An Efficient Bist and Bira for Processors with Minimum Input Change (Mic) Pattern

S.Ilakkiya Priyanga (ME-VLSI DESIGN)
ECE Department, Syed Ammal Engineering College, Ramanathapuram.
Mrs.C.Priya M.E.,(Ph.D.),
Assistant Professors /ECE, Syed Ammal Engineering College, Ramanathapuram.

Abstract*: In this paper, Built -in Self-Repair approach has emerged as a promising solution for testing the memories in VLSI technology. The performance of the processors are degraded by the faulty memories. In order to detect the faults and improve the performance the Built-in self-repair (BISR) scheme is used . All memories are concurrently tested by the small dedicated built-in self-test (BIST) by means of applying the test patterns to figure out the faults in the memories. The generated pattern has large number of transitions and hence more power consumption. After all memories are tested, only faulty memories are separated, serially tested and repaired by the shared built-in redundancy analysis(BIRA) according to the sizes of memories in descending order. The proposed Minimum input change (MIC) pattern has minimum input transitions which is generated by the Johnson counter and is applied to the memories for testing and to reduce the power consumption. Thus, the fast test and repair process are performed with low area overhead. More number of memories can be tested with superior performance. Our proposed BISR has been coded in verilog HDL and simulated using xilinx12.1.*

Keyword *- Built-in redundancy analysis (BIRA), built-in self-repair (BISR), built-in self-test (BIST), Minimum Input Change (MIC) pattern.*

I INTRODUCTION

More and more system-on-chip (SOC) designs are memory dominated designs. Among various types of cores in SOCs, memory cores often are the densest part and designed with aggressive design rules. Thus memory yield usually dominates the yield of SOCs. Efficient defect tolerance techniques are imperative for large memories. Adding redundancy is one popular and widely used defect tolerance technique for memories. However, memories in SOCs usually have different sizes and types. Also, accessibility of embedded memories is usually very poor. These cause that repairing embedded memories with external equipments becomes very difficult and high cost. Therefore, built-in self-repair (BISR) technique is gaining popular for repairing embedded memories. One of key components in BISR designs is the built-in redundancy-analysis (BIRA) component.

Various BIRA schemes have been proposed. In the Alpha 21264, a redundancy analyzer for a RAM with 1_1 redundancies per memory block was implemented. To achieve optimal utilization of redundancy, a comprehensive real-time exhaustive search test and analysis (CRESTA) scheme is used to allocate redundancy. However, the hardware cost for realizing the CRESTA scheme is very high. In [3], a BIRA scheme for a RAM with divided word-line structure was reported. However, a processor-based BISR scheme uses specific instructions to construct the redundancy analysis algorithm. This results in long redundancy analysis time, since a statement of the redundancy analysis algorithm may need multiple instructions to realize it. Therefore, a time-efficient and area-efficient BISR scheme is needed to improve the yield of RAMs in SOCs economically. After the BISR circuit completes the test/repair process of a defective RAM, an electrically programmable fuse macro enables the on-chip self-reconfiguration process for skipping defective elements of the RAM operated in normal mode. Various implementations of fuse macros have been proposed Typically, multiple RAMs of an SOC share a fuse macro to store the repair signatures of the RAMs. Once the power of the SOC is turned on, the repair signatures are serially shifted to the individual repair register of each RAM. The loading time of the repair signatures is called the repair setup time of the RAMs. three BIRA algorithms for bit oriented memories are presented.

In [6], a BIRA scheme extended from the CRESTA scheme is used to support the redundancy allocation for word-oriented memories. In [7], a BIRA scheme for word oriented memories with 2D redundancy is introduced. In [8], a BIRA scheme for word-oriented memories with 2D redundancy using 1D local bitmap is reported. This scheme can execute 2D redundancy allocation with low area cost.

In this paper,Built in self repair scheme(BISR) uses the built in self test(BIST) and Built in repair analysis(BIRA). BIST is used test and detect faulty memories. It send the fault information to BIRA. BIRA is used to find the repair solutions to the faulty memories. Among hiding countermeasures, essentially distinguish strategies: one based on the randomisation of the execution of cryptographic algorithms one or more Pseudo Random Number Generators (PRNGs) are also included to generate the masks, which should be updated at each step of the datapath for a more efficient masking scheme. Prng used to generate key values to overcome the limitations of manual insertion values.

The rest of this brief introduces the BISR in section I. Next the related works and an BISR-based BIRA design are considered in Section II. Then, in Section III, the proposed is presented. Section IV and V presents a experimental Results and performance analysis to illustrate the effectiveness of the hardware security approach. Finally, the conclusions are summarized in Section V.

II. RELATED WORKS

In previous method presents an Optimized Built-In Self-Repair for Multiple Memories. A new built-in self-repair (BISR) scheme is proposed 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. The performance of the proposed BISR is located in the optimum point between the test and repair time, and the area overhead.

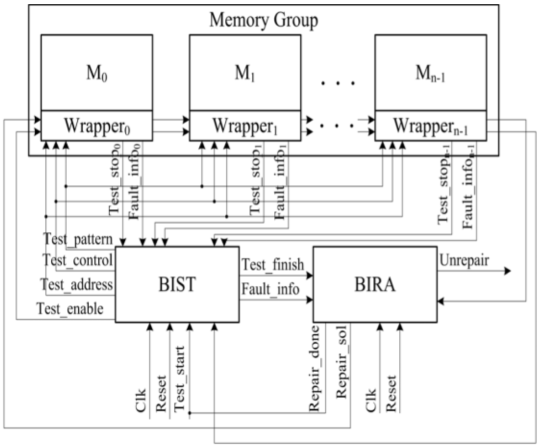

Fig 1: Block diagram of the BISR architecture

Figure 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. In this paper, a hard repair is performed using programmable electrical fuse (eFuse) schemes. The hard repair permanently repairs the faulty memory. Electrically programmable fuses are developed to perform hard repair using an eFuse and antifuse. After the repair is done, the signal Repair_done is asserted. It is connected to the port Test_start, and test is started for the next faulty memory.

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.

According to the test order, the target memory under test is tested. Whenever a fault is detected, the fault information is sent through the port Fault_infotarget. When the detected number of faults reaches the number of faults stored in the first test, the test is stopped by the signal Test_stoptarget. Then, the signal Test_ finish is activated to prevent test time from being wasted, while the test algorithm is being completed during the serial test. The test time is greatly reduced, because most faults are detected in the first few read operations of a March test.

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, and the area overhead by maintaining the optimal repair rate. In addition, the verification procedure is simply conducted through the parallel test. Therefore, the proposed BISR scheme is a solution that trades off test and repair time, and area overhead to accomplish an optimal repair rate for multiple embedded memories in the SoC.

III. PROPOSED METHODOLOGY

This paper proposes a new approach to implement a novel BISR scheme ,which can mitigate radiation-induced temporary faults by Minimum Input Change(MIC). A new Enhanced Lockstep scheme built using a pair of DUT cores is proposed and implemented on Xilinx Virtex-5 FPGA. Thus, the fast test and repair are performed with low area overhead. The test pattern generator generates the test patterns for the CUT. Examples of pattern generators are a ROM with stored patterns, a counter, and a linear feedback shift register (LFSR). A typical response analyzer is a comparator with stored responses or an LFSR used as a signature analyzer. It compacts and analyzes the test responses to determine correctness of the CUT. This is 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.

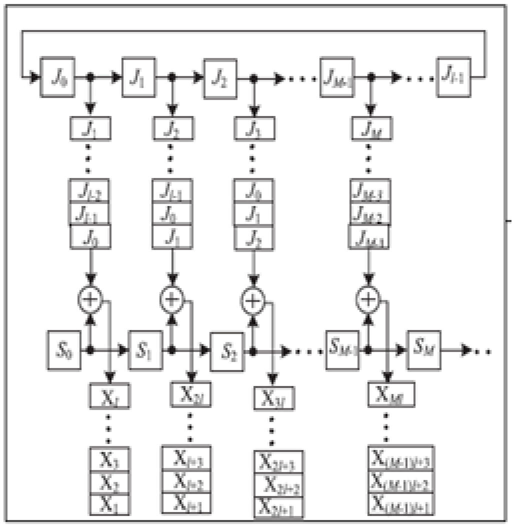

Fig 2. Johnson counter for MIC

This section develops a TPG scheme that can convert an SIC vector to unique low transition vectors for multiple scan chains. First, the MIC vector is decompressed to its multiple code words. Meanwhile, the generated code words will bit- XOR with a same seed vector in turn. Hence, a test pattern with similar test vectors will be applied to all scan chains.

This is also known as the twisted-ring counter(shown in fig,2)., is exactly the same as the ring counter except that the inverted output of the last flip-flop is connected to the input of the first flip-flop. The Johnson counter works in the following way: Take the initial state of the counter to be 000. On the first clock pulse, the inverse of the last flip-flop will be fed into the first flip-flop, producing the state 100. On the second clock pulse, since the last flip-flop is still at level 0, another 1 will be fed into the first flip-flop, giving the state 110. On the third clock pulse, the state 111 is produced. On the fourth clock pulse, the inverse of the last flip-flop, now a 0, will be shifted to the first flip-flop, giving the state 011. On the fifth and sixth clock pulse, using the same reasoning, we will get the state’s 001 and 000, which is the initial state again. Hence, this Johnson counter has six distinct states: 000, 100, 110, 111, 011 and 001, and the sequence are repeated so long as there is input pulse. Thus this is a MOD-6 Johnson counter.

For test-per-clock schemes, M segments of the CUT’s primary inputs are applied with M unique SIC vectors. The mean input transition density of the CUT is close to 1/l. For test-per-scan schemes, the

CUT’s PIs are kept unchanged during 2l2 shifting-in clock cycles, and the transitions of a Johnson codeword are not greater than 2. Therefore, the mean input transition density of the CUT during scan-in operations is less than 2/l.

IV. EXPERIMENTAL RESULTS

The proposed circuit are simulated and synthesized by using modelsim and xilinx12.1 which occurs low area than the existing. The experimental results are given in Table 1 and the simulation results of layout and the waveforms are shown in the fig.4 and fig.5. Then the RTL schematic of the proposed are shown in fig.5.

TABLE 1

s.no	Parameter	Existing	proposed
1	Slice	69	59
2	Lut	120	104

Comparison table

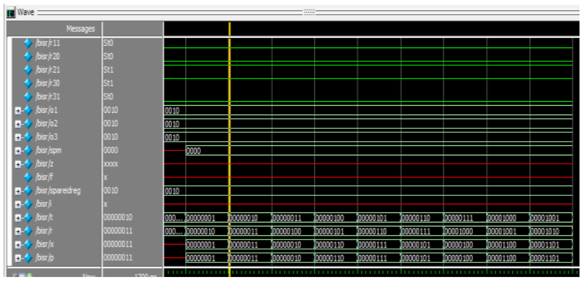

Fig.3 simulation results

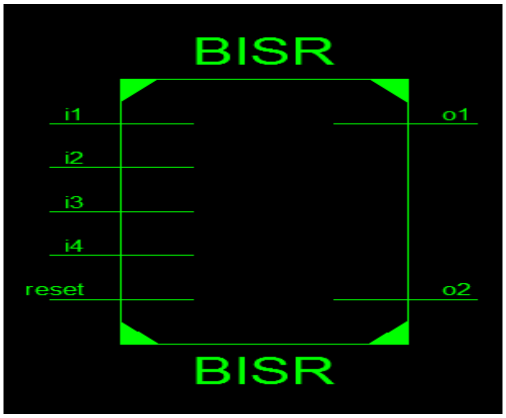

Fig.4 RTL schematic

V. PERFORMANCE ANALYSIS

The Figure given below is shown that there is a considerable reduction based on no of transistors and the performance chart has been shown below in fig.7

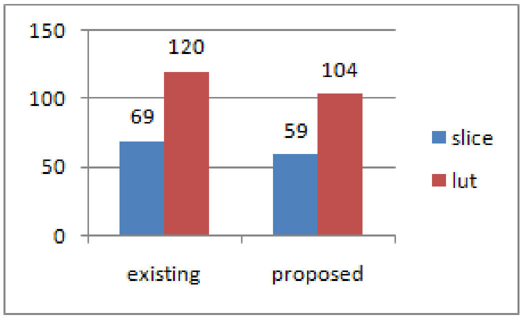

Fig.5 Performance analysis of existing and proposed

VI. CONCLUSION

A BISR technique for multiple embedded memories is proposed. To find permanent faults in memories SIC pattern as been applied to multiple memories . All memories are concurrently tested by the small dedicated BIST to figure out the faulty, the number of faults, and irreparability. . The proposed BISR scheme finds the optimum point between the test and repair time, and the area overhead by maintaining the optimal repair rate. In addition, the verification procedure is simply conducted through the parallel test. Therefore, the proposed BISR scheme is a solution that

trades off test and repair time, and area overhead to accomplish an optimal repair rate for multiple embedded memories in the SoC.

References

[1] Wooheon Kang, Changwook Lee, Hyunyul Lim, and Sungho Kang (Feb 2016) ,”Optimized built in self test for multiple memories*”IEEE transactions on very large scale integration (vlsi) systems Syst., vol. 33, no. 11*,pp. 2136–2879.

[2] Kelly A.Ockunzzi;Michael R.Ouellette;kevin w.Gorman (2015),”Optimizing delay test at the memory boundary” ,*IEEE Des. Test Comput., vol. 27, no. 6, pp. 32-54*,.

[3] W. Kang, H. Cho, J. Lee, and S. Kang,(, Nov. 2014) “A BIRA for memorieswith an optimal repair rate using spare memories for area reduction,”*IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 22, no. 11*,pp. 2336–2349

[4] W. Jeong, J. Lee, T. Han, K. Lee, and S. Kang,(*2014–2026, Dec. 2010)*

[5] “An advanced BIRA fora branch analyzer,” *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst., vol. 29, no. 12, pp.*

[6] R.-F. Huang, J.-F. Li, J.-C. Yeh, and C.-W. Wu, (Aug 2013) “Raisin: Redundancyanalysis algorithm simulation,” *IEEE Des. Test Comput., vol. 24, no. 4*,pp. 386–396,

[7] S.-K. Lu, Z.-Y.Wang, Y.-M. Tsai, and J.-L. Chen,(Apr. 2012) “Efficient built-in selfrepairtechniques for multiple repairable embedded RAMs,” *IEEE Trans.Comput.-Aided Design Integr. CircuitsSyst., vol. 31, no. 4, pp. 620–629*,.

[8] T.-W. Tseng, J.-F. Li, and C.-C. Hsu,(Jun. 2010.) “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*,.

[9] T.-W. Tseng, J.-F. Li, and C.-S. Hou(, Nov./Dec. 2010.), "A built-in method to repair SoC RAMs in parallel," *IEEE Des. Test Comput., vol. 27, no. 6, pp. 46–57,*

[10] W. Jeong, I. Kang, K. Jin, and S. Kang,(Dec 2009) "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,

[11] C.-D. Huang, J.-F. Li, and T.-W. Tseng,(*Oct. 2007)* "ProTaR: An infrastructure IPfor repairing RAMs in system-on-chips," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 15, no. 10, pp. 1135–1143.,*

[12] K. Pagiamtzis and A. Sheikholeslami,(Mar. 2006.) "Content-addressable memory (CAM) circuits and architectures: A tutorial and survey," *IEEE J. Solid-State Circuits, vol. 41, no. 3, pp. 712–727,*

[13] Y. Zorian and S. Shoukourian,(May/Jun. 2003.) "Embedded-memory test and repair:Infrastructure IP for SoC yield," *IEEE Des. Test Comput., vol. 20, no. 3,*pp. 58–66,