Implementation and Performance Improvement of Built in Self Test with Linear Feedback Shift Register

Rishi Shukla, Prof. Anshuj Jain
[1]M.Tech Scholar, [2]Associate Professor & HOD
[1,2]Department of Electronics and Communication Engineering, RGPV Bhopal,
[1,2]SCOPE College of Engineering, Bhopal, India.

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.

IndexTerms - 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.

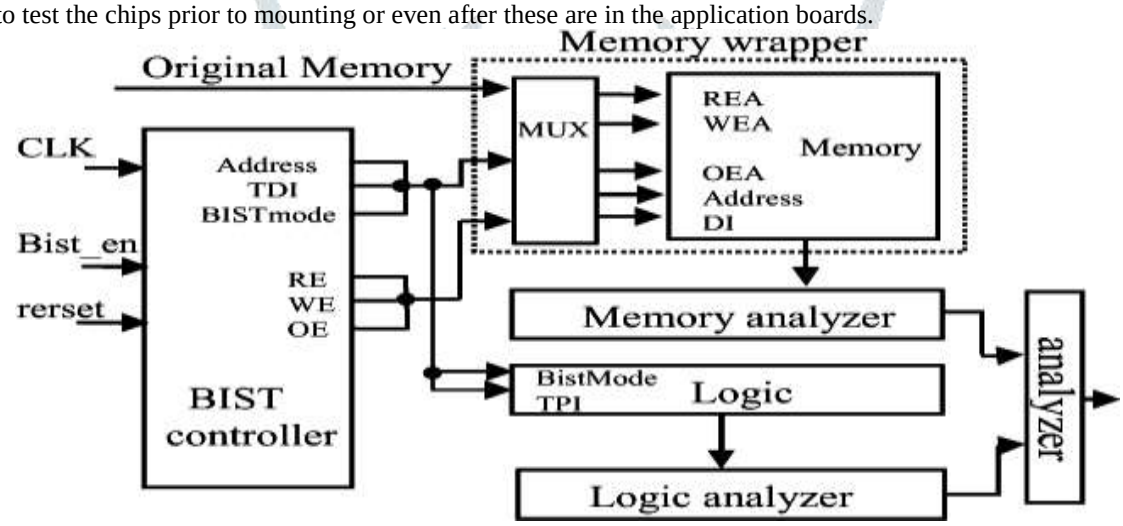


Figure 1: BIST internal architecture

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.

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.
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.

II. LITERATURE OVERVIEW

W. T. Hale et al.,[2018] A method is proposed to improve system reliability in terms of self-detection and isolation of discrete faults. The application of this method on the design of built-in tests for maintenance is presented using the case of aircraft environmental control system maintenance testing. Built-in testing during aircraft on-ground maintenance allows for wider system input variability due to the absence of operational constraints and requirements. This provides an opportunity to optimally design

tests that improve the fault detection and isolation capabilities for operators and maintenance crews. A general mathematical framework for built-in test design is shown using a model-based active fault detection and isolation technique. The motivation for this work is first illustrated through a case study showing the inability to detect and isolate faults at different conditions, demonstrating how system states contribute to these issues. The success of the proposed framework is demonstrated by designing built-in tests that are capable of fully detecting and isolating a multiplicity of faults common to aircraft environmental control systems. it is conclude by presenting the value of the proposed method of detecting and isolating faults, as a solution to a constrained optimization problem with the admissible system inputs as manipulated variables. [1]

K. Akhila, et al.,[2018] Due to ever increasing number of transistors on chip and decrease in feature size have posed challenges in manufacturing and have risen defects due to them. Thus, testing has become vital for any Very Large Scale Integrated (VLSI) design. The design engineers concentrate more on design development and test techniques used to test those designs are neglected due to design cycle time. Any design will not be passed unless it is 100% fault free and Design For Testability (DFT) technique facilitates to detect the faults. Multiple standards are developed to test different part of Integrated circuits. In this work power efficient & high fault coverage Built In Self-Test (BIST) is designed and implemented to test combinational logic. The developed technique is tested on standard combinational circuits and has given promising results. The conventional Linear Feedback Shift Register (LFSR) is modified to generate all the states, hence improving fault coverage. Compared to conventional method, 100% fault coverage & 12.25% reduction in power is achieved by the proposed design. The design is coded in Verilog, verified for functionality using Xilinx ISIM simulator, synthesized by targeting the design to slow_vdd1v0 1.0 library for 45nm technology using Cadence genus tool and validated on FPGA Spartan 6 boards. [2]

S. Wang et al., [2018] Multi-cycle Test applies more than one capture cycles during the capture operation which is a promising way to reduce the test volume of Logic-BIST (Logic Built-in Self-Test) based POST (Power-on Self-Test) for achieving high fault coverage. However, the randomness loss of the capture patterns due to the large number of capture cycles obstructs the further improvement of fault coverage and pattern reduction. In this work, it is propose a novel approach to control the capture patterns by modifying the captured values of scan Flip-Flops (FFs) during capture operation to enhance the test quality of the capture patterns. In the approach, it is insert FF-Control circuits between the scan FFs and the combinational circuit to improve the randomness of the capture patterns by loading toggle vectors/pseudo-random vectors. The experimental results of ISCAS89 and ITC99 benchmarks validated the effectiveness of the proposed methods in fault coverage improvement and random pattern reduction for Logic-BIST. [3]

A. K. S. Pundir et al.,[2017] This Work presents an exhaustive review on BIST technique used in different fault tolerant systems and also consider the fault detection methodologies used in these systems. BIST is an alternative method of at-speed testing of high speed ATE that is expensive and having some unavailability issues. With supplement of an external high speed clock, BIST requires on-chip circuitry for vector generation and response analysis. The pseudo-random vectors gives a good coverage of stuck-at faults and transition faults even if it is applied at high speed. But in case of robust testing, the Coverage of path-delay faults requires some additional modifications in the combinational logic. BIST also provide the testing feature to test circuits for timing delay. For this a standard BIST architecture with a hybrid pattern generator is required, for replacing LFSR-TPG, which can test both stuck at and delay faults. For test insertion BIST is now universally adopted, this is not only because BIST hardware overheads have come down, particularly for memory BIST (1-3%) but also because it enables partitioning of the testing problem for large hardware systems into small ones so currently memory BIST is preferred. Most commonly used schemes to provide pattern generation and response compaction for BIST are LFSRs, MISRs and built-in logic block observers. [4]

T. Koshy et al.,[2017] This Major portion of the SoC's are covered by memories. Dense design creates a wide variety of faults leading to the failures in the memory functions. Built-in self-test (BIST) mechanism is a promising method to test and diagnose embedded memories like RAMs. The advantages include reduced test cycle duration and fewer complexes. March tests are widely used in production test thanks to their low time complexity and high fault coverage. They were capable of locating and identifying the fault types. Diagnostic data exportation by the BIST is time - consuming since it is exported bit-by-bit. To overcome this, a BIST with March-element-based (MEB) Compression scheme is proposed. MEB Compression scheme can efficiently compress the diagnostic data of a RAM tested by a March test. To optimize the performance of the BIST controller, another design is also proposed here. Detection of different faults in a RAM memory using a single counter based BIST design is advantageous. As a result a fusion of March algorithms like MATS, March C and March X are used for the detection of Stuck-at fault, Coupling fault and Transition fault respectively. Advantages of counter-based BIST architecture includes reduced area overhead and complexity. Compression of the test data is carried out using a decoder module. The implementations are carried out by using Verilog Hardware description language and Xilinx ISE 13.2 design suite. Experimental results shows that the proposed methods reduces the fault detection time, redundant diagnostic data, tester storage requirement, area overhead and complexity in the design. [5]

III. PROPOSED METHODOLOGY

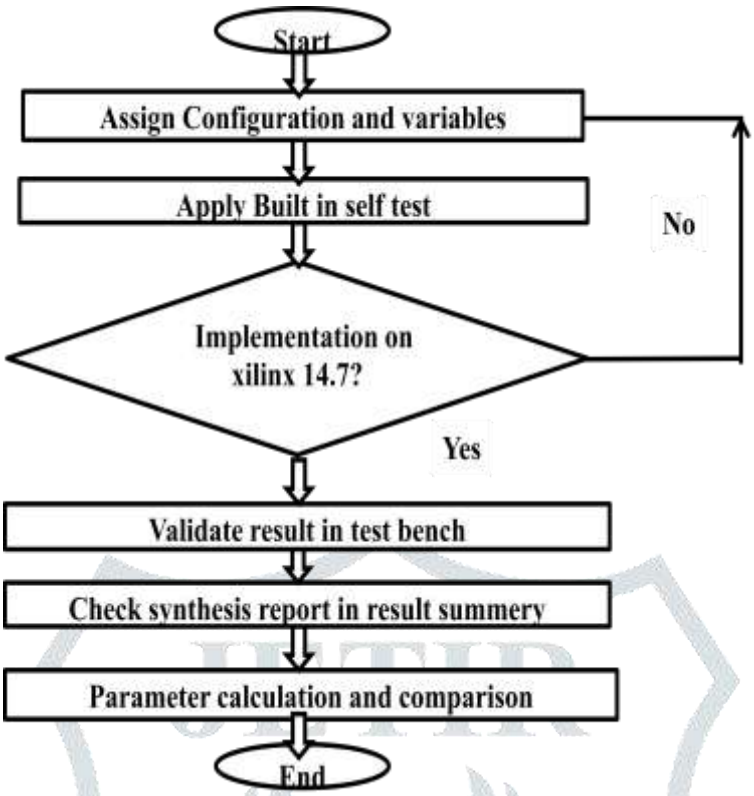


Figure 2: Flow Chart

Consider a micro programmed data-path for division of fractional numbers, presented in Fig.1. 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.

IV. IMPLEMENTATION RESULT

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

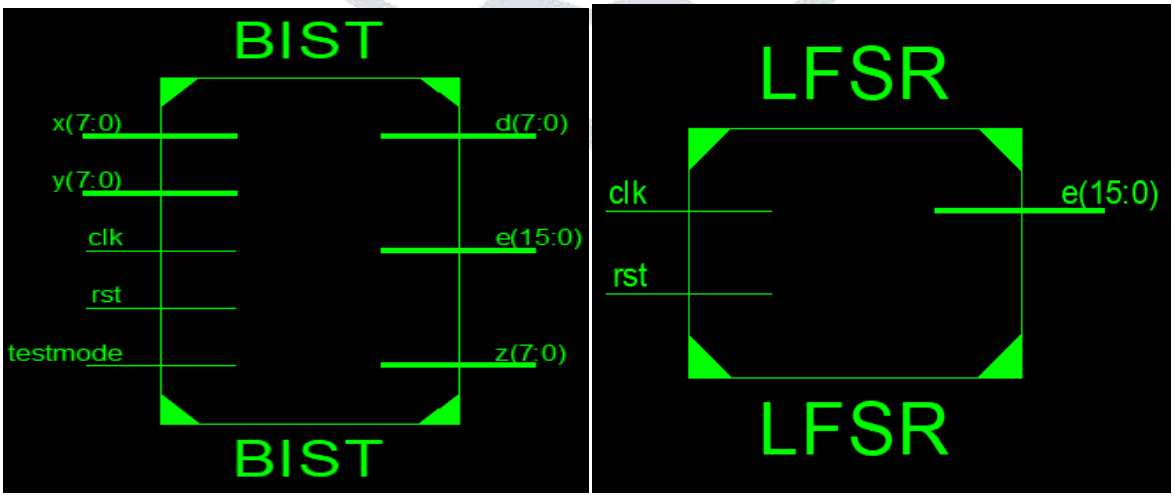


Figure 3: BIST top level block and Linear Feedback Shift Register

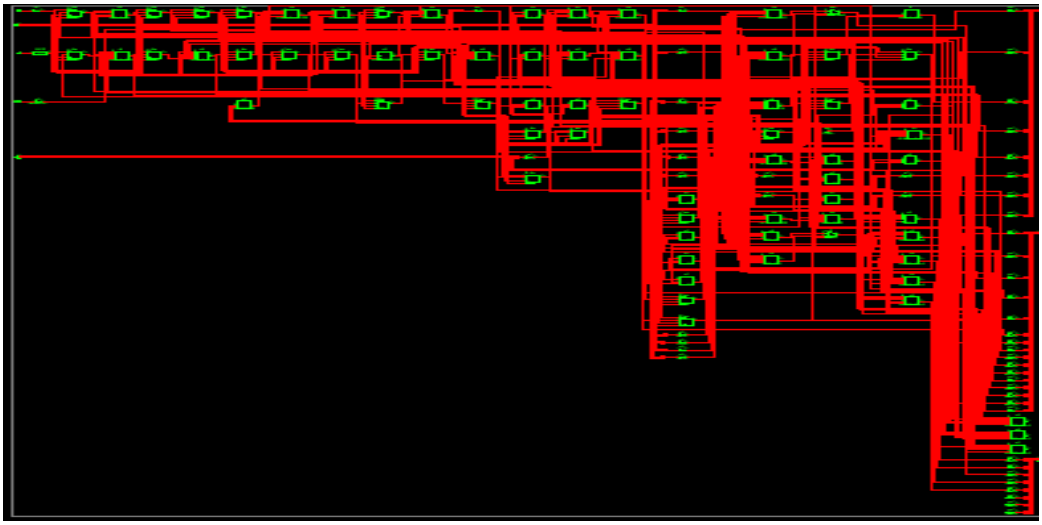
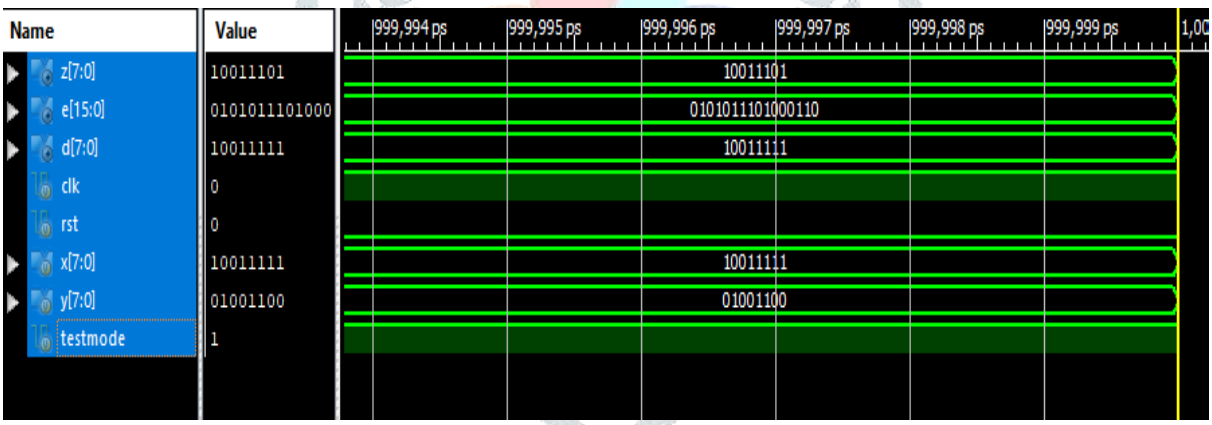
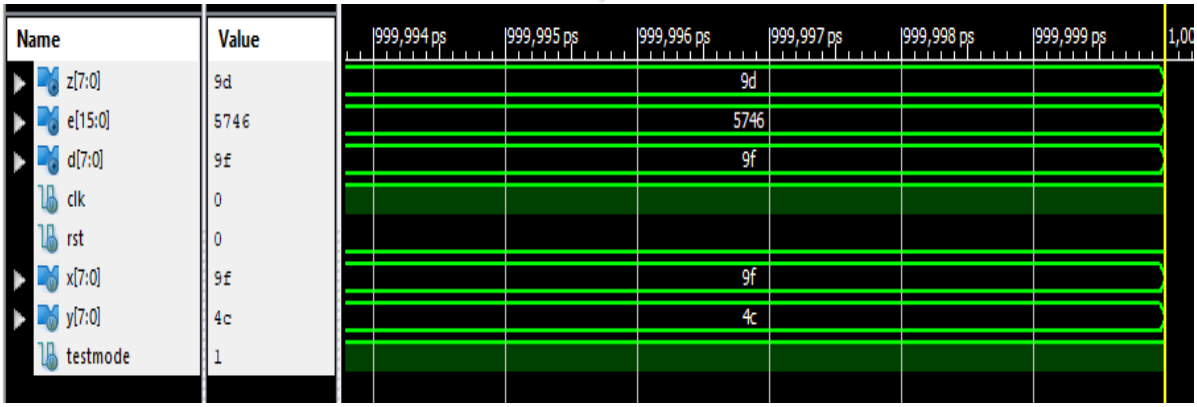


Figure 4: Complete RTL view

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%





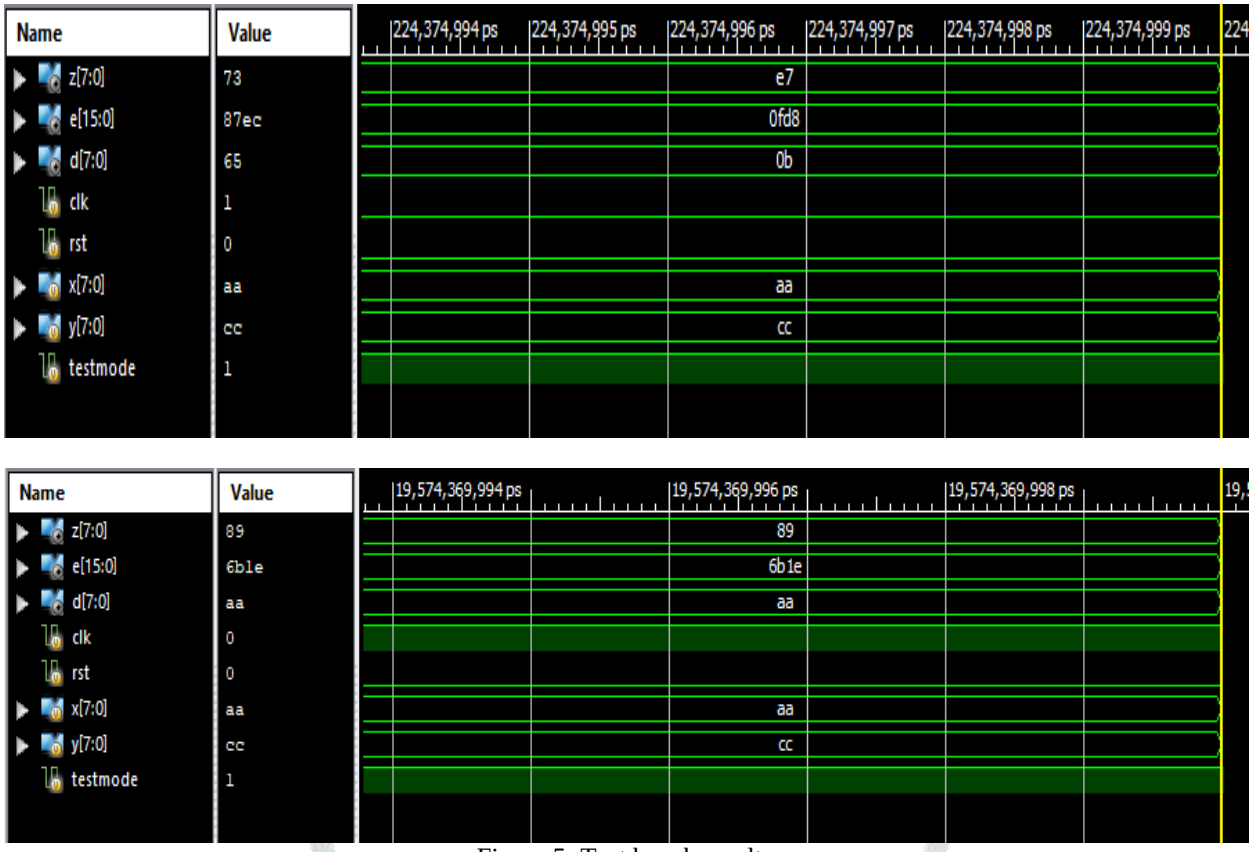


Figure 5: Test bench results

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: Comparison chart of proposed work with Base Work

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.

V. 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. W. T. Hale and G. M. Bollas, "Design of Built-In Tests for Active Fault Detection and Isolation of Discrete Faults," in *IEEE Access*, vol. 6, pp. 50959-50973, 2018.
2. K. Akhila, N. Karuna and Y. J. M. Shirur, "Design and Implementation of Power Efficient Logic BIST With High Fault Coverage Using Verilog," *2018 International Conference on Networking, Embedded and Wireless Systems (ICNEWS)*, Bangalore, India, 2018, pp. 1-6.
3. S. Wang *et al.*, "Capture-Pattern-Control to Address the Fault Detection Degradation Problem of Multi-cycle Test in Logic BIST," *2018 IEEE 27th Asian Test Symposium (ATS)*, Hefei, 2018, pp. 155-160.
4. A. K. S. Pundir and O. P. Sharma, "Fault tolerant reconfigurable hardware design using BIST on SRAM: A review," *2017 International Conference on Intelligent Computing and Control (I2C2)*, Coimbatore, 2017, pp. 1-16.
5. 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.
6. T. Q. Bui, L. D. Pham, H. M. Nguyen, V. T. Nguyen, T. C. Le and T. Hoang, "An Effective Architecture of Memory Built-In Self-Test for Wide Range of SRAM," *2016 International Conference on Advanced Computing and Applications (ACOMP)*, Can Tho, 2016, pp. 121-124.
7. C. Oh, S. Kim and J. Yang, "A BIRA using fault-free memory region for area reduction," *2016 IEEE Asia Pacific Conference on Circuits and Systems (APCCAS)*, Jeju, 2016, pp. 480-482.
8. L. Xia, J. Wu, C. Huang and M. Zhang, "Built-in self-test structure for fault detection of charge-pump phase-locked loop," in *IET Circuits, Devices & Systems*, vol. 10, no. 4, pp. 317-321, 7 2016.
9. W. Kim, S. Cha and L. Milor, "Memory BIST for on-chip monitoring of resistive-open defects due to electromigration and stress-induced voiding in an SRAM array," *Design of Circuits and Integrated Systems*, Madrid, 2014, pp. 1-6.
10. G. Devi Prasanna, P. Abinaya and J. Poornimasre, "Non-intrusive bit swapping pattern generator for BIST testing of LUTs," *International Conference on Information Communication and Embedded Systems (ICICES2014)*, Chennai, 2014, pp. 1-4.
11. G. Harutyunyan, S. Shoukourian, V. Vardanian and Y. Zorian, "Extending fault periodicity table for testing faults in memories under 20nm," *Proceedings of IEEE East-West Design & Test Symposium (EWDTS 2014)*, Kiev, 2014, pp. 1-4.
12. 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.