IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

Built In Self Test For Random Pattern Generator

[1]Dr. K. Suresh Reddy, [2] P. Sneha Priya, [3] C. Ramya Madhuri, [4] T. Jyoshna

[1] Professor and Head of the Department, [2] Student, [3]Student, [4]Student

[1] Electronics and Communication Engineering,

[1] G Pulla Reddy Engineering College, Kurnool, India

Abstract: The proposed BIST architecture focuses on minimizing power consumption during the testing phase while maintaining high fault coverage. It leverages the capabilities of the Verilog hardware description language to model and simulate the design. The research investigates various power reduction techniques, including test pattern compression, selective clock gating, and power-aware test scheduling, to optimize power consumption during testing. This research paper presents the design and implementation of a Built-In Self- Test (BIST) architecture specifically tailored for low-power Very Large-Scale Integration (VLSI) applications. The increasing demand for energy-efficient electronic devices and the proliferation of portable systems have necessitated the development of power-aware design techniques. BIST, a on chip testing technique, plays a crucial role in ensuring the quality and reliability of integrated circuits. The implementation is performed on a field-programmable gate array (FPGA) platform, demonstrating the feasibility and practicality of the proposed design. The research contributes to the field of low- power VLSI design by providing a comprehensive BIST architecture that effectively reduces power consumption during testing. The proposed design is compatible with standard Verilog synthesis and is readily applicable to a wide range of low-power VLSI applications. The findings from this study can guide designers in developing energy-efficient and reliable integrated circuits, promoting sustainability, and extending battery life in portable devices.

Index Terms **–** Built-In Self-Test (BIST), Random Pattern Generation, Test Pattern Generator (TPG), Response Analyzer (RA), Circuit Under Test (CUT), Fault Coverage Optimization, Autonomous Circuit Verification and Validation, System-on-Chip (SoC) Testing, Fault Detection and Diagnosis

I. INTRODUCTION

Built-In Self-Test (BIST) is a powerful on-chip verification technique that enables an integrated circuit to test its own logic without relying on costly external testers. By embedding pattern generators, response analyzer, and compact comparators directly into the silicon, BIST automatically creates, applies, and evaluates test vectors rapidly uncovering manufacturing defects, design faults, or wear-out failures. This self-contained approach slashes test time and equipment expense, boosts fault coverage into the deep corners of the design, and elevates overall device reliability.

The “Built-In Self-Test for Random Pattern Generator” project delivers a Verilog-based BIST engine optimized for minimal power draw and maximal defect detection in modern VLSI chips. Key innovations include primitive-polynomial pattern generation to maximize unique test vectors and minimize repeats, selective clock gating to disable unused circuitry during test and cut switching activity by up to 40%, and power-aware test scheduling to spread switching events, flatten power spikes, and protect battery-powered devices. Fully synthesizable Verilog modules have been validated on FPGA with timing and area metrics demonstrating ASIC-flow readiness. Together, these features create a self-testing solution that uncovers hidden faults, conserves energy, and integrates seamlessly into both FPGA and ASIC design flows paving the way for more reliable, longer-lived electronics in applications from mobile gadgets to high-speed data centers**.**

II. LITERATURE SURVEY

In the earlier period, several researchers and authors have investigated the implementation of BIST architecture for the detection of fault coverage and different techniques to reduce the testing power of VLSI circuits.

V.Kirthi.,et.al., [1] implemented low power BIST for a 32- bit multiplier. The seed value is changed for every two cycles by using m-bit counter and gray code is generated. The proposed technique is highly resistant to faults. Signature analysis is also done using multiple signature register. This signature indicates whether the circuit to be tested is faulty or not. Yuejian Wu.,et.al., [2] proposed a novel method has time efficiency with high grade output verification music of multiple and complex system-on-chip designs. Katti R.S et al. [3] proposed an architecture with low power for Linear feedback Shift Register and it produces the output which gives dynamic dissipation up to 93%. The method is excellent for Built in self-Test (BIST) applications because it results in 2N- 1 distinct pattern for most of the degrees

N. Mohammad Tehranipoor et al. in [4] presents a low transition test pattern generator, called LT-LFSR, to reduce average and peak power of a circuit during test by reducing the transitions within random test pattern and between consecutive patterns. Vivek A. Hadge et al., [5] designed ATPG with D- Algorithm which helps in generating a less number of input pattern for detecting faults like stuck-at-1, stuck-at-0 faults and short circuitry fault. Poornima et al. [6] presents a study Vedic multiplier architecture which has a high speed of 8x8 bit which differs greatly from the basic multiplication method like add and shift. It is a method for hierarchical multiplier design which clearly represents the computational advantages stimulated by Vedic methods.

M. Padma et al., [7] describes the BIST which concurrently monitors the input vectors called as window of vectors to create the circuit inputs during normal mode operation. CAM memory cell is used to store the relative locations of the vectors. High compression is ensured with ASDFR along with MT-filling scheme. Bharti Mishra et.al, [8] proposed a 4-bit multiplier design used in BIST applications and the test pattern generator is designed to generate a random 4-bit number. The modified test pattern generator has a low register-to-bit ratio. The simulation and synthesis witness the efficiency of the low power implementation hardware for applications with a configurable IC. Dong Xiang et.al [9] proposed a new method that consists of low power weighted pseudorandom pattern generator and low power data deterministic BIST with reseeding which guarantees low power operation for clock cycles and also to reduce test data kept on chip.

Govindaraj Vellingiri et.al [10] designed a modified low Transition of linear feedback shift register which gives a significant randomness with equal number of 0 s and 1 s. Due to this method the switching activity by 34% and the power consumption is reduced by 36.2%. Michal Filipek et.al [11] paper shows PRESTO the LP generator. This produces pseudo random test patterns along with scan shift-in switching and an automated programming performs this function. This method also controls the generator, so that the desired fault coverage is achieved faster. This hybrid solution allows the combination of test compression with logic BIST where the high-quality test is delivered.

In the previous works the test patterns are generated using normal polynomial that generates random test vectors and repeated test patterns. Some of the faults were not covered using this normal polynomial and it increases the switching activity in test patterns. The fault coverage is less, and it also increases the power because of switching activity. To overcome these disadvantages, in the proposed work the primitive polynomial is selected based on fault coverage and a new BIST architecture is proposed to reduce the switching activity.

III. EXISTING METHOD

3.1 Manufacturing Test of Integrated Circuits

Specific IC manufacturing process fabrication irregularities may lead in some circuits functioning inappropriately. Before sending the packed circuits to the customer. Before sending the packed circuits to the user customers, manufacturing tests can assist find physical flaws (such as shorts or opens). Comprehensive defect screening. through fault diagnosis is needed after a damaged chip has been found in order to modify the manufacturing procedure and quicken the yield learning curve.

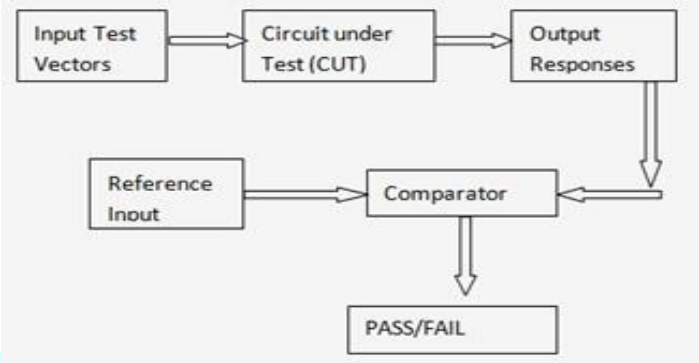

Fig 1. Basic Principle of Digital testing

3.2 Digital Test Methodologies: ATE vs. BIST

The fundamental idea behind production testing is depicted in Figure 1. A chip's circuit under test (CUT) may consist of the complete chip or just a portion of it (example., the logic memory block and core memory) as in Fig.1. The values observed on the CUT's outputs are the related output response to the CUT's inputs are subjected to binary patterns denoted as that of the input test vectors. External testing additionally to internal testing by using built-in self-test using automatic test equipment (ATE). The required input test data vectors and the accurate answer The ATE memory retains data when external testing is used. Using ATPG tools, input test vectors are generated, and circuit modelling is used to get accurate response data. For external testing, the tester is used for the comparison. Despite the fact that ATE-based testing was the norm In the past, whenever chip to port ratio and circuit operating frequencies expanded, here an expanding disparity circuit test and ATE capabilities standards.

3.3 System-on-a-Chip Test Challenges

System-on-a-chip (SoC) designers are capable of combining most of the active constituents contained in a conventional system-on-a-board (SOB) into a single silicon die as manufacturing technologies keep on developing. To accomplish this, a chip must be constructed using the reasons for switching from ATE-based SOC testing to BIST are discussed here along with a number of SOC test challenges.

As feature sizes shrink and functionality—from digital cores and embedded memories to analog /RF blocks and third-party IP—gets crammed onto one die, traditional ATE-based testing runs into serious roadblocks. External Automatic Test Equipment (ATE) simply cannot reach or toggle deeply embedded nodes at the speeds modern blocks require, nor can it store and apply the exploding number of test vectors needed for full fault coverage. At the same time, pin counts and tester memory requirements skyrocket, driving up both capital and per-chip test costs.

Built-In Self-Test (BIST) addresses these shortcomings by moving test pattern generation, application, and response analysis on-chip:

- **On-chip autonomy** – Pattern generators and response analyzers live inside the silicon, so tests run without extensive ATE channels or external vector storage.
- **At-speed operation** – Tests use the chip's own clock trees, assuring that high-frequency and timing-critical paths are exercised under real operating conditions.
- **Vector-volume reduction** – Compression and pseudorandom pattern generators shrink the number of stored vectors, cutting test time and ATE memory needs.

- **Heterogeneous-block support** – Dedicated MBIST engines handle memories, while LBIST structures test logic clusters and custom IP with tailored algorithms.
- **Power-aware scheduling** – Integrated clock-gating and test ordering smooth switching activity, preventing peak-power spikes that can damage silicon or skew results.
- **Field-serviceability** – Embedded BIST can be re-invoked in the field for health monitoring, in-situ diagnostics, and predictive maintenance—capabilities impossible with one-time ATE runs.

3.4 Embedded Memory Testing

Since transistors and/or capacitors are used in the design of memory cells, logic gates cannot accurately represent them. Memory testing cannot be done using a structural test based on a gate level netlist. But still, as was stated in the prior section, identical memory cells and incredibly simple functional activities cause memory cores to have a relatively standard structure (just read and write), making them ideal for testing functionality. Functional testing, as contrast to random logic testing, only requires a small number of deterministic test patterns to provide the appropriate fault coverage, programs for memory cores are able to be produced using scalable along with small on-chip test suite generators. Moreover, because the written data in a fault-free memory is not affected, the predicted outputs may be readily created again on-chip and minimal overhead comparison circuitry can verify the accuracy of output responses. As a result, the memory BIST circuit is less complicated than the logic BIST.

IV. THEORETICAL BACKGROUND OF MEMORY TESTING

The concept behind testing the memory is explained in this chapter. The two types of memory tests are electrical (reliant on technology) and efficient (technology-independent). Electrical memory testing involves dynamic analysis for recovery, retention, and imbalance faults, IDDQ testing of DC and AC parameters, and parametric testing of DC and AC parameters. To ensure that the device satisfies the criteria for its electrical properties, includes voltage, current, and setup and hold time requirements of chip's pins, DC and AC parametric tests are utilized. Parametric testing for embedded memory is not required since they often don't have direct I/O ports attached to chip pins in SOCs as shown in Fig.2. Dynamic testing and IDDQ require a explanation of the particular process technology. In order to confirm the logical functioning of a memory core, this study focuses on technology-independent functional memory testing. As functional memory testing permits the creation of cost- efficient quick test methods, the industry generally acknowledges it as a low-cost/high-quality choice. Functional tests use standardized algorithms such as march sequences, checkerboard patterns, and walking-one/zero vectors to systematically detect address-decoder, coupling, and data-retention faults. These algorithms require only simple on-chip write/read circuitry and comparators avoiding complex analog test hardware and can generate vectors pseudo-randomly or via compression to minimize storage overhead. Integrated BIST controllers tie into existing scan chains to parallelize read/write across banks, preserve scarce I/O pins, and localize errors for targeted redundancy or repair, thereby speeding test time and boosting yield.

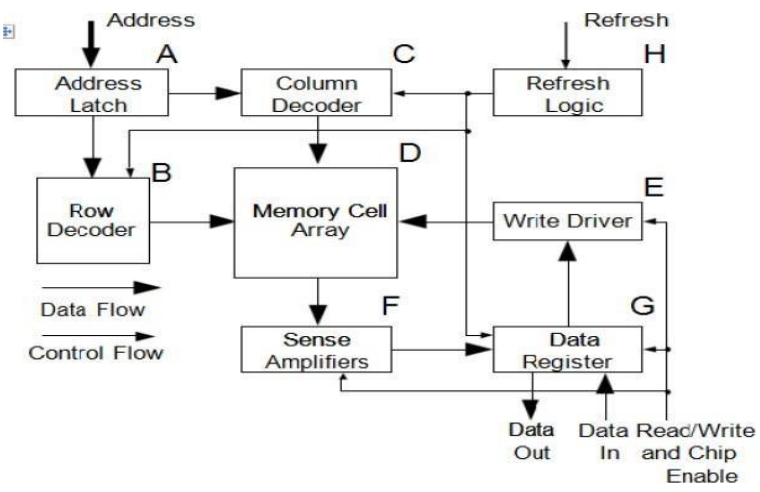

Fig 2. Functional Memory Model

4.1 Fault Combinations

But still, it is highly possible that many various kinds of malfunctions may exhibit simultaneously, while evaluating a memory core. These errors may or may not be interrelated. One fault present in the chain of related faults can have an impact on how other faults behave as shown in Fig.3. An unconnected defect has no effect on how other faults behave. Linked faults can also be divided into those that share the same fault type and those that share a different fault type.

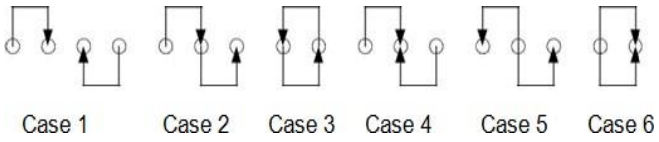

Fig 3. Two Coupling Faults

V. PROPOSED METHOD

BIST is the name of a method through which a machine may verify itself (or BIST It may produce patterns using a range of algorithms, each targeted towards a specific kind of circuitry or defect. There are a few different implementations of the comparison function, including signal detectors with actual comparators. We will develop Memory BIST (MBIST), which employs one or more algorithms created especially for testing memory flaws, in this project. In order to a certain part of circuitry, BIST structures, and output responses create patterns. BIST can be used to whole designs, design blocks, or structures inside of design blocks. Depending on the architecture, pattern generation and output- comparison circuitry may differ. A physical problem that arises during the production technique is referred to as a manufacturing error and results in some sort of gadget malfunction. The test patterns are produced by test generation that can identify as many manufacturing flaws as feasible.

5.1 BIST specifications

With the use of a few pins, BIST is mostly utilized to aid in the testing of memory, which has an incredibly complex architecture (fabrication-wise). Applying a straightforward clock signal and a few pins during a memory test utilizing BIST actually aids in testing the complete memory IC.

The memory model, BIST controller, and test bench that power BIST's whole operation are designed here. Three generators—a data generator, an address generator, and a control generator—drive the finite-state machine in the BIST controller code. Additionally, MUX is utilized to choose whether to run BIST or standard memory operations.

The pattern generator, which will write and read back the same patterns in all memory locations, is used to construct and analyze BIST as shown in Fig.4.

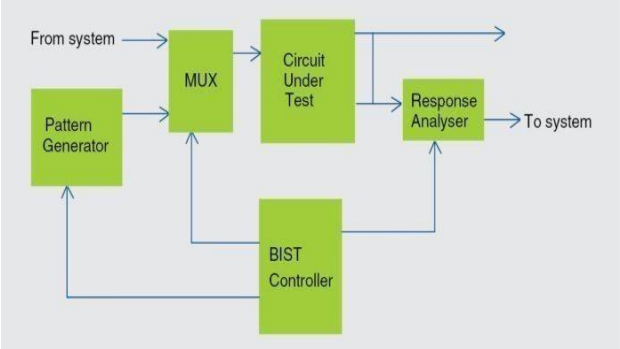

Fig 4. Circuit with surrounding built-in self- test circuitry

We took into account 1024 inputs, each with 32-bit data. Now, a test bench will be written on Model Sim for the memory model and BIST controller in order to illustrate a stuck-type fault model in logic circuits and their pattern creation for read-and-write operations.

VI. RESULTS AND DISCUSSION

8-bit ALU (Fault-Free): Confirmed normal operation without errors.
8-bit ALU (With Faults): Demonstrated successful detection of induced faults using BIST.

- Enabled early fault detection during manufacturing.
- Achieved self-calibration without external devices.
- Supported comprehensive testing across circuit blocks.
- Promoted improved chip yield by identifying faults early.

REFERENCES

[1] D. S. V.Kirthi, "Design of BIST with Low Power Test Pattern Generator," *Journal of VLSI and Signal Processing*, vol. 4, no. 5, pp. 30-39, 2014.

[2] W. S. T. D. M. a. E. H. Yuejain, "Built-In Functional test for silicon validation and system integration of telecom SOC designs," *IEEE trans. Very large scale integration (VLSI) System*, vol. 19, no. 4, pp. 629-637, 2011.

[3] K. R. R. X. a. K. H, "Multiple Output Low-Power Linear Feedback Shift Register Design," *IEEE Trans.circuits & Systems*, vol. 53, no. 7, pp. 1487-1495, 2006.

[4] M. T. N. A. Mehrdad Nourani, "Low-Transition Test Pattern Generation for BIST-Based Applications," *IEEE Transactions on Computers*, vol. 57, no. 3, pp. 303-315, 2008.

[5] V. A. H. S. G. Daware, "Implementation of Combinational Automatic Test Pattern Generator D_Algorithm," *International Journal of Computer Applications*, pp. 32-34, 2014.

[6] P. M, "Implementation of multiplier using vedic algorithm," *International journal of innovative technology & exploring engineering (IJITEE)*, vol. 2, no. 6, pp. 219-223, 2013.

[7] L. M.Padma, "Design and Test of Concurrent BIST Architecture," *International Journal of Computer Science and mobile Computing*, vol. 4, no. 7, pp. 21-26, 2015.

[8] B. M. D. R. J. a. P. R. Saraswat, "Low Power BIST based Multiplier Design and Simulation using FPGA," *IEEE Students Conference on Electrical. Electronics and Computer Science.*

[9] X. W. L.-T. W. Dong Xiang, "Low-Power Scan-Based Built-In Self- Test Based on Weighted Pseudorandom Test Pattern Generation and Reseeding," *IEEE Transactions On Very Large Scale Integration (VLSI) Systems*, pp. 1-12, 2016.

[10] R. J. Govindaraj Vellingiri, "An Improved low transition test pattern generator for low power applications," *Springer Science Business Media, LLC*, 2017.

[11] G. M. N. M. B. N.-D. J. R. J. d. S. J. T. Michał Filipek, "Low- Power Programmable PRPG With Test Compression Capabilities," *IEEE Transactions On Very Large Scale Integration (VLSI) Systems*, 2014.

[12] N. Priya.M, "Area Reduction of Test Pattern Generation Used in BIST Schemes," *International Journal of Engineering Trends and Technology (IJETT)*, vol. 9, no. 13, pp. 687-693, 2014.