BUILT-IN SELF-TEST REPAIR (BISTR) TECHNIQUE FOR RADOM ACCESS MEMORIES (RAMs)

[1]A. SRUTHI, [2]V.R. SHESHAGIRI RAO

[1,2]Institute of aeronautical engineering, Hyderabad

Abstract— Built-in self-test repair (BISTR) technique has been most widely used to test repair embedded random access memories (RAMs). This paper proposes a reconfigurable BISTR (ReBISTR) scheme totest repairing RAMs with different sizes and redundancy organizations. An efficient redundancy BIST algorithm is proposed to allocate redundancies of defective RAMs. In the ReBISTR, a reconfigurable built-in self-test and test repairredundancy analysis is (ReBIRA) design circuit is to perform the redundancy algorithm for various RAMs. Also, an adaptive reconfigurable methodology is proposed to reduce the test repair setup time when the RAMs are operated in normal mode. Due to the complexity of memory architectures, the possibility of occurring manufacturing defects is high. Hence memory testing is necessary. Built inSelf-Testrepair (BISTR) has been proven to be most cost-effective and widely used solutions for memory testing. BISTR technique is used to reduce test repair time. The design architecture is simulatedin Xilinx ISE 14.7 tools.

Keywords— SOC, BIST, BISTR, Test Pattern Generator, FPGA.

I. INTRODUCTION

Very Large Scale Integration (VLSI) has a dramatic technology impact on the growth of digital technology. VLSI has not only reduced the size and the cost but also it increases the complexity size of the circuits.

These improvements have resulted in significant performance to increase in circuits. But it creates some potential problems, which may retard the effective use and growth of future VLSI technology.

Fast Development of VLSI technology results in continuously increasing density of Memory chips. The exponential increase in density makes yield improvement and testing issues.

As the feature size of component shrinks, the sensitivity to faults is also increases. Built inSelf-Testrepair (BIST) can solve the memory testing problems, which increases the comfort predictability. Test patterns generated by a BIST controller can be either deterministic or pseudo random generation. Built inSelf-test and repair (BISTR) is the most efficient technique that is used to faults to be test repair and to improve the yield of memory. To increase memory yield the most of manufactures use incorporated redundancy to replace faulty cells.

Embedded random access memory (RAM) is one of the key component in modern complex system-on-chip (SOC) designs. Typically, many RAMs having with various sizes are included in SOC which they occupy a portion of the chip area. Further, RAMs are subject to elude design rules, such that they are more chance to manufacturing defects. That is, RAMs have more serious problems of design yield and reliability than any other embedded cores in to an SOC.

To make the RAM cores at a reasonable perfect yield level is the very vital for product SOC. Built-in self-test repair (BISTR) technique has been shown to improves the RAM yield most efficiently. Built-in redundancy-analysis (BIRA) algorithm is one of the key component of a BISTR scheme, and it is responsible for allocating redundancies of memory under test. Thus, the BIRA circuit is to have heavily influence on the test repair efficiency of the BISTR scheme.

II. LITERATURE SURVEY

Built-in self-test repair (BISTR): Deep submicron technologies allow the implementation on the multiple memories cells on a single chip. Due to their high densities, memories are more prone to faults. These faults impact on the total chip area. One way to solve this problem is memory enhance by redundant memory locations. The address mapping of the fault free working memory is programmable within certain limits. In order to recognise, a memory test is needed to identify the faulty regions.

The memory is tested by external test hardware or by on chip testing by a dedicated hardware (memory BIST). The second testing strategy is the preferred embedded memories method. After memory testing the memory address map is programmed by means of volatile or non-volatile storage on or off chip. To provide the pattern test from a memory BIST a multiplexer in front of the memory is used widely. The redundant spare rows and columns are often included into the memory. This impacts the performance and area of the memory. The memory is test repaired during testing by storing faulty addresses in registers. These addresses can be streamed out after test to be completion. Furthermore, the application can be started immediately after the memory BIST passes. The redundancy calculation logics will not increase the test time of the memory BIST. The memory BISTR(MBISTR) concept contain interface between the memory BIST(MBIST)

logic and redundancy wrapper for storing faulty addresses. This allows already using existing MBIST solutions. The MBIST controller output must provide three signals to the wrapper logic during test.

Types of RAM:

The two main forms of modern RAM are static RAM (SRAM) and dynamic RAM (DRAM).

SRAM

In static RAM, a bit of data is stored using the state of a flip-flop.

DRAM

Dynamic RAM stores a bit of data using a transistor and capacitor pair, which they combine together comprise a memory cell.

ECC RAM

ECC memory (which can be either SRAM or DRAM) includes special test circuitry to detect and/or correct random faults (memory errors) in the stored data, using parity bits or error correction code.

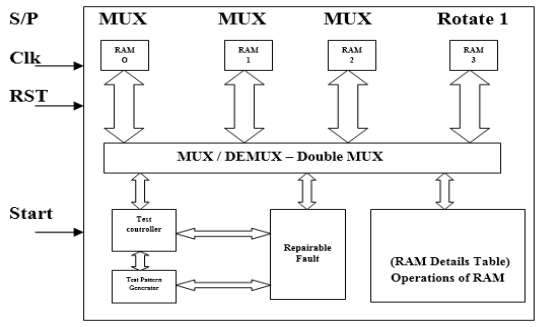

Figure 1 Block Diagram

2.2.2 Models of Memory Faults:

Embedded memory testing isn't new and because memory is used as a test vehicle for technology development, testing of memory has been well studied fairly. Nevertheless, such testing remains a major problem in embedded-core-based microprocessors of complex as well as in SOC. In so many cases, built-in self-tests are used to test embedded memory. However, several methods of testing are used in the designs of SOC. Repaired memory is also a key issue in SOC for large embedded memories.

There is diversity of internal faults can be occurred in semiconductor memories, causing various types of failures in the function of the memory. These faults to be detected necessarily to test the procedures can be classified as in any digital circuit test into three more classes i.e., parametric testing of DC, parametric testing of AC and testing of function.

Multiplexers, or MUX's, can be either circuit of digitals made up of switch digital used to logic gates which are high speed or binary data.

The rams are used to store data and reconfigurable rams are used to restore the correct data. First the operation is done with respective of their application and testing patterns are generated to test their operation.

Test controller will control the circuit with pattern generations and total details are stored in ram detail table.

The fault pulse acts as an activation signal for programming the array. The FA (fault asserted) indicates that a fault has been detected. The field of a word contains the faulty operation, here as the data field is programmed to contain the correct data which is compared with the memory output.

In this mode the input multiplexer connects test collar input for memory under test as generated by the BIST circuitry of controller. While detecting the memory locations as fault by the fault diagnosis module of BIST Controller, the redundancy array is programmed in the figure 1

Memory is used as a unit under test. Fault Diagnosis is used to compare the data that expected with the data in original. If any change is there, it gives that location address and actual data to the Redundant Logic Array as input. This Redundant Logic Array acts as the redundant memory. In this we will store the data and the address of the faulty locations of memory. In normal mode it compares normal input address with the existing faulty locations; for read and write operations redundant logic memory will be used if it matches. If it doesn't match it will use the memory which is in original for the operations of read and write. Output multiplexer is used to select one value from the Redundant memory and Memory depending whether it is faulty or not.

The total slices, LUTs, IOBs used is shown in below table 1.

TABLE 1 Slices, LUT's & IOB's

Logic Utilization	Used	Available	Utilization
Number of Slice LUTs	15	63400	0%
Number of fully used LUT-FF pairs	0	15	0%
Number of bonded IOBs	43	210	20%

The output waveform is shown in below figure 2, the four faulty rams are undergoes BIST test and then it produces faultless output.

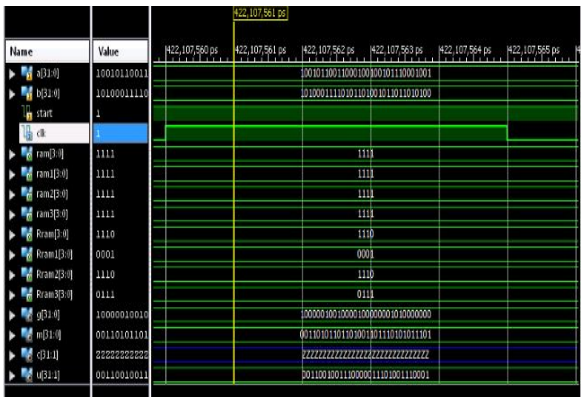

Figure 2 Output Waveform

The dealy and memory used reprts are shown in below.

```
Timing Details:
All values displayed in nanoseconds (ns)
Timing constraint: Default path analysis
Total number of paths / destination ports: 258 / 32
Delay:          1.917ns (Levels of Logic = 5)
Source:         b<0> (PAD)
Destination:    ram<3> (PAD)
Data Path: b<0> to ram<3>
                        Gate    Net
  Cell:in->out      fanout  Delay   Delay  Logical Name (Net Name)
  ----------------------------------------  ------------
  IBUF:I->O             5  0.001   0.530  b_0_IBUF (b_0_IBUF)
  LUT3:I0->O            3  0.097   0.389  ga1/G1 (u<1>)
  LUT5:I3->O            7  0.097   0.407  ga2/G1 (u<2>)
  LUT5:I3->O            5  0.097   0.298  Mxor_ram<3>_xo<0>1 (Rram_3_OBUF)
  OBUF:I->O                0.000         ram_3_OBUF (ram<3>)
  ----------------------------------------
  Total                   1.917ns (0.292ns logic, 1.625ns route)
                                  (15.2% logic, 84.8% route)
=========================================================================
Total memory usage is 318332 kilobytes
```

CONCLUSION

The simulation results have shown that the ReBISR architecture is successfully able to implement new test algorithms. Implementation of a single test operation in one micro word ensures that any future test algorithms with any number of test operations per test element are successfully implemented using the current ReBISR architecture. Moreover, it provides a flexible approach as any new algorithm, other implemented using the same ReBIST hardware by changing the instructions in the RAM storage unit, without the need to redesign the entire circuitry. The Synthesis Report, Map Report, RTL Schematics are generated using Xilinx 14.7 i. The simulation results are generated and verified.

REFERENCES

[1] "High-Density and High-Performance 6T-SRAM for System-on-Chip in 130 nm CMOS Technology", W. Kong, R. Venkatraman, R. Castagnetti, F. Duan and S. Ramesh, VLSI Symposium, 2001.

[2] "*Design-Manufacturing Interface in the Deep Submicron: Is Technology Independent Design Dead*?" Carlo Guardiani, 1st Intl. Symposium on Quality of Electronic Design, March 20 - 22 , 2000, p. 447.

[3] Kim, Y. Zorian, G. Komoriya, H. Pham, F. P. Higgins, and J. L. Lweandowski, "Built in self repair for embedded high density SRAM," in Proc. Int. Test Conf. (ITC), Oct. 1998, pp. 1112–1119.

[4] 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.

[5] M. Nicolaidis, N. Achouri, and S. Boutobza, "Dynamic data-bit memory built-in self-repair," in Proc. IEEE/ACM Int. Conf. Comput.- Aided Des. (ICCAD), San Jose, CA, Nov. 2003, pp. 588–594.

[6] P. Ohler, S. Hellebrand, and H.-J. Wunderlich, "An integrated built-in self-test and repair approach for memories with 2D redundancy," in Proc. IEEE Eur. Test Symp. (ETS), Freiburg, May 2007, pp. 91–99.

★★★