International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347-6435(Online) Volume 12, Issue 09, September 2023)

Design and Implementation of a Low Area and Delay Test Pattern for BIST with LFSR

Shailesh Yede[1], Vijay Yadav[2]

[1]*Research Scholar, Department of Electronic & Communication Engineering, LNCT, Bhopal, India*
[2]*Professor, Department of Electronic & Communication Engineering, LNCT, Bhopal, India*

[1]shaileshyede120@gmail.com, [2]vijayky@lnct.ac.in

***Abstract*-- 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. A linear-feedback shift register (LFSR) is a shift register whose input bit is a linear function of its previous state. Field programmable gate array (FPGA) has become widely accepted design approach for low- and medium-range application because of functional flexibility and low development cost. The role of testing is to detect whether something went wrong and the role of diagnosis is to determine exactly what went wrong. This paper proposes an implementation of built in self test using verilog coding on xilnx 14.7 software. In this work, also implement Linear Feedback Shift Register and SRAM controller to support BIST for performance improvement.**

***Index Terms* – BIST, LFSR, SRAM, Controller, Latency, Area, High Speed, Low Power.**

I. INTRODUCTION

The BIST is also the solution to the testing of critical circuits that have no direct connections to external pins, such as embedded memories used internally by the devices. In the near future, even the most advanced tester may no longer be adequate for the fastest chip, a situation where in self-testing may be the best solution for. Implementing BIST lower cost of test, since the need for external electrical testing using an ATE will be reduced, if not eliminated. Better fault coverage, since special test structures can be incorporated onto the chips. Shorter test times if the BIST can be designed to test more structures in parallel. The consumers themselves to test the chips prior to mounting or even after these are in the application boards.

The process of testing the fabricated chip on ATE involves the use of external test patterns applied as stimulus. The device's response is analyzed on the tester, comparing it against the golden response which is stored as part of the test pattern data. MBIST makes this easy by placing all these functions within a test circuitry surrounding the memory on the chip itself. It implements a finite state machine (FSM) to generate stimulus and analyze the response coming out of memories.

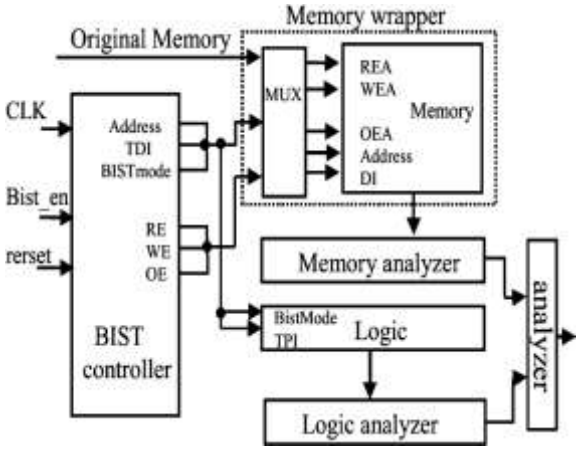

Figure 1: BIST internal architecture

Figure 1 shows efficient built-in self-test internal architecture. The memory-based architecture is suitable for high computation point applications such as ADSL and OFDM systems. The FFT processor is first divided into the memory part and the logic part which can be tested under the supervision of the same BIST controller. The BIST controller can not only perform traditional memory test algorithms but also generates test patterns required for the logic part. The adopted memory test algorithm can be programmed by the users which covers different types of memory faults. For the logic part, the single cell fault model is assumed. Our BIST architecture tests both parts simultaneously such that the test time can be reduced greatly. The hardware overhead of our approach is also very low since novel design-for-testability techniques are applied for the logic part which mainly consists of multipliers.

This extra self-testing circuitry acts as the interface between the high-level system and the memory. The challenges of testing embedded memories are minimized by this interface as it facilitates controllability and observability. The FSM provides test patterns for memory testing; this greatly reduces the need for an external test pattern set for memory testing.

International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347-6435(Online) Volume 12, Issue 09, September 2023)

II. PROPOSED METHODOLOGY

The proposed methodology is explained using following flow chart-

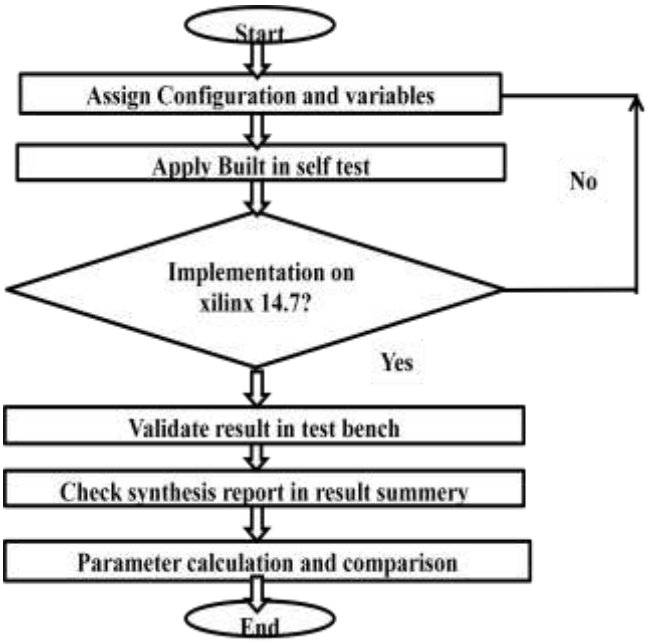

Figure 1: Flow Chart

A micro programmed data-path for division of fractional numbers, presented in Fig.2. It consists of a register block for storing the dividend, the divisor, intermediate results of division, the quotient, and the counter of cycles. All the micro operations needed in the division procedure are carried out in the Arithmetic and Logic Unit (ALU) which has the role of CUT in this work. The ALU has data inputs and outputs connected via buses to the register block. The control signals from the control unit serve as additional inputs for ALU, and status signals of the ALU serve as additional outputs connected to the control unit. During N cycles of the micro program ALU is exercised with N functional patterns, and the responses of ALU will be compressed in the signature analyzer which monitors the whole division process.

1. System Overview:

- *Define Objectives:* Clearly state the testing objectives, including the targeted fault coverage, area constraints, and desired test time.
- *Select LFSR:* Choose an LFSR with an appropriate feedback polynomial and size based on area and fault coverage requirements.

2. LFSR Configuration:

- *LFSR Initialization:* Initialize the LFSR with a seed value. The seed value should be selected to maximize fault coverage and minimize test time.
- *Feedback Polynomial:* Define the feedback polynomial for the LFSR. The choice of polynomial directly affects the test patterns' quality and the circuit area overhead.

3. Test Pattern Generation:

- *LFSR Operation:* Configure the LFSR to generate pseudo-random test patterns. The LFSR should be able to run for a sufficient number of clock cycles to generate a comprehensive set of test patterns.
- *Pattern Compression:* Implement a compression algorithm to reduce the number of generated patterns if needed. This helps in minimizing test time.

III. SIMULATION AND RESULTS

Proposed BIST with LFSR is implemented in Xilinx 14.7 platform with verilog language. Implementation is performed on Vertex-V family and simulation is validating using Isim simulator.

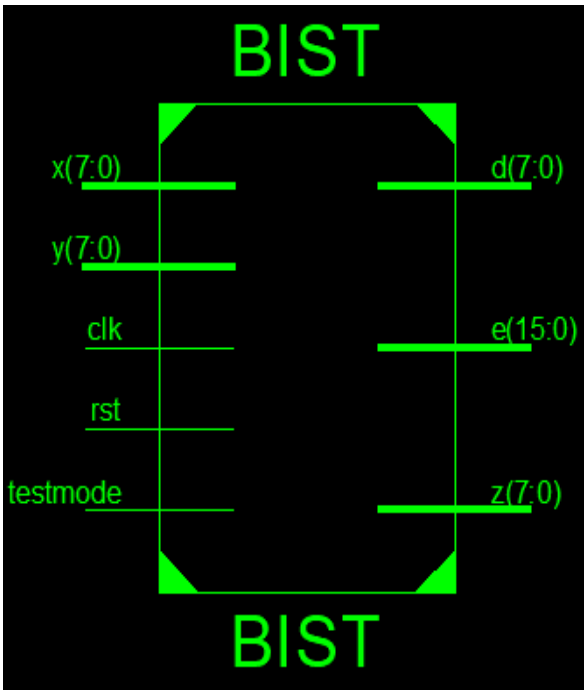

Figure 3: BIST top level block

International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347-6435(Online) Volume 12, Issue 09, September 2023)

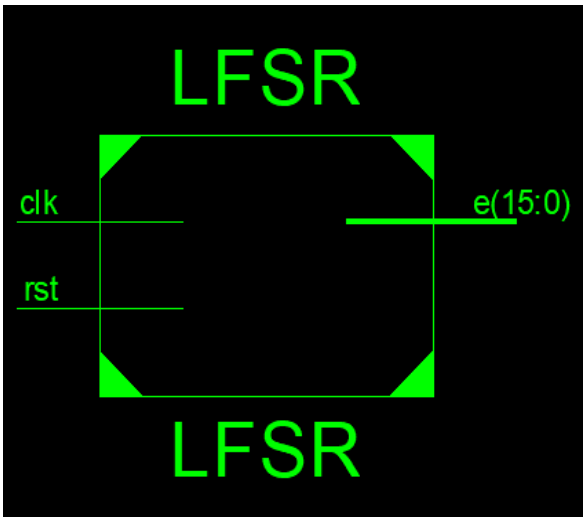

Figure 4: Linear Feedback Shift Register

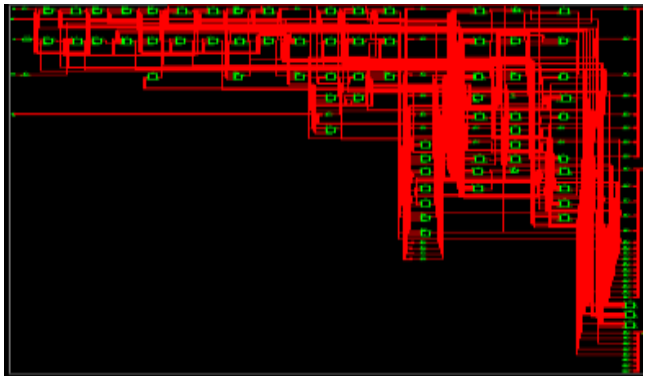

Figure 5: Complete RTL view

Table 1:
Utilization summary

Device Utilization Summary (estimated values)			
Logic Utilization	**Used**	**Available**	**Utilization**
Number of Slice Registers	24	12480	0%
Number of Slice LUTs	44	12480	0%
Number of fully used LUT-FF pairs	11	57	19%
Number of bonded IOBs	51	172	29%
Number of BUFG/BUFGCTRLs	1	32	3%

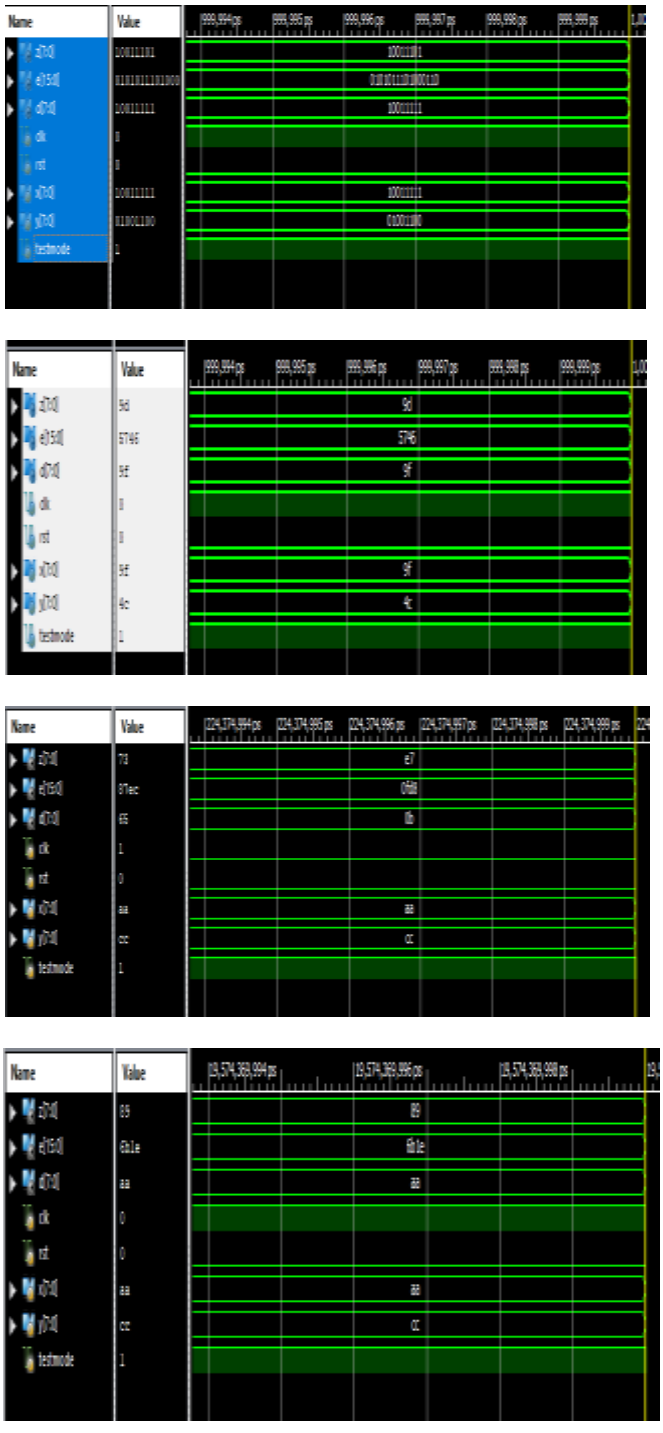

Figure 6: Test bench results

Test bench results show that validation of proposed BIST-LFSR Architecture. As value change in input then self checking start and it end until find fault and fix in output port.

Table 2:
Simulation Parameters

Sr no	Parameter	Value
1	Area	10%
2	Delay	2.338 ns
3	Frequency	427.661MHz
4	Memory	4554920 k
5	Completion Time	17.00 secs
6	Throughput	2.9 Gbps
7	Power	0.156W

In table 2, simulation parameters are showing which is taken during the execution of verilog script.

Table 3:
Result Comparison

Sr. No	Parameters	Previous Work	Proposed Work
1	Area	20%	10%
2	Delay	18.338 ns	2.338 ns
3	Throughput	1.2 Gbps	2.9 Gbps
4	Power	0.192W	0.156W
5	Frequency	400 MHz	427.66 MHz

Therefore proposed work result is better than previous work BIST-LFSR approach is considerable and significant result is achieved.

IV. CONCLUSION

The testing scheme was simulated with Isim simulator for Xilinx Virtex-5. The proposed technique is found to overcome the drawbacks of the previously used BIST and improve performance. Therefore design and implemented Built in self test is better.

We found parameters generated from proposed approach like latency, area and power gives significant achievement than existing BIST with controller.

REFERENCES

[1] M. Mukherjee, G. R S and R. Bhakthavatchalu, "A Proposal for Design and Implementation of a Low Power Test Pattern Generator for BIST Applications," 2022 Second International Conference on Artificial Intelligence and Smart Energy (ICAIS), Coimbatore, India, 2022, pp. 1520-1524, doi: 10.1109/ICAIS53314.2022.9742920.

[2] K. Akhila, N. Karuna and Y. J. M. Shirur, "Design and Implementation of Power Efficient Logic BIST With High Fault Coverage Using Verilog," 2021 International Conference on Networking, Embedded and Wireless Systems (ICNEWS), Bangalore, India, 2021, pp. 1-6.

[3] A. K. S. Pundir and O. P. Sharma, "Fault tolerant reconfigurable hardware design using BIST on SRAM: A review," *2020* International Conference on Intelligent Computing and Control (I2C2), Coimbatore, 2020, pp. 1-16.

[4] T. Koshy and C. S. Arun, "Diagnostic data detection of faults in RAM using different march algorithms with BIST scheme," 2016 International Conference on Emerging Technological Trends (ICETT), Kollam, 2016, pp. 1-6.

[5] S. Kostin, R. Ubar, M. Gorev and G. Mägi, "Comparison of two approaches to improve functional BIST fault coverage," 2014 14th Biennial Baltic Electronic Conference (BEC), Tallinn, 2014, pp. 105-108.

[6] N. Das, P. Roy and H. Rahaman, "Built-in-self-test technique for diagnosis of delay faults in cluster-based field programmable gate arrays," in IET Computers & Digital Techniques, vol. 7, no. 5, pp. 210-220, September 2013.

[7] M. H. Husin, S. Y. Leong, M. F. M. Sabri and R. Nordiana, "Built in self test for RAM using VHDL," 2012 IEEE Colloquium on Humanities, Science and Engineering (CHUSER), Kota Kinabalu, 2012, pp. 272-276.

[8] P. Manikandan, B. B. Larsen and E. J. Aas, "Experiments with ABIST test methodology applied to path delay fault testing," 2010 East-West Design & Test Symposium (EWDTS), St. Petersburg, 2010, pp. 59-63.

[9] N. Q. Mohd Noor, A. Saparon and Y. Yusof, "An overview of microcode-based and FSM-based programmable memory built-in self test (MBIST) controller for coupling fault detection," 2009 IEEE Symposium on Industrial Electronics & Applications, Kuala Lumpur, 2009, pp. 469-472.

[10] Chung-Fu Lin and Yeong-Jar Chang, "An area-efficient design for Programmable memory Built-In Self-Test," 2008 IEEE International Symposium on VLSI Design, Automation and Test (VLSI-DAT), Hsinchu, 2008, pp. 17-20.

[11] M. Y. Niamat, A. Sahni and M. M. Jamali, "A built in self test scheme for automatic interconnect fault diagnosis in multiple and single FPGA systems," 2007 50th Midwest Symposium on Circuits and Systems, Montreal, Que., 2007, pp. 229-232.

[12] H. Takahashi *et al*., "Effective Post-BIST Fault Diagnosis for Multiple Faults," 2006 21st IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems, Arlington, VA, 2006, pp. 401-109.