ISSN: 2366-1313

Realization of Built-In Self Test (BIST) Enabled Memory (RAM) Using VHDL and Implementation inSpartan6 FPGA board

[1]RAMYA RAPAKA, [2]Mrs. U. RAJITHA,

[1]M Tech Student, Dept. Of ECE, Vaagdevi Engineering College, Bollikunta, Warangal.

[2] Assistant Professor, Dept. Of ECE, Vaagdevi Engineering College, Bollikunta, Warangal.

Abstract: *Systems that use the Built-In Self Test (BIST) approach are able to discover and fix mistakes automatically. The BIST screening method physically places the testing functions with the CUT. When the system's dependability is paramount and "failure is not an option," BIST might substantially simplify layout at the system level. The system as a whole must be error-free before deciding to carry out a crucial operation. Analyzed are the results of the tested exclusive circuit using BIST structures, which provide pseudo-random combinations. Everything from whole designs to individual design blocks—or even structures inside design blocks—can be examined by BIST. Memory is an intricate design with several uses from a manufacturing standpoint. Adding a few more pins to BIST may make memory testing a breeze. Actually, a BIST memory test can check the whole memory IC with only a clock signal and a few of pins. A SPARTAN 6 FPGA board was successfully integrated with the suggested BIST enabled RAM via the use of a VHDL design.*

Key words: BIST, MEMORY, VHDL, FPGA

I. INTRODUCTION

We put each part through its paces using the BIST method of design. With a well-designed BIST architecture, the benefits of higher dependability and lower maintenance costs will more than outweigh the extra cost of testing equipment. Innumerable methods exist in which BIST contributes to cost reduction. Among its many benefits are enhanced system-level maintenance, more efficient component repair, less frequent chip testing, and much more. [1][2].One further option to cut down on testing time and test several circuits simultaneously is to use a functional real-time clock. With each test cycle, less time is consumed. [3].Memory chips are important to the operation of any and all consumer gadgets, desktop computers, and portable computers. At the end of the day, all users should go with RAM that supports BIST. [4]. The proposed procedure is to build traditional Random Access Memory (RAM) chips

ISSN: 2366-1313

with BIST enabled designs using Very High Speed Integrated Circuit Hardware Description Language (VHDL) and linear feedback shift registers (LFSRs). [5].The outline of the article is as follows: The BIST architecture is covered in full in Section II. In Section III, we take a look at how VHDL handles CUT (RAM) implementations, in Section IV, we present the comparator, and in Section V, we investigate LFSR in depth. The section-specific simulation results will also be sent to you. The SPARTAN 6 FPGA hardware with the ISE Xilinx12 software run all algorithms and simulations on a single system clock.

II GENERAL BIST CONSTRUCTION

The fundamental BIST design for a digital circuit, which includes the CUT, requires three more hardware/software construction components. Give these a look: i) A test pattern generator ii) The components highlighted in Figure 1, which include a test controller and a response analyzer.

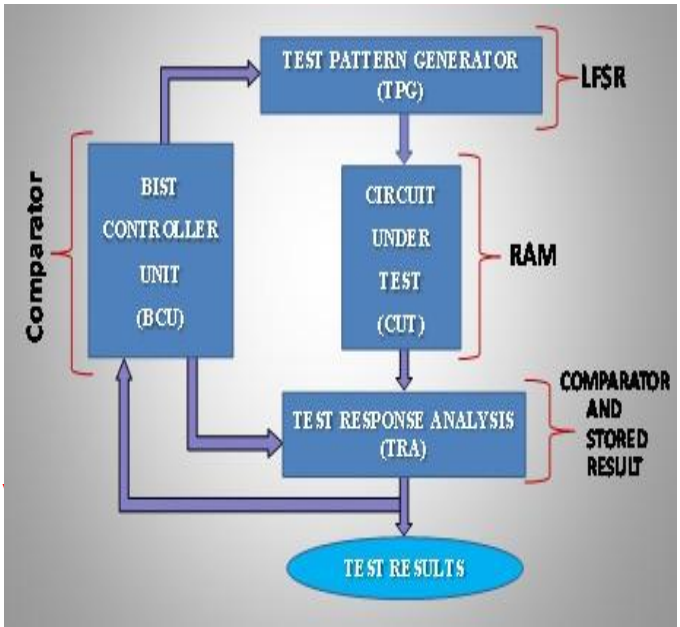


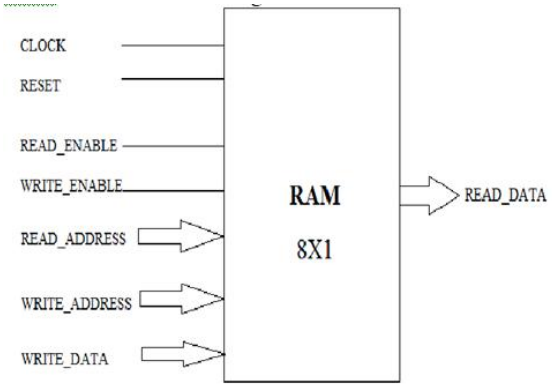
Fig 1: General BIST Construction

The test pattern generator is a feature that the CUT may use to create test patterns. The CUT is often tasked with carrying out directives given by different parts of the facility. While the CUT is in test mode, the comparator verifies its output using pseudo-random test patterns. According to the project proposal, the Test Response Analyzer (TRA), Test Pattern Generator (TPG), and CUT will each have a BIST Controller Unit (BCU), a comparator, and usual RAM installed.

III CIRCUIT UNDER TEST (RAM)

The main building block of the proposed work is

Standard RAM as shown in Fig. 2



ISSN: 2366-1313

Fig. 2: RAM architecture block diagram

The functionality of all the pins is described in Table 1

Table 1: Port Name and purpose of standard RAM

PORT NAME	TYPE	DESCRIPTION
CLOCK	INPUT	Provides clock signal to the block.
RESET	INPUT	Provides reset signal to the block
READ_ENABLE	INPUT	Enable reading operation of a ram.
WRITE_ENABLE	INPUT	Enable reading operation of a ram.
READ_ADDRESS	INPUT	It shows that address from where reading operation will be done.
WRITE_ADDRESS	INPUT	It shows that address where we have to write our data.
WRITE_DATA	INPUT	It is that data which should be written on given write address.
READ_DATA	OUTPUT	It will show the data placed at given read address after reading operation

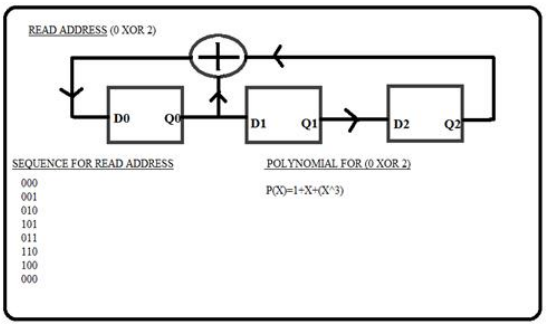
The simulation result of RAM only is shown in Fig. 3.

Fig. 3: Simulation Result of RAM

IV LFSR (BIT PATTERN GENERATOR)

A hybrid of XOR gates and shift registers, Linear Feedback Shift Registers (LFSRs) integrate these two concepts. The typical result of LFSR is pseudo-random sequences. The term "pseudorandom binary sequence" describes a string array of '0's and '1's that is designed to repeat after a certain clock pulse but seems to be generated at random. A D-flip flop and an XOR gate are the basic components of an LFSR. The usage of three LFSRs is operationally justified for three separate reasons. All they can do is read addresses, write data, and make up random numbers. A pseudo-random sequence and an LFSR block are shown in these illustrations.

The LFSR1 has to produce pseudo-random sequences if it wants to save read addresses into RAM during testing. Figure 5 shows the building's plans while Figure 4 shows the results of the simulation.



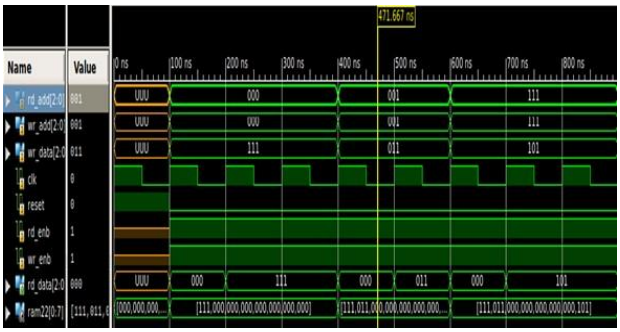


Fig. 4: Architecture of LFSR1 (generate Read Address)

ISSN: 2366-1313

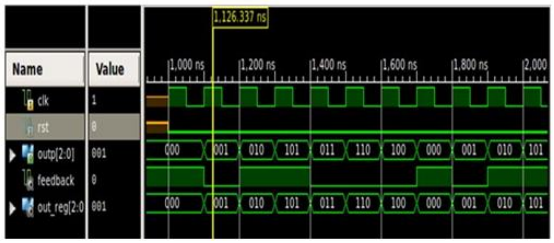


Fig. 5: Simulation Result of LFSR1

LFSR2: It is used for generating pseudo random sequences for write address that is going into ram when test mode is on. The architecture and simulation result are shown in Fig. 6 and Fig. 7.

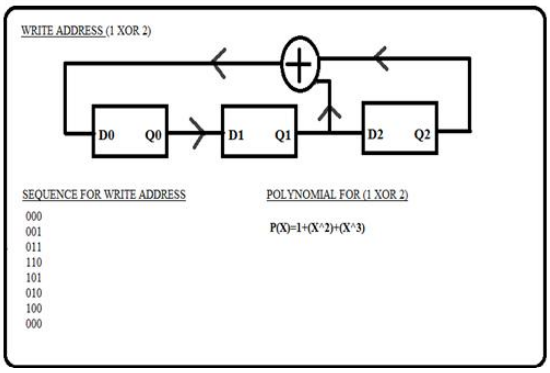


Fig. 6: Architecture of LFSR2 (generate Write Address)

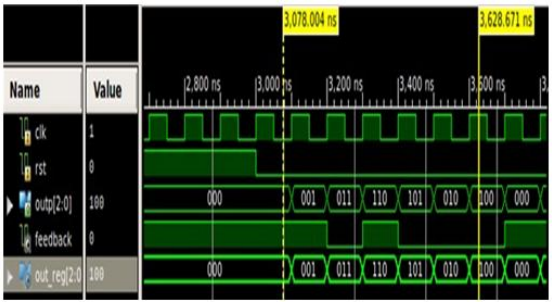


Figure 7: Simulation Result of LFSR2

LFSR3: It is used for generating pseudo random sequences for write data that is going into ram when test mode is on. The architecture and simulation result are shown in Fig. 8 and Fig. 9.

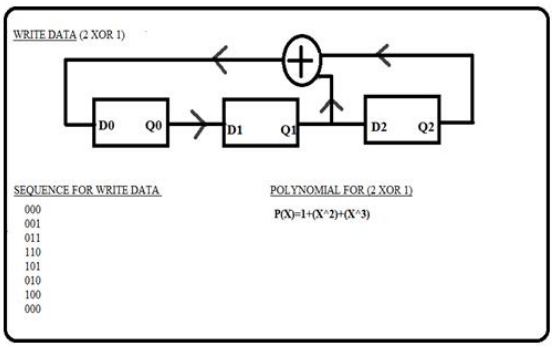


Fig. 8: Architecture of LFSR3 (generate Write Data)

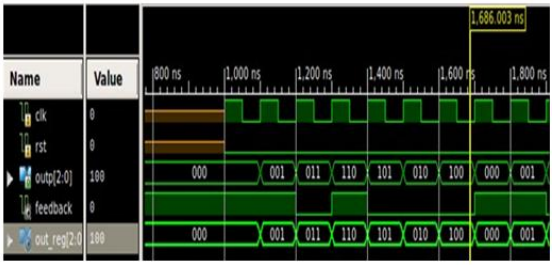


Fig. 9: Simulation Result of LFSR3

V BIST ENABLED RAM (FINAL PRODUCT)

As previously indicated, more pins are needed for the BIST design in order to verify the CUT. Figure 10 shows how the suggested solution adds a pin named check_pin to the CUT, making it testable. Testing mode is activated when check_pin=0 and normal mode is entered when it reaches 1. In a typical mode of operation and with properly functioning RAM, there is no correlation between the READ_DATA output and the stored data. While running tests, RAM reads data from all three LFSRs, uses a comparator to compare the READ_DATA output to the projected stored output, and then uses

ISSN: 2366-1313

BIST_OUTPUT to return the RAM's state. While in test mode, the comparator will assert the error signal (BIST_OUTPUT) if it finds a discrepancy between the received data and the recorded intended output. Because comparing in normal mode is superfluous, the error signal won't be issued. Having three 2:1mux is one way to differentiate between test mode and normal mode, as seen in Figure 11.

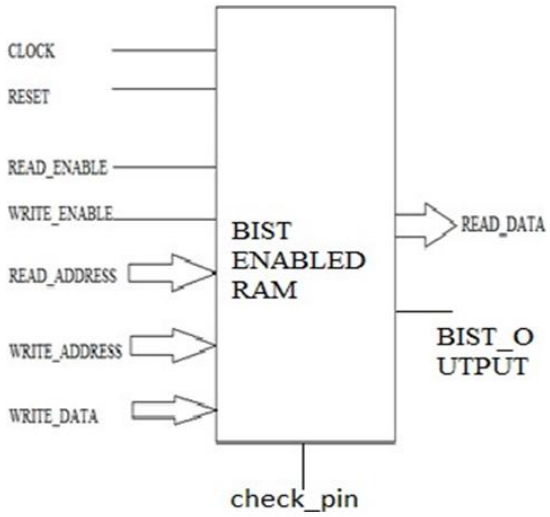


Fig. 10: Entity of Final Product

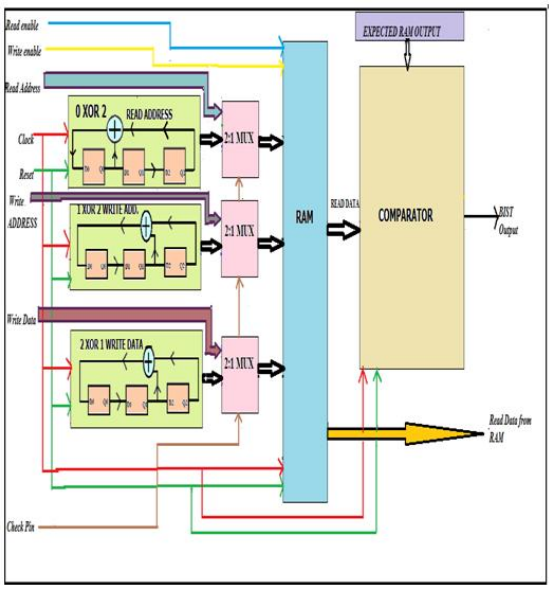


Fig. 11: Internal Diagram of Final Product

The complete working principle is shown with the help of a flow chart in fig. 12.

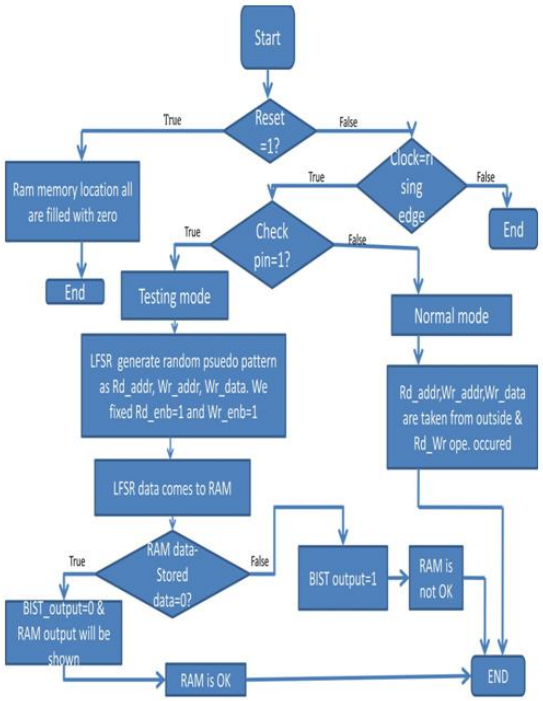


Fig. 12: Flow Chart of the final product

The simulation outcome of the BIST embedded RAM is shown in Fig. 13, Fig. 14 and Fig. 15 respectively depicting three different cases.

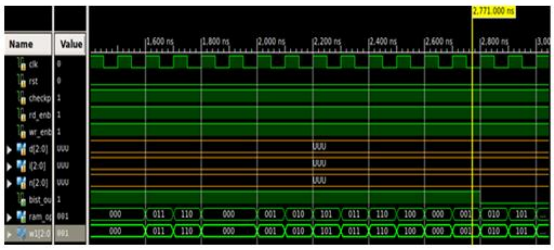


Fig. 13: Simulation result (CASE 1)

CASE 1: check_pin=1 and reset=0 then for rising edge clock and read

ISSN: 2366-1313

enable=1, write enable=1 and data stored in register in a proper order are matched with read data then BIST_OUTPUT will be 0 and READ_DATA will be shown.

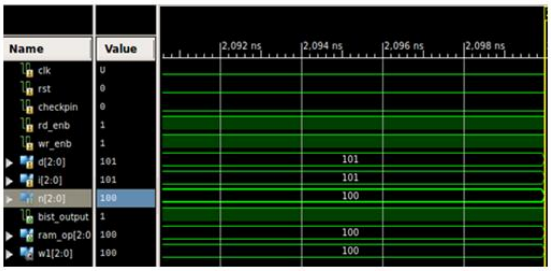


Fig. 14: Simulation result (CASE 2)

CASE 2: check_pin=0, read enable=1, write_enable=1 and reset=0 at rising edge clock READ_ADD, WRITE_ADD, WRITE_DATA taken from outside and READ_DATA will beshown.

Finally, in case 3, if reset=0 and check_pin=1. If the data saved in the register does not match the received data, then BIST_OUTPUT=1 is set for the rising edge clock and read_enable=1.

Based on the simulation findings, the suggested RAM with BIST enabled is fully functional. The same has now been successfully done in the SPARTAN 6 FPGA.

VI HARDWARE IMPLEMENTATION OF THEPROPOSED WORK

After synthesis there were no errors which indicate that the proposed work can be successfully implemented in FPGA board.

Synthesis Report:

Counters:

1. 3 bit up counter = 1
2. 4 bit up counter = 1

Register:

1. Flip flop= 2

Multiplexer:

1. 1 bit 2 to 1 mux =13
2. 3 bit 2 to 1 mux =3
3. 3 bit 8 to 1 mux =1
4. 7 bit 2 to 1 mux =1

XORS:

1. 1 bit xor 2->3

Device Utilization Summary:

1. Number of slice Registers: 59 out of 30064 =0.01%
2. Number used as logic: 84 out of 15032=0.5%
3. Number of LUT Flip flop pair used =90
4. Number of unique control sets: 11
5. Number of bonded IOBS 42 out of 259 =16%
6. Number of BUFG/BUFGP 1 out of 16=6%

ISSN: 2366-1313

Clock Information:

Clock signal	clock buffer
Clk	BUPGP

No asynchronous control signal is there.

Timing Summary:

Speed Grade:-3

Minimum period: 2.897 ns (Max frequency 345.173MHz) Minimum input arrival time before clock: 4.386 ns Maximum output retuned time after clock: 4.172 ns

Delay: 2.897 ns Source: T2/flag_2 (FF) Destlutron: T2/flag_3 (FF) Source clock: CLK rising Destination clock: CLK rising Total memory usage is 138892 Kilobytes Total Real time to Xst completion: 3.00 sec Total CPU tome to Xst completion: 2.79 sec The synthesis report indicates no use of asynchronous clock signal. Only single on board synchronous clock signal is used.

VII CONCLUSION

We successfully implemented the standard RAM with BIST enabled capabilities on a Spartan6 FPGA board after building it in VHDL. The Spartan6 FPGA board is used to test and verify all of the proposed design components before they are integrated. There is complete concordance between the real and virtual results obtained from the FPGA board. The standard mode of operation employing multiplexers may keep memory speeds comparable to those without BIST architecture. Instead of using a comparator in the BIST architecture, a multiple input signature analysis register (MISR) may be used to boost reliability if the CUT is extremely big. It is definitely within the range of possibilities to extend this procedure to assess many IC memories simultaneously. With BIST testing enabled, RAM improved its own security and dependability.

REFERENCES

[1] Jayashri Patil, Shashank Pujari, and Dr.A.D. Shaligram, "Designing, Modelling and Implementation of Memory Built In Self Test (BIST) Controller," Proc. On Int. Conference on Control, Communication and Power Engineering, 2010, pp359-361.

[2] Tapas Tewary, and Atanu Sen, "A novel approach to realize Built-in- self-test (BIST) enabled UART using VHDL,"ACES 14.

[3] M.H. Husin, S.Y.Leong, M.F.M. Sabri and R.Nordiana, "Built in self Test for RAM using VHDL," CHUSER 2012, pp272-276

[4] K. Zarrineh, and S. J.

ISSN: 2366-1313

Upadhyaya, “On programmable memory built-in self test architectures,” Design, Automation and Test” in Europe, Conference and Exhibition 1999. Proceedings, 1999, pp. 708 -713

[5] P. J. Anderson, “The designer’s guide to VHDL”, Morgan Kaufman, 2nd edition, 2002.