ISSN:0975-9646

Implementation of Optimized Reconfigurable Built-in Self-Repair Scheme for RAMs in SOCs

Venkatesh S, Laxmi Prasanna Rani M

Department Of ECE, MVGR College of Engineering, Vizianagaram, Andhrapradesh

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. Built-in self-repair (BISR) technique has been widely used to repair embedded random access memories (RAMs). This paper presents a reconfigurable BISR (ReBISR) scheme for repairing RAMs with different sizes and redundancy organizations. An efficient redundancy analysis algorithm is proposed to allocate redundancies of defective RAMs. In the ReBISR, a reconfigurable built-in redundancy analysis (Re-BIRA) circuit is designed to perform the redundancy algorithm for various RAMs. Also, an adaptively reconfigurable fusing methodology is proposed to reduce the repair setup time when the RAMs are operated in normal mode. The experimental results show that proposed ReBISR circuit reduces the area and increases the yield of the memory.**

Keywords: Built-in self-test, built-in self-repair, built-in redundancy-analysis, reconfigurable built in self repair, semiconductor memory.

I. 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). The redundancy analysis (RA) algorithm for 1D is simple compare to 2D. In [2], a reconfigurable BIST architecture is proposed by adding some data processing unit and address processing unit. Where the data width and address width of BIST 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. 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 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.

II. TYPICAL MEMORY BISR ARCHITECTURE

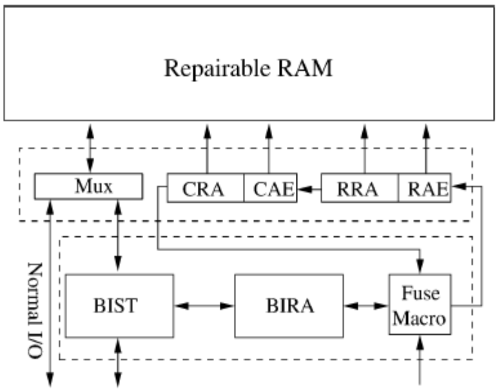

Fig.1. Typical BISR scheme for embedded RAMs.

Fig. 1 shows the block diagram of a typical BISR scheme for a RAM, which consists of four major components.

1) *Repairable RAM*: A RAM with redundancies and reconfiguration circuit. Fig. 2 depicts an example of an 8*8 bit-oriented RAM with 1 spare row and 1 spare column. If a spare row is allocated to replace a defective row, then the row address of the defective row is called row repair address (RRA). Then a decoder decodes the RRA into control signals for switching row multiplexers to skip the defective row if the row address enable (RAE) signal is asserted. The reconfiguration of the defective column and the spare column is performed in a similar way, i.e., give a column repair addresses (CRA) and assert the column address enable signal to repair the defective column using the spare column.

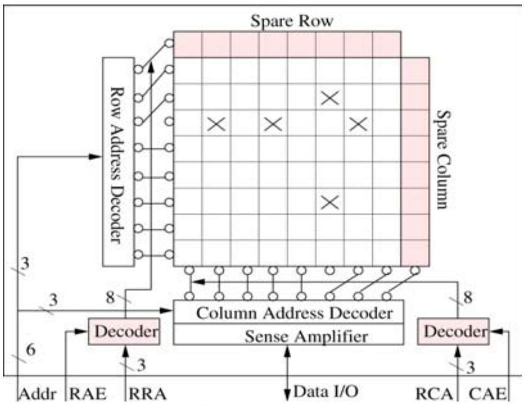

Fig.2. Conceptual diagram of an 8*8 bit-oriented repairable RAM with one spare row and one spare column.

2) Built-in Self-Test (BIST) Circuit. It can generate test patterns for RAMs under test. While a fault in a defective RAM is detected by the BIST circuit, the faulty information is sent to the BIRA circuit.

3) BIRA Circuit. It collects the faulty information sent from the BIST circuit and allocates redundancies according to the collected faulty information using the implemented redundancy analysis algorithm.

4) Fuse Macro. It stores repair signatures of RAMs under test. The fuses of the fuse box can be implemented in different technologies, e.g., laser blown fuses, electronic-programmable fuses, etc. The fuse register is the transportation interface between the fuse box and the repair register in the repairable RAM. If a fault is detected, then the fault information is stored in the BIRA circuit. Then, the BIRA circuit allocates redundancies to replace defective elements. As soon as the repair process is completed, the repair signatures are blown in the fuse box. Subsequently, the repair signatures are loaded into the fuse register first and then are shifted to the repair registers (i.e., registers in the wrappers for storing RRA, RAE, CRA, and CAE data) in normal operation mode. Finally, the repairable RAM can be operated correctly.

III. MEMORY REBISR SCHEME

As Fig. 1 shows, if each RAM in an SOC has individual BISR circuit, then the total area of BISR circuits for the RAMs in the SOC is large, since an SOC usually has many RAMs. To reduce the area cost of BISR circuits, we propose a ReBISR scheme for RAMs in an SOC. A ReBISR circuit can be shared by multiple RAMs such that the total area cost of BISR circuits in an SOC can be drastically reduced.

A. Architecture of the ReBISR Scheme

Fig. 3 shows the 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.

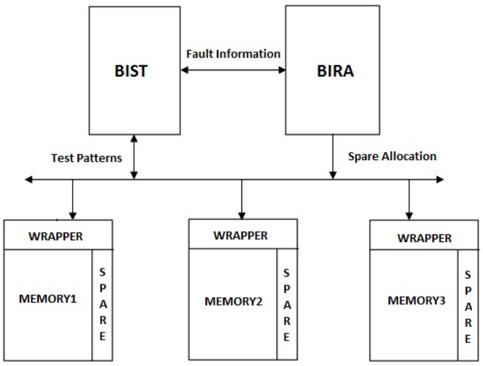

Fig.3. Architecture of ReBISR

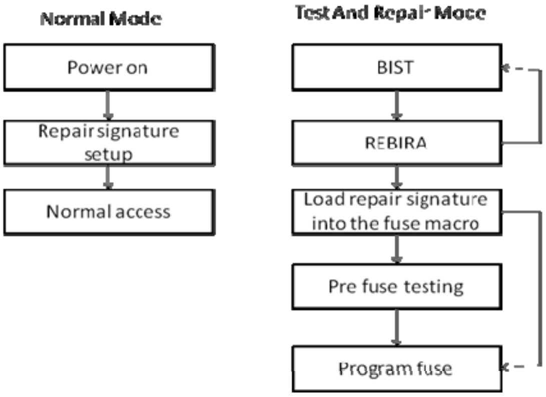

Fig. 4. (a) Repair process during normal operation phase. (b) Repair process during test and repair phase.

Fig. 4(a) shows the repair process during the normal operation phase. Note that if a soft repair strategy (only registers are used to store repair signatures [5]) is used in the ReBISR scheme, then this phase is not needed. In the life time of the repaired RAMs, once the power of the RAMs is turned on, the repair signatures stored in the fuses are loaded into the Fuse Register of the Fuse Macro by asserting the signal LD. Then the repair signatures in the Fuse Register are shifted into the repair registers in Wrappers through the Fuse input (FI) and Fuse output (FO). The time for setting the repair signatures is called *repair setup time*. Once the repair signatures have been loaded into the repair registers, the RAMs can be accessed normally. In an SOC, multiple ReBISR circuits are allowed. Thus, RAMs in the SOC can be divided into groups and each group is served by one ReBISR circuit. The grouping should consider the issues of power, repair time, area cost, etc. Then the IEEE 1500 test wrappers [6] can be used to control the ReBISR circuits. Also, all Fuse Macros can be connected through the TDI and TDO. Therefore, the chip-level IEEE 1149.1 [7] can communicate with the Fuse Macros through TDI and TDO.

B. Redundancy Analysis Algorithm:

In this section, we describe the proposed redundancy analysis algorithm—*range-checking first algorithm* (RCFA). The RCFA can support two types of redundancy configurations—spare row/spare column and spare row/spare

IO. Also, the RCFA can be used for a RAM with local redundancies (i.e., each RAM block has individual redundancies) or for a RAM with global redundancies and local redundancies. For brevity, we only present the RCFA algorithm for a RAM with more complicated redundancies, i.e., a RAM with global redundancies and local redundancies. But, the RCFA can easily be modified to fit a RAM with local redundancies.

Subsequently, more detail redundancy allocation procedure of the RCFA is described. First, more notations used in the description of RCFA is defined:

1) N_{RE} —number of row entries in the local bitmap.

2) N_{CE} —number of column entries in each sub-bitmap.

3) R_{MF} —the row with maximal number of faulty bits in the local bitmap.

4) C_{MF} —the column with maximal number of faulty bits in the sub-bitmap.

Major steps of the redundancy allocation procedure of RCFA are shown in below Algorithm.

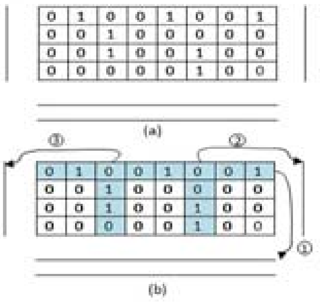

Fig.5. Example of defective ram

Step 1: Run BIST; pause and jump to Step 2 when it detects a fault.

Step 2: Check whether the detected fault has been repaired. If so, go to Step 1. Otherwise, go to Step3.

Step 3: Check whether the Hamming syndrome has multiple 1s (The Hamming syndrome is defined as the modulo-2 sum of the expected (fault-free) data output vector and the output vector from the RAM under test, for word-oriented Rams) If so, repair the faulty word with an available spare row. Otherwise, store the corresponding Hamming syndrome in the local bitmap.

Step 4: Check whether the local bitmap is full. If so, go to the next step. Otherwise, go to Step 1.

Step 5: If $N_{FRE} > N_{FCE}$ allocate an available spare column to replace the C_{MF} (if available spare column is exhausted, allocate an available spare row to replace the); else if $N_{FRE} < N_{FCE}$ allocate an available spare row to replace the R_{MF} (if available spare row is exhausted, allocate an available spare column to replace the); else $N_{FRE} = N_{FCE}$ replace the faulty element with the largest number of faulty bits with a corresponding available spare element. If spare elements are exhausted, the RAM is un-repairable.

Step 6: Check whether the BIST is done. If so, go to the next step. Otherwise, go to Step 1 when the local bitmap is not full; go to Step 5 when the local bitmap is still full.

Step 7: Check whether the local bitmap is empty. If so, export the repaired addresses and then stop. Otherwise, go to Step 5.

Fig. 5 shows the allocation of spare rows and columns by fault rows and columns according to the RCFA algorithm.

C. Typical Fusing Methodology

Typically, the repair signatures of multiple RAMs in an SOC are stored in a Fuse Macro. Once the SOC is used in normal mode, the repair signatures stored in the Fuse Macro are uploaded into the repair registers of the corresponding RAMs. As

Fig. 6 shows two typical methods can be used to access the fuse information: serial access and parallel access methods. If the serial access method is used for loading the repair signatures into the repair registers of the RAMs under repair, then the repair setup time of the RAMs is long. But, if the parallel access method is considered, then the routing between the fuse macro and the RAMs is very complicated. Also, the timing of the signal wires results in another critical issue. Thus, most of fusing methodologies use the serial access method to load repair signatures [9]. Therefore, we consider the fusing methodology with serial access method for loading repair signatures as a typical fusing methodology in this paper.

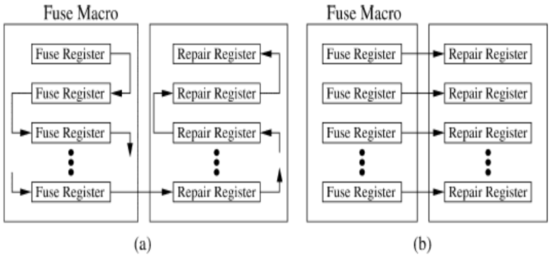

Fig.6. (a) Serial access of the fuse information. (b) Parallel access of the fuse information.

D. Design Of The Rebira

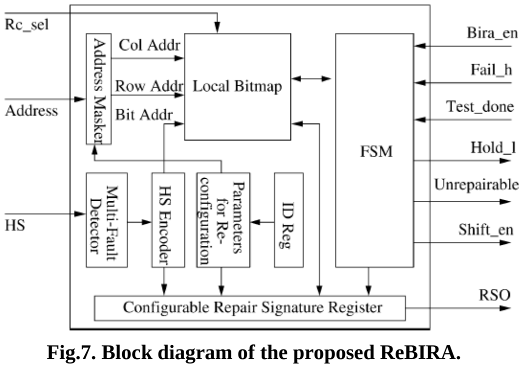

Fig.7. Block diagram of the proposed ReBIRA.

Fig. 7 shows the simplified block diagram of the proposed ReBIRA, which mainly consists of a multi-fault detector (MFD), a hamming syndrome (HS) encoder, a local bitmap, an FSM, an address masker (AM), and a configurable repair signature register (CRSR). The MFD can detect whether the Hamming syndrome of the detected fault has multiple 1s or not. If so, the corresponding faulty row is repaired by an available spare row. Otherwise, the HS only has one 1 and the HS is encoded into a bit address by the HS encoder. Then the bit address is stored in the local bitmap. If the local bitmap is full, then the redundancy allocation is executed. The FSM major realizes the redundancy allocation procedure of RCFA. The identification register (ID Reg) and Parameters are configuration data for the AM, local bitmap, and CRSR for each RAM, since different RAMs have different lengths of addresses and different number of redundancies. The AM can set the corresponding registers of the local bitmap to fit the configuration of the RAM under test according to the parameters. The length of CRSR for a RAM is associated to the number of redundancies and the number of bits of row address and column address. Therefore, the CRSR also should be configured as different lengths to fit the configuration of the RAM under repair.

IV. EXPERMENTAL RESULTS

Fig.8 shows the output waveforms of the BIRA module. In this BIRA module it shows the total bits in the bitmap in those bits bit 1 shows the fault information and bit 0 shows fault free information and it also shows the address of the fault row or column according to the given algorithm.

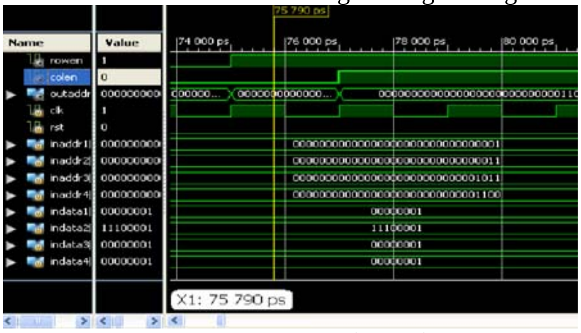

Fig.8.output waveforms of BIRA

Fig.9 shows the output waveforms of the ReBISR module. If the signal Bira_en en is asserted, the ReBIRA monitors the signal Fail h which is from the BIST circuitry. If the BIST detects a fault, it asserts the corresponding bit of Fail h signal to 1 and exports the fault information (Address and HS) to the ReBIRA. Also, the ReBIRA sets the Hold l to 1 and the BIST is paused simultaneously.

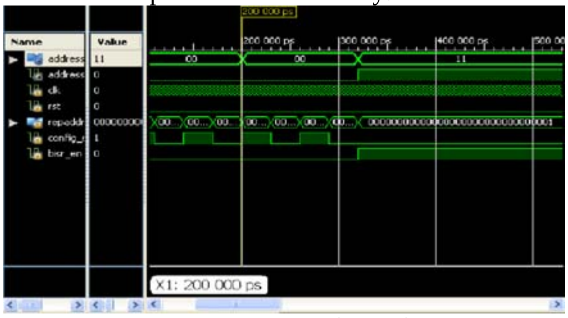

Fig.9.output waveforms of ReBISR

Fig.10 shows the output waveforms of the fuse register module. The output register fuse register module stores the repair signature of multiple RAMS in SOC.

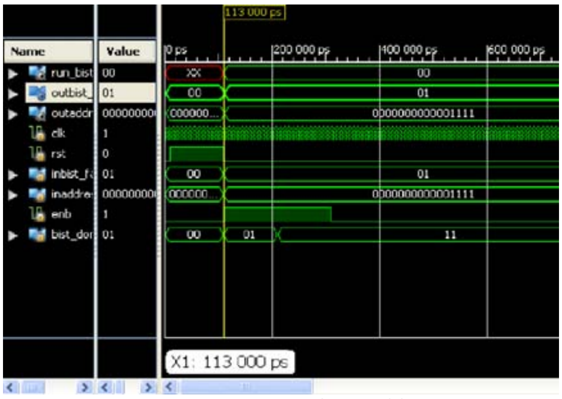

Fig.10.output waveforms of fuse register

V. CONCLUSION

A reconfigurable BISR scheme for repairing multiple repairable RAMs with different sizes and redundancy configurations has been presented in this paper. An efficient BIRA algorithm for 2-D redundancy allocation has also been introduced. The yield of memory plays major role in SOC designs, the proposed ReBISR effectively increases compare to traditional yield.

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] V. Schober, S. Paul, and O. Picot, "Memory built-in self-repair using redundant words," in *Proc. Int. Test Conf. (ITC)*, Baltimore, MD, Oct.
2001, pp. 995–1001.

[6] *IEEE 1500 Standard for Embedded Core Test (SECT)*, IEEE 1500, 2005. [Online]. Available: http://grouper.ieee.org/groups/1500/

[7] *IEEE 1149.1 Standard Test Access Port and Boundary-Scan Architecture*,
IEEE 1149.1, May 1990.

[8] T.-W. Tseng and J.-F. Li, "A shared parallel built-in self-repair scheme for random access memories in SOCs," in *Proc. Int. Test Conf. (ITC)*,
Santa Clara, CA, Oct. 2008, pp. 1–9, Paper 25.2.

[9] B. Cowan, O. Fransworth, P. Jakobsen, S. Oakland, M. R. Ouellette, and D. L. Wheater, "Onchip repair and an ATE independent fusing methodology," in *Proc. Int. Test Conf. (ITC)*, Oct. 2002, pp. 178–186.