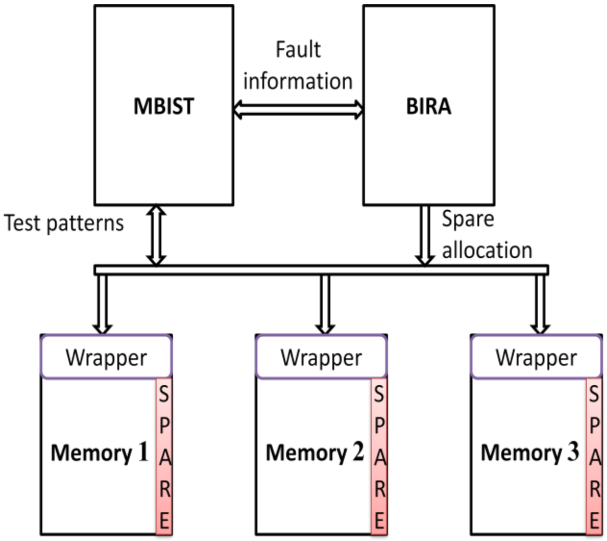


Fig 3: Architecture of ReBISR

3. REPAIR ALGORITHM

Since 70% of defects occurred in the memory are single cell faults. This proposed scheme mainly concentrated on the single cell faults. The RAMs are initially manufactured with certain spare row/column and spare cells. The algorithm uses the bitmap to store the faulty information from the MBIST. After completion of memory test, the algorithm compares the faulty addresses in the bitmap to find the multiple row/column faults and single cell faults.

1. MBIST enable: if fault detects, store the faulty address and continue until the end of MARCH complete.

2. BIRA is used to compare the faulty addresses (row and column address) stored in bitmap.

3. In every repetition of row/column address, corresponding row/column counter is incremented.

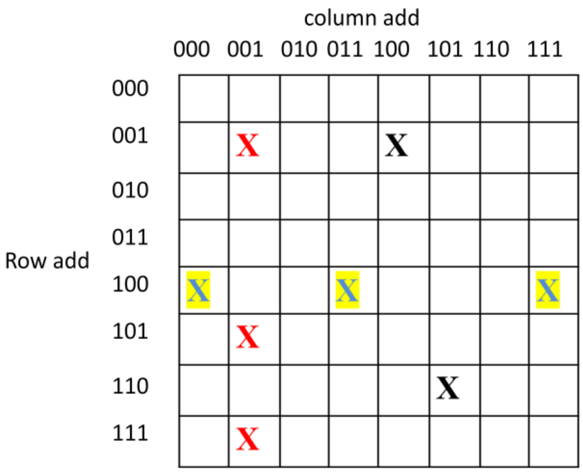
4. If row counter>column counter & spare row is available, then the entire faulty row is replaced by spare row and the row address is stored.

5. Else if row counter<column counter & spare column is available, then the entire column is replaced with spare column and the column address is stored.

6. Else if row counter=column counter or if no spare row/column, the address is remapped with spare cell.

7. Check whether the all the faulty addresses are repaired, if so then stop.

The above algorithm is explained in more detail by taking an example with 6 bit address, in which 3bits are column address and 3 bits are row address.



Row add \ column add	000	001	010	011	100	101	110	111
000								
001		X			X			
010								
011								
100	X			X				X
101		X						
110						X		
111		X						

Fig 4: 8x8 bits RAM representing the defects

In the fig 4, X represents defected cells. By testing the above memory, the faulty address is stored in bitmap shown in Table 1.

Table 1. BITMAP showing faulty addresses

	Row address	Column address
1	001	001
2	001	100
3	100	000
4	100	011
5	100	111
6	101	001
7	110	101
8	111	001

After completing MBIST operation, BIRA compares the faulty addresses. From the above table, faulty address 1 is compared with the other faulty address. Row addresses 001 repeats twice but column address 001 is repeated three times, so the entire column of 001 is remapped with spare column and faulty addresses 1, 6 and 8 is marked as repaired. Now faulty address 2 is no repetition so the spare cell is used to repair and 2 is marked as repaired. Faulty address 3, the row address 100 repeats three times, so the entire row is replaced with spare row and faulty addresses 3, 4 and 5 are marked as repaired. Now only remaining faulty address is 7, which is replaced by spare cell.

4. RESULTS COMPARISION

The proposed ReBISR is simulated several times. The results in fig 5 & fig 6 show the improved yield and repair rate of the memory by repair circuit. The repair rate shown below depends on the fault pattern on the memory.

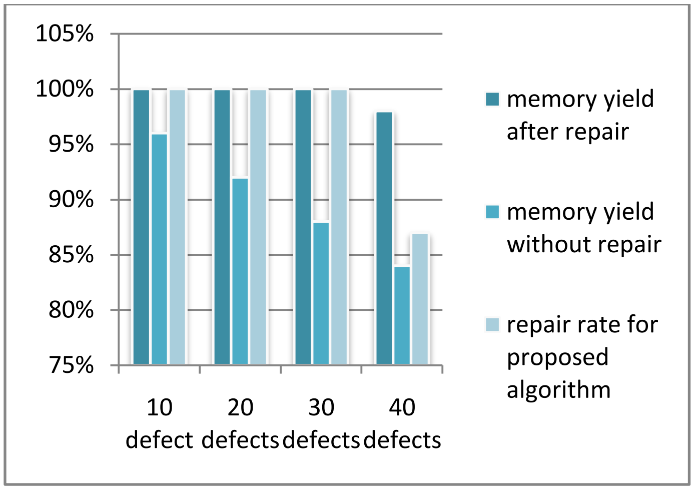


Fig 5: comparison of memory yields with and without repair circuit and repair rate for 16x16 bits memory along with spare row, spare column and 16 spare cells.

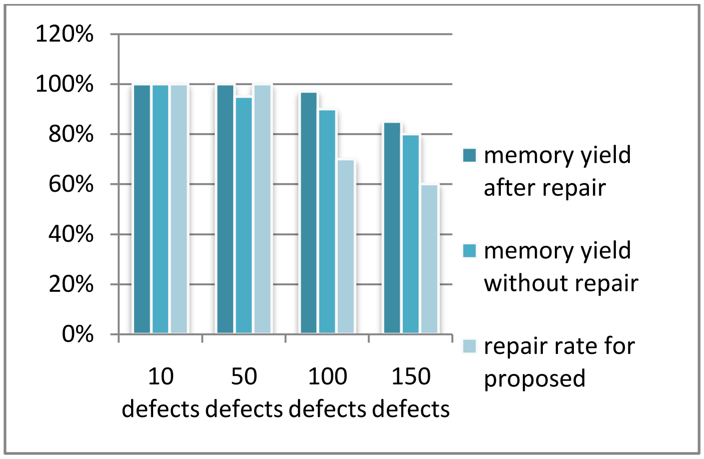


Fig 6: comparison of memory yields with and without repair circuit and repair rate for 32x32 bits memory along with spare row, spare column and 32 spare cells.

Table 2, 3, 4, 5 and 6 shows that the Area and power of individual repair circuit for 16x16, 32x32 and 64x64 bits memories and reconfigurable repair circuit respectively. Table 6 shows the reduction of area and power of ReBISR as compare to individual repair circuit. The tool used for used for synthesis is Cadence-RTL compiler.

Table 2. Area and Power for BISR circuit for RAM1: 16X16 bits

	Area	Power	Time period
MBIST	116 cells 3476 μm^2	113855.4 nw	3400 ps
BIRA	1233 cells 36730 μm^2	1734973.3 nw	11400 ps

Table 3. Area and Power for BISR circuit for RAM2: 32X32 bits

	Area	Power	Time period
MBIST	129 cells 3889µm^2	179592.2nw	3400ps
BIRA	1359 cells 41517µm^2	1893482nw	11400ps

TABLE 4. Area and Power for BISR circuit for RAM3: 64X64 bits

	Area	Power	Time period
MBIST	133 cells 4241µm^2	185378.8 nw	3400ps
BIRA	1490 cells 46300µm^2	2082923.6 nw	11400ps

Table 5. Reconfiguarable BISR for RAM1, RAM2 & RAM3

	Area	Power	Time period
MBIST	144 cells 4371µm^2	192190.9nw	3700ps
BIRA	1660 cells 49733µm^2	2269316.3nw	12000ps

TABLE 6. Comparision of REBISR with indivisual BISR circuit for RAM1, RAM2 & RAM3

	% of Area reduced	% of Power reduced
MBIST	62.33%	59.66%
BIRA	60.00%	60.26%

5. CONCLUSION

A reconfigurable built in self repair circuit is efficiently implemented in this paper. The redundancy algorithm for 2D redundancy allocation is introduced. The BIRA is reconfigurable to support for multiple memories with different redundancies. So the area of the repair circuit is effectively reduced, i.e. nearly 60% of repair circuit area is reduced compare to individual repair circuits for 16x16, 32x32 and 64x64 bits memory modules. Another important factor is power is also reduced by 60%, shown the Table 6. The yield of memory plays major role in SOC designs, the proposed ReBISR effectively increases compare to traditional yield.

6. REFERENCES

[1] R. Rajsuman, “Design and test of large embedded memories: an overview,” IEEE, vol. 18, no. 3, May 2001.

[2] Yu-Ming Jia, Quan-Lin Rao and Chun He “A Memory Built-In Self-Test Architecture for memories different in size,” IEEE 2009.

[3] Kiamal Pekmestzi, Nicholas Axelos, Isidoros Sideris and Nicolaos Moshopoulos, “A BISR Architecture for Embedded Memories,” 14th IEEE International On-Line Testing Symposium 2008.

[4] Muhammad Tauseef Rab, Asad Amin Bawa, and Nur A. Touba, “Improving Memory Repair by Selective Row Partitioning,” 24th IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems, 2009.

[5] Jin-Fu Li, Rei-Fu Huang, Jen-Chieh Yeh 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, June 2005.

[6] Chun-Lin Yang, Shyue-Kung Lu, and Han-Wen Lin, “Efficient BISR Techniques for Word-Oriented Embedded Memories with Hierarchical Redundancy” 2006.

[7] Jin-Fu Li, Member, IEEE, Tsu-Wei Tseng 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, JUNE 2010.

[8] 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) Syst., vol. 15, no. 10, pp. 1135–1143, Oct. 2007.

[9] T.-W. Tseng, C.-H. Wu, Y.-J. Huang, J.-F. Li, A. Pao, K. Chiu, and E. Chen, “A built-in self-repair scheme for multiport RAMs,” in Proc. IEEE VLSI Test Symp. (VTS), Berkeley, CA, May 2007, pp. 355–360.

[10] A. Benso, S. Chiusano, G. D. Natale, and P. Prinetto, “An on-line BIST RAM architecture with self-repair capabilities,” IEEE Trans. Reliab. vol. 51, no. 1, pp. 123–128, Mar. 2002.

[11] C.-D. Huang, T.-W. Tseng, and J.-F. Li, “An infrastructure IP for repairing multiple RAMs in SOCs,” in Proc. IEEE Int. Symp. VLSI Des., Autom., Test (VLSI-DAT), Hsinchu, Apr. 2006, pp. 163–166.

[12] J.-F. Li and C.-W. Wu, “Memory fault diagnosis by syndrome compression,” in Proc. Conf. Des., Autom., Test Eur. (DATE),Munich, Germany, Mar. 2001, pp. 97–101.

[13] 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), Santa Clara, CA, Oct. 2006,pp. 1–8, Paper 30.2.

[14] M. Nicolaidis, N. Achouri, and S. Boutobza, “Optimal reconfiguration functions for column or data-bit built-in self-repair,” in Proc. Conf. Des., Autom., Test Eur. (DATE), Munich, Germany, Mar. 2003, pp. 590–595.