ISSN (ONLINE): 2454-9762
ISSN (PRINT): 2454-9762
Available online at www.ijarmate.com

International Journal of Advanced Research in Management, Architecture, Technology and Engineering (IJARMATE)
Vol. 1, Issue 4, November 2015

IMPLEMENTATION OF REBISR FOR SRAM USED IN EMBEDDED CORE

S.Brenija Stanley[1], T.Abisha[2], R.Steffi[3]
Assistant Professor, Vins Christian Women's College of Engineering, Nagercoil [1,2,3]

Abstract **- 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 (ReBIRA) 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 BIRA developed with an architectural change. By the change in BIRA architecture will improve the process of Re-BISR by means of communication between the RAM and Re-BISR.A BIST architecture developed with an enhancement in its architecture. I implement this BIST in Xilinx. I am expecting that the experimented results will improve the speed and performance of the Re-BISR. The area cost of the ReBISR is very small, which is only about 2.7% for four RAMs (one 4 Kbit RAM, one 16 Kbit RAM, one 128 Kbit RAM, and one 512 Kbit RAM). Moreover, the time overhead of redundancy analysis is very small. Fuse Macro methodology is used to reduce the repair setup time.**

Keywords: **Built in self-repair, Built in self- test, Built in redundancy Analysis.**

I. INTRODUCTION

Built in Self Test or BIST, is the technique of designing additional Hardware and software features into integrated circuits to allow them to perform self-testing, i.e., testing of their own operation (functionally, parametrically, or both) using their own circuits, thereby reducing dependence on an external automated test equipment (ATE).BIST is a Design-for-Testability (DFT) technique, because it makes the electrical testing of a chip easier, faster, more efficient, and less costly. The concept of BIST is applicable to just about any kind of circuit, so its implementation can vary as widely as the product diversity that it caters to. As an example, a common BIST approach for DRAM's includes the incorporation onto the chip of additional circuits for pattern generation, timing, mode selection, and go-/no-go diagnostic test.

Today's System-on-Chip typically embeds memory IP cores with very large aggregate bit count per SOC. This trend requires using dedicated resources to increase memory yield, while containing test &repair cost and minimizing time-to-volume. This paper summarizes the evolution of such yield optimization resources, compares their trade-offs, and concentrates on on-chip Infrastructure IP. To maximize the repair efficiency, this Infrastructure IP need to leverage the memory design knowledge and the process failure data. The ideal solution is to integrate the memory IP and its Infrastructure IP into a single composite IP that yields itself effectively.

A word oriented memory Built-In Self-Repair methodology is described without modifying the memory module. Faulty addresses and its data will be stored in the redundancy logic immediately after its detection during test. Fuse boxes can be connected via scan registers to the redundancy logic.

This paper presents a built-in self-test (BIST) scheme, which consists of a flexible pattern generator and a practical on-macro two-dimensional redundancy analyzer, for GHz embedded SRAMs. In order to meet the system requirements and to detect a wide variety of faults or performance degradation resulting from recent technology advances, the microcode-based pattern generator can generate flexible patterns. A practical new repair algorithm for the Finite State Machine (FSM)-based on macro redundancy analyzer is also presented. It can be implemented with simple hardware and can show fairly good performance compared with conventional software- based algorithms.

An innovative self-test and self-repair technique supports Built-in Self-test and Built-in Self-repair of large embedded RAM arrays with spare rows and columns. The technique generates and analyzes the required failure bitmap information on the fly during self-test and then automatically repairs and verifies the repaired RAM arrays.

An embedded processor-based built-in self-repair (BISR) design for embedded memories is proposed. In the proposed design we reuse the embedded processor that can be found on almost every system-on-chip (SOC) product, in addition to many distinct features. By reusing the embedded processor, the controller and redundancy analysis circuit of a typical BISR design can be removed. Also, the test algorithm and redundancy analysis/allocation algorithm are easily programmable, greatly increasing the design flexibility. The area overhead of the proposed BISR scheme is low, since only the memory wrapper needs to be realized explicitly. As memories are designed very tightly to the limits of the technology they are more prone to failures than logic. As memories are designed very rightly to the limits of the technology they are more prone to failures than logic. Thus, they concentrate the large majority of defects and affect circuit yield dramatically. Thus, Built-In Self-Repair is gaining significant importance. regular units.

This paper presents a built-in self-repair (BISR) scheme for semiconductor memories with two–dimensional (2-D) redundancy structures, i.e., spare rows and spare columns. The BISR design is composed of a built-in self-test module and a built-in redundancy analysis (BIRA) module. The BIRA module executes the proposed RA algorithm for RAM with a 2-D redundancy structure.

II. RECONFIGURABLE BISR FOR RANDOM ACCESS MEMORY IN SOCs

Embedded random access memory (RAM) is one key component in modern complex system-on-chip (SOC) designs. Typically, many RAMs with various sizes are included in an SOC, and they occupy a significant portion of the chip area. Furthermore, RAMs are subject to aggressive design rules, such that they are more prone to manufacturing defects. That is, RAMs have more serious problems of yield and reliability than any other embedded cores in an SOC. Keeping the RAM cores at a reasonable yield level is thus vital for SOC products. Built-in self-repair (BISR) technique has been shown to improve the RAM yield efficiently. For example, the work shows that the BISR circuit can improve the RAM yield from 5% to 20%,such that the net SOC yield increase can range from 2% to 10%. Therefore, the BISR technique is apromising and popular solution for RAM yield improvement.

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

A. Repairable RAM

A RAM with redundancies and reconfiguration circuit Fig.1 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.

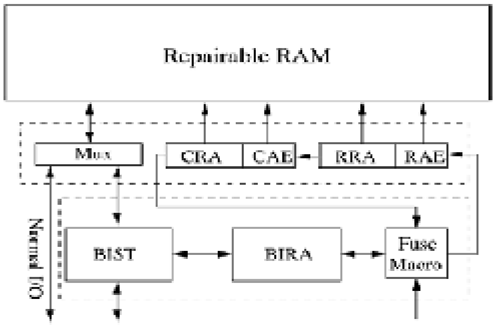

Fig.1 Typical BISR scheme for a RAM

B. 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.

C. 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.

D. Fuse Macro

It stores repair signatures of RAMs under test. Fig.2 shows the conceptual block diagram of a typical implementation of fuse macro. 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.

As Fig.2 shows, the overall RAM BISR flow is roughly described as follows. Firstly, the BIST tests the repairable RAM.

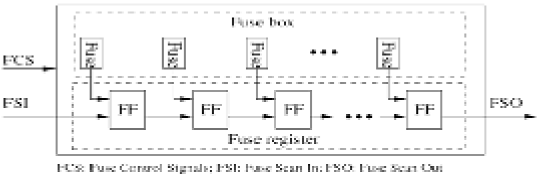
Fuse box
FCS
FSI
Fuse
FF
FSO
Fuse register
FCS: Fuse Control Signals; FSI: Fuse Scan In; FSO: Fuse Scan Out

Fig.2 Conceptual block diagram of a fuse macro

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. ARCHITECTURE DESIGN

A. Architecture of the REBISR Scheme

Fig.3 shows the simplified block diagram of the proposed ReBISR scheme for repairing multiple repairable RAMs in an SOC, where detail control signals for the ReBISR circuit is not shown. The Wrapper of a RAM under test consists of multiplexers, a test pattern generator (TPG), and repair registers. The multiplexers switch the RAM between test/repair mode and normal mode. The TPG generates desired test patterns according to the given command from the test controller (CTR). The repair registers store the repair signatures. The CTR coordinates the operations of the TPG and the ReBIRA circuit. The Fuse Macro consists of the fuse and the fuse register. The number of bits of the fuse, the fuse register and the repair register is the same. Different technologies can be used to implement the fuse, e.g., the laser-blown fuse, the programmable electronic fuse, etc. If the laser-blown fuse is used, then the ReBISR only can repair the target RAMs one time. In our ReBISR scheme, moreover, the repairable RAMs can be equipped with one of the following two redundancy organizations: 1) spare rows and spare columns and 2) spare rows and spare IOs.

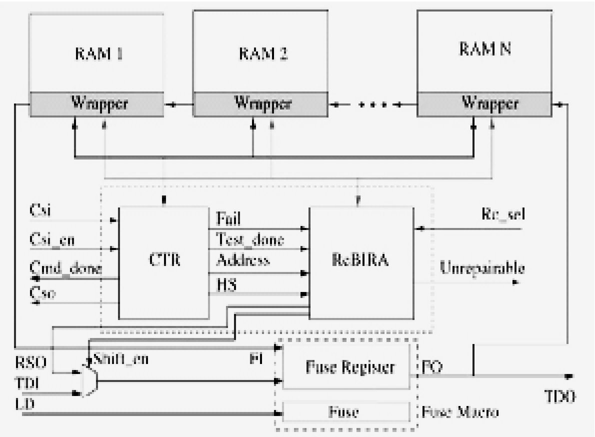

Fig.3 Simplified block diagram of the proposed ReBISR scheme for multiple RAMS

Fig.4 shows the repair process of the proposed ReBISR scheme during test and repair phase. If the BIST detects a fault, then the fault information is exported to the ReBIRA circuitry and then the ReBIRA performs redundancy allocation on the fly using the rules of the implemented redundancy algorithm (the proposed redundancy algorithm will be described in the next subsection). The ReBIRA allocating redundancy on the fly means that the redundancy allocation process and the BIST process are performed concurrently. After the ReBIRA allocates a redundancy to repair a corresponding faulty row or column, the local bitmap is updated and the BIST is resumed. This process is iterated until the test and repair process is completed. When the BIST and BIRA are completed, the repair signatures stored in the Repair Signature Register are shifted into the Fuse Register of Fuse Macro through the RSO (repair signature output). Before programming the fuses, the user can first load repair signatures into the repair registers in the Wrappers. Then the BIST is used to test the repaired RAMs again. This is called *pre-fuse testing*. Subsequently, if the pre-fuse testing is completed and no fault is detected, the repair signatures in the Fuse Macro are exported to the fuse-programming equipment or circuit through TDO. Then the repair signatures can be programmed into the fuses. Note that the pre-fuse testing can be optional.

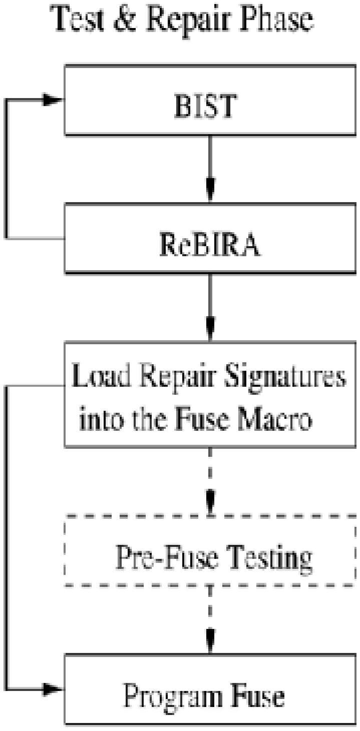

Fig.4 Repair process during test and repair phase

Fig.5 shows the repair process during the normal operation phase. Note that if a soft repair strategy (only registers are used to store repair signatures) is used in the ReBISR scheme, then this phase is not needed. 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*.

When the chip is used in field, the fuse control signal (FCS) is asserted and the repair signature stored in the Fuse of each Fuse Macro is updated to the corresponding Fuse Register. Then the repair signature in each Fuse Register is shifted into the repair registers of the Wrappers of memories in corresponding group through the FI and FO. Thus the memory repair is completed.

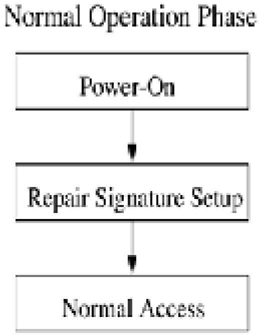

Fig.5 Repair process during normal operation phase

IV. RESULTS AND DISCUSSIONS

A. HDL Simulation using Modelsim

Mentor Graphics was the first to combine single kernel simulator (SKS) technology with a unified debug environment for Verilog, VHDL, and SystemC. The combination of industry-leading, native SKS performance with the best integrated debug and analysis environment make ModelSim the simulator of choice for both ASIC and FPGA design. The best standards and platform support in the industry make it easy to adopt in the majority of process and tool flows.

B. Verilog Module

Verilog is a hardware description language (HDL). A hardware description language is a language used to describe a digital system: for example, a network switches, a microprocessor or a memory or a simple flip-flop. This just means that, by using a HDL, one can describe any (digital) hardware at any level. One can describe a simple Flip flop as that in the above figure, as well as a complicated design having 1 million gates. Verilog is one of the HDL languages available in the industry for hardware designing. It allows us to design a Digital design at Behavior Level, Register Transfer Level (RTL), Gate level and at switch level.

C. Xilinx

Xilinx is known for inventing the field programmable gate array (FPGA) and as the first semiconductor company with a fables manufacturing model. Xilinx designs, develops and markets programmable logic products including integrated circuits (ICs), software design tools, predefined system functions delivered as intellectual property (IP) cores, design services, customer training, field engineering and technical support Xilinx sells.

The ISE Design Suite is the central electronic design automation (EDA) product family sold by Xilinx. The ISE Design Suite features include design entry and synthesis supporting Verilog or VHDL, place-and-route (PAR), completed verification and debug using ChipScope Pro tools, and creation of the bit files that are used to configure the chip.

Xilinx's Altera FPGA kit supports the embedded PowerPC 405 and 440 cores and the Micro blaze core. Xilinx's System Generator for DSP implements DSP designs on Xilinx FPGAs.

D.Outputs

Multiplexer is actually a switch which switches the one of the multiple inputs to the output with the use of a selector pin. In Fig.5 the d pin is given as input here and switches the same input as output

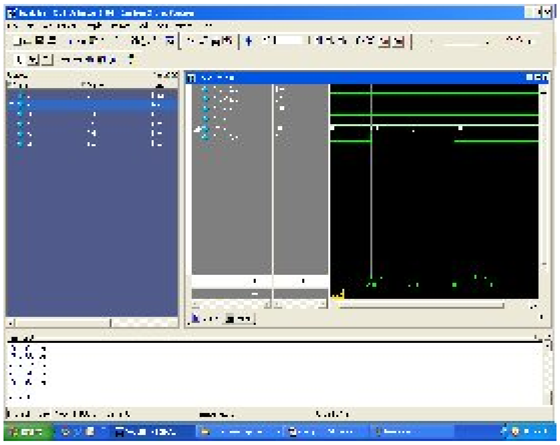

Fig.5 Multiplexer Output Waveform

In Fig.6 A d flip flop also called data flip flop gives the same input at output with a clock delay. After ring counter used, it just rotates d input like a ring. This represent a mechanism for interconnection

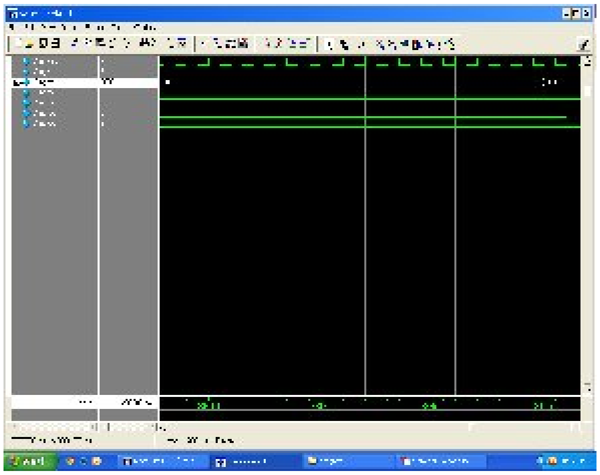

Fig.6 Fuse Macro Output Waveform

In Fig.7.Initial block which occurs only at the start of simulation at 0ns.This is to read from memory means every time any change occurs in any of the inputs d statement written below. It would execute it means if e=1 then it means if read =1 , it implies read operation if write is 1 then we have to write the a port value into the particular address of the memory if its not read/write then you are making the port b as zero.

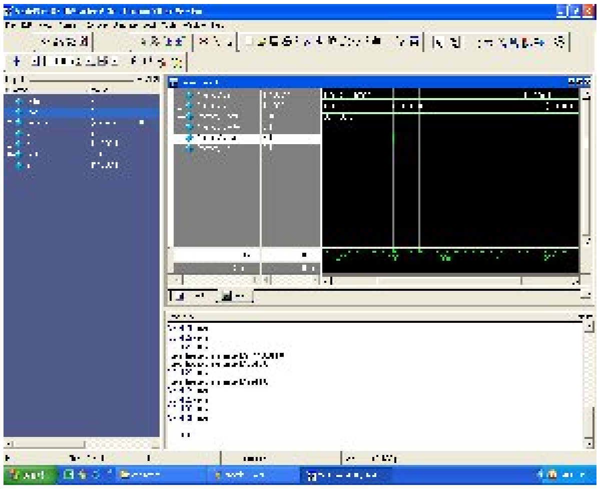

Fig.7 SRAM Output Waveform

In Fig.8,the BIST is used to detect the fault memory and after it is repair the fault detection.

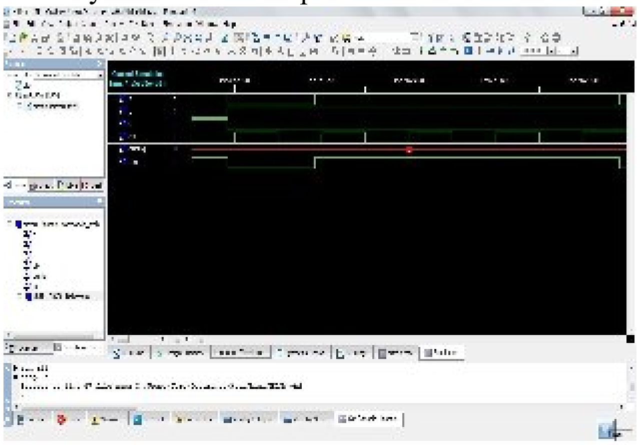

Fig.8 BIST Output Waveform

IV. CONCLUSION & FUTURE WORK

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 BIRA algorithm has been realized in a reconfigurableBIRA hardware such that it can support different RAMs. Experimental results show that the repair rate of the proposed BIRA scheme approximates to that of the exhaustive search algorithm. The area cost of the reconfigurable BISR is very small. For example, the area cost is only about 2.7% for four RAMs. Also, the ratio of the redundancy analysis time to the test time is very small. An adaptively reconfigurable fusing methodology has been proposed to reduce the repair setup time. Simulation results show that the repair setup time can be reduced.

A. Future Work

Future Enhancement plan is to use more transistors in DRAM for more speed and performance. DRAM is used for refresh the data for example addresses and its data will be stored in the redundancy logic immediately after its detection during test,if any mistakes attain during the test,DRAM can refresh the address and data. Fuse boxes can be connected via scan registers to the redundancy logic.Therefore embedded memories are commonly equipped with spare rows and columns (2D redundancy). To avoid the storage of large failure bitmaps needed by classical algorithms for offline repair analysis, existing heuristics for built-in repair analysis (BIRA) either follow very simple search strategies or restrict the search to smaller local bitmaps.

REFERENCES

1. Bhavsar, D. K. (2011) “An algorithm for row-column self-repair of RAMs and its implementation in the Alpha 21264,” in Proc. Int. Test Conf. (ITC), Atlantic City, NJ, ,pp. 311–318.

2. Kawagoe, T. Ohtani, J. Niiro, M. Ooishi, T. Hamada, M. and Hidaka, H. (2010) “A built-in self-repair analyzer (CRESTA) for embedded DRAMs,” in Proc. Int. TestConf. (ITC), pp. 567–574.

3. Kim, I. Zorian, Y.Komoriya, G. Pham, H. Higgins, F. P. and Lweandowski, J. L (2010)“Built in self repair for embedded high density SRAM,” in Proc. Int. Test Conf. (ITC), pp. 1112–1119.

4. Li, J.F. Yeh, J.C. Huang, R.F. and Wu, C.W. (2009) “A built-in self-repair design for RAMs with 2-D redundancies,” IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 13, no. 6, pp. 742–745.

5. Nakahara, S. Higeta, K. Kohno, M. Kawamura, T. andKakitani, K.(2008) “Built-in self-test for GHz embedded SRAMs using flexible pattern generator and new repair algorithm,” in Proc. Int. Test Conf.(ITC), pp. 301–310.