

Built-In-Self-Test Using Weighted True Random Pattern Generator

K. Sai SivaJyothi¹, Ch. Divya², D. Jagannadha rao³, M. Ramya⁴,

K. Mani Varaha deepak⁵, Dr. B. RaviChandra⁶

^{1,2,3,4,5} M.Tech Students, Department of Electronics and Communication Engineering,

⁶Associate Professor, Department of Electronics and Communication Engineering,

Nadimpalli Satyanarayana Raju Institute of Technology

ABSTRACT

Built-In Self-Test (BIST) is an effective testing technique used in VLSI systems to improve fault detection while reducing dependence on external testing equipment. As circuit complexity increases, conventional testing methods become expensive and time-consuming. BIST enables circuits to test themselves using on-chip test pattern generation and response analysis. This project proposes a Weighted True Random Pattern Generator (WTPG) integrated with BIST architecture to improve testing efficiency and fault coverage. Unlike conventional pseudo-random generators, the proposed WTPG combines true randomness, Galois feedback, and weighting logic to generate high-quality test patterns with reduced correlation and improved entropy. The generated weighted random patterns are applied to the Circuit Under Test (CUT), and the output responses are analyzed using a Test Response Analyzer (TRA) with MISR. The system improves detection of random-pattern-resistant faults while reducing switching activity and power consumption during testing.

The design is implemented using Verilog HDL and synthesized using Xilinx Vivado.

Simulation results demonstrate improved fault coverage, reduced test time, and better power efficiency compared to conventional BIST architectures

CHAPTER 1 — INTRODUCTION

1.1 Introduction

The continuous advancement of VLSI technology has enabled the design of highly complex

digital systems containing millions of transistors on a single chip. As integration density

increases, testing these circuits becomes more challenging. Efficient testing is essential to

ensure product quality, reliability, and performance.

Traditional external testing methods require expensive Automatic Test Equipment (ATE) and

large testing time. These methods become inefficient for modern VLSI circuits due to

increasing complexity and limited access to internal nodes.

Built-In Self-Test (BIST) addresses these challenges by incorporating testing hardware directly

into the chip. It allows circuits to generate test patterns internally and analyze output

responses without heavy dependence on external testers.

1.2 Need for BIST

Testing modern integrated circuits faces several challenges:

Increased circuit complexity

Limited accessibility of internal nodes

☐ High testing cost

Longer testing time

☐ Higher power consumption during testing

BIST provides a practical solution by enabling self-testing capabilities within the chip.

Main advantages of BIST:

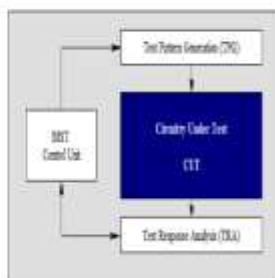
☐ Reduced testing cost

☐ Faster testing process

Improved fault coverage

☐ Reduced dependency on external test equipment

BIST ARCHITECTURE



1.3 Problem Statement

Conventional pseudo-random test pattern generators often produce correlated patterns,

resulting in poor detection of complex faults. They may fail to detect random-pattern-

resistant faults effectively.

Additionally, excessive switching activity during testing increases power consumption and may

affect system reliability.

To overcome these limitations, an improved test pattern generation technique is required.

1.4 Proposed Solution

This project proposes a Weighted True Random Pattern Generator (WTPG) for BIST

applications.

The proposed system:

☐ Generates high-randomness test patterns

☐ Uses Galois feedback for efficient pattern generation

☐ Applies weighting logic for controlled probability distribution

☐ Reduces switching activity

☐

Improves fault detection capability

The WTPG-generated patterns are applied to the Circuit Under Test, and responses are

analyzed using MISR-based Test Response Analyzer.

1.5 Objectives

The main objectives of this work are:

☐ Design an efficient WTPG architecture

Improve fault coverage in BIST

☐ Reduce test power consumption

☐ Minimize hardware overhead

Improve testing efficiency of VLSI circuits

1.6 Applications

The proposed system can be applied in:

☐ VLSI circuit testing

☐ SoC testing

☐ FPGA testing

☐ Memory testing

☐ Safety-critical systems

☐ Embedded systems

CHAPTER 2 — LITERATURE SURVEY

2.1 Introduction

Testing plays a crucial role in ensuring the reliability and functionality of VLSI circuits. With the

rapid growth of circuit complexity, traditional testing methods face limitations in terms of

cost, time, and fault coverage. To overcome these challenges, researchers have proposed

several advanced testing techniques, including Built-In Self-Test (BIST), pseudo-random test

pattern generation, and weighted random pattern generation.

This chapter presents a review of existing research related to BIST architectures, test pattern

generators, and fault detection techniques in VLSI testing.

2.2 Built-In Self-Test (BIST)

Built-In Self-Test is a testing technique in which test generation and response analysis are

integrated within the chip. It reduces dependency on external Automatic Test Equipment

(ATE) and enables faster testing.

A typical BIST system consists of:

- ☐ Test Pattern Generator (TPG)
- ☐ Circuit Under Test (CUT)
- ☐ Test Response Analyzer (TRA)
- ☐ BIST Control Unit

Researchers have shown that BIST improves testing efficiency, reduces test cost, and provides

better fault coverage for complex digital systems.

Advantages of BIST:

- ☐ Reduced external testing requirements
- ☐ Lower testing cost

- ☐ Faster fault detection

- ☐ Improved reliability

However, conventional BIST architectures may suffer from limited randomness and high

switching activity.

2.3 Pseudo-Random Test Pattern Generators

Pseudo-random pattern generators, such as Linear Feedback Shift Registers (LFSRs), are

widely used in BIST systems. They generate test patterns with simple hardware and low area

overhead.

Common advantages include:

- ☐ Easy implementation
- ☐ Low hardware complexity
- ☐ Fast pattern generation

Despite these benefits, conventional pseudo-random generators have limitations:

- ☐ Pattern correlation
- ☐ Reduced randomness
- ☐ Poor detection of random-pattern-resistant faults

These limitations reduce fault coverage in advanced VLSI circuits.

2.4 Weighted Random Pattern Generation

Weighted random pattern generation improves testing by controlling the probability of logic

‘1’ and logic ‘0’ in test patterns. This increases the likelihood of activating difficult faults.

Key benefits include:

- ☐ Improved fault activation
- ☐ Better fault coverage
- ☐ Reduced number of test patterns
- ☐ Lower test time

Lower test time

Weighted pattern generation is particularly useful for detecting faults that are difficult to

identify using purely random patterns.

2.5 True Random Pattern Generation

True Random Pattern Generators (TRNGs) use physical randomness sources such as:

- ☐ Thermal noise
- ☐ Clock jitter
- ☐ Metastability
- ☐ Environmental noise

Unlike pseudo-random generators, TRNGs provide high entropy and non-deterministic

outputs. These characteristics improve randomness quality and reduce pattern repetition.

High-randomness test patterns improve fault detection efficiency in VLSI testing.

2.6 Research Gap

Existing BIST systems mainly rely on pseudo-random pattern generators, which often suffer

from:

- ☐ High correlation
- ☐ Limited randomness

Increased switching activity

Lower fault coverage

Although weighted generators improve fault detection, there is still a need for a system that

combines:

- ☐ True randomness
- ☐ Weight control
- ☐ Efficient hardware implementation
- ☐ Low power consumption

This research addresses this gap by proposing a Weighted True Random Pattern Generator

integrated with BIST.

2.7 Summary

From the literature review, it is observed that efficient test pattern generation is critical for

improving BIST performance. Conventional pseudo-random generators provide simplicity but

suffer from limited randomness and reduced fault coverage.

The proposed WTPG-based BIST architecture overcomes these limitations by combining true

randomness, Galois feedback, and weighting logic to improve testing efficiency and fault

detection performance.

CHAPTER 3 — VLSI OVERVIEW

3.1 Introduction to VLSI

Very-Large-Scale Integration refers to the process of integrating millions of transistors onto a

single chip to build complex electronic systems. VLSI technology is the foundation of modern

digital electronics and plays a critical role in designing compact, high-speed, and low-power

integrated circuits.

The advancement of VLSI has enabled the development of powerful electronic devices with

improved performance, reduced size, and lower power consumption.

Applications of VLSI include:

- ☐ Computers and laptops
- ☐ Smartphones
- ☐ Consumer electronics
- ☐ Medical devices
- ☐ Automotive systems
- ☐ Communication systems

3.2 Evolution of IC Technology

Integrated Circuit (IC) technology has evolved through several stages:

- ☐ SSI (Small Scale Integration): Few transistors per chip
- ☐ MSI (Medium Scale Integration): Hundreds of transistors
- ☐ LSI (Large Scale Integration): Thousands of transistors
- ☐ VLSI: Millions of transistors
- ☐ ULSI (Ultra Large Scale Integration): Billions of transistors

This evolution has significantly increased computing power while reducing cost and physical size.

3.3 Need for Testing in VLSI

As circuit complexity increases, testing becomes more difficult. Manufacturing defects,

process variations, and design errors can affect circuit functionality.

Common faults in VLSI circuits include:

- ☐ Stuck-at faults
- ☐ Bridging faults
- ☐ Delay faults
- ☐ Open circuit faults
- ☐ Short circuit faults

Testing ensures:

- ☐ Functional correctness
- ☐ Reliability
- ☐ Performance
- ☐ Quality assurance

Efficient testing is essential for modern VLSI systems.

3.4 Challenges in VLSI Testing

Testing advanced VLSI circuits involves several challenges:

1. High Complexity

Modern chips contain millions of transistors, making fault detection difficult.

2. Limited Accessibility

Internal nodes are difficult to access using external testing methods.

3. Increased Testing Cost

External test equipment is expensive.

4. Higher Power Consumption

Switching activity during testing increases power dissipation.

5. Longer Testing Time

Large circuits require more test patterns and analysis time.

These challenges motivate the use of advanced testing techniques such as BIST.

3.5 Design-for-Testability (DFT)

Design for Testability is a design methodology used to improve the testability of digital circuits

by adding testing features during the design stage.

Main DFT techniques include:

- ☐ Scan insertion
- ☐ Boundary scan
- ☐ Built-In Self-Test (BIST)

Benefits of DFT:

- ☐ Improved controllability
- ☐ Improved observability
- ☐ Better fault coverage
- ☐ Reduced testing complexity

DFT techniques simplify testing and improve fault detection efficiency.

3.6 Built-In Self-Test in VLSI

Built-In Self-Test is one of the most widely used DFT techniques. It allows the circuit to test

itself using internal hardware.

A typical BIST architecture contains:

- ☐ Test Pattern Generator (TPG)
- ☐ Circuit Under Test (CUT)
- ☐ Test Response Analyzer (TRA)
- ☐ Control Unit

BIST offers several advantages:

- ☐ Reduced dependency on external testers
- ☐ Faster testing
- ☐ Lower cost
- ☐ Improved fault detection

This makes BIST highly suitable for complex VLSI systems.

3.7 Importance of Low-Power Testing

Power consumption during testing is often much higher than during normal operation

because of increased switching activity.

High test power may cause:

- ☐ Excessive heat generation
- ☐ Signal integrity problems
- ☐ Reduced reliability
- ☐ Potential chip damage

Therefore, reducing switching activity during testing is essential.

The proposed WTPG-based BIST system addresses this issue by generating weighted patterns

that reduce unnecessary transitions while maintaining high fault coverage.

3.8 Summary

VLSI technology has revolutionized modern electronics by enabling highly complex and

efficient integrated circuits. However, increasing circuit complexity makes testing more

challenging.

Testing techniques such as DFT and BIST are essential for ensuring reliability and

performance. Efficient test pattern generation plays a major role in improving fault coverage

and reducing testing cost.

The proposed WTPG-based BIST architecture provides an efficient solution for low-power and

high-coverage VLSI testing.

CHAPTER 4 — METHODOLOGY

4.1 Introduction

This chapter presents the methodology used to design and implement the proposed Built-In

Self-Test (BIST) architecture using a Weighted True Random Pattern Generator (WTPG). The

primary objective of the proposed system is to improve fault coverage, reduce testing time,

and minimize power consumption during testing.

Unlike conventional BIST systems that use pseudo-random generators, the proposed design

integrates:

- ☐ True randomness
- ☐ Galois feedback logic
- ☐ Weight generation mechanism
- ☐ Weighted multiplexer selection

This combination improves randomness quality and enhances detection of complex faults.

4.2 Overview of Proposed System

The proposed BIST system consists of four major blocks: BIST Control Unit

- o Weighted True Random Pattern Generator (WTPG)
- o Circuit Under Test (CUT)
- o Test Response Analyzer (TRA/MISR)

□ Working Flow:

1. WTPG generates weighted random test patterns
2. Test patterns are applied to the CUT
3. Output responses are captured by TRA
4. MISR compresses responses into signature
5. Final signature is compared with golden signature

If signatures match → Circuit is fault-free

If signatures mismatch → Fault detected

4.3 Weighted True Random Pattern Generator (WTPG)

The WTPG is the core block of the proposed system. It generates high-quality weighted

random test patterns with better entropy and lower correlation.

The WTPG consists of:

1. Shift Register
2. Galois Feedback Logic
3. True Random Source
4. Weight Generator
5. Weighted MUX

These blocks work together to produce optimized test patterns.

4.4 Shift Register with Galois Feedback

The shift register is built using D flip-flops connected in series. It generates random sequences

using Galois feedback.

In Galois feedback:

Selected tap outputs are XORed

XOR output is fed back to input

Advantages:

1. Fast pattern generation
2. Reduced propagation delay
3. Efficient hardware implementation
4. Better randomness

The feedback polynomial determines the quality and length of generated sequences.

4.5 True Randomness Source

To improve unpredictability, true randomness is introduced using stochastic variations such as:

1. Clock jitter
2. Metastability
3. Internal noise sources

These sources generate non-deterministic behaviour, which increases entropy and improves

randomness quality.

Benefits:

1. Reduced pattern repetition
2. Improved randomness
3. Better fault activation

This overcomes the limitations of purely pseudo-random generators.

4.6 Weight Generator Mechanism

The weight generator controls the probability distribution of generated bits.

It determines:

Probability of logic '1'

Probability of logic '0'

Functions of weight generator:

1. Generate weight control signals
2. Adjust bit probabilities
3. Enable targeted fault activation

This helps detect random-pattern-resistant faults effectively.

Example probability values:

☐ 0.55

☐ 0.65

☐ 0.75

☐ 0.85

These values improve pattern selection based on testing requirements.

4.7 Weighted Multiplexer Logic

A weighted multiplexer selects the final output pattern.

MUX selects between:

Pure random output

Weighted output

Control signals:

WE (Weight Enable)

WA (Weight Adjust)

Working:

If weighting is disabled → random pattern selected

If weighting is enabled → weighted pattern selected

Final output is represented as Yw.

This improves:

1. Fault coverage
2. Power efficiency
3. Pattern quality

4.8 Galois-Based Weighted Pseudorandom TPG

The proposed WTPG uses Galois operation with weighted logic to generate efficient test patterns.

Key advantages:

Reduced switching transitions

Lower power consumption

Fewer hardware components

Improved fault coverage

The weighted pseudorandom generator reduces unnecessary transitions in scan chains.

Compared to conventional TPG:

Better randomness

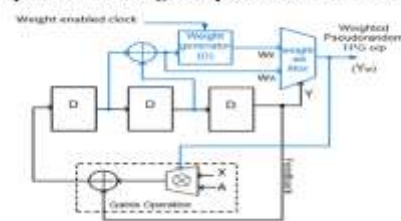
Lower correlation

Higher efficiency

The system generates maximum-length weighted patterns while maintaining reduced

switching activity.

Proposed 3-bit weighted pseudorandom TPG.



4.9 Integration with BIST Architecture

The WTPG is integrated into BIST as the primary Test Pattern Generator.

System operation:

BIST enters test mode

- o WTPG generates weighted test patterns
- o Patterns applied to CUT
- o Responses captured
- o MISR compresses outputs
- o Signature comparison performed

This enables autonomous on-chip testing without external test equipment.

4.10 Test Response Analyzer (TRA)

The TRA analyzes output responses from the CUT.

The TRA uses a Multiple Input Signature Register (MISR) to compress outputs into a compact signature.

Functions:

- ☐ Capture responses
- ☐ Perform XOR-based compression
- ☐ Generate final signature

A mismatch between generated and reference signatures indicates fault presence.

Benefits:

- ☐ Reduced storage requirement
- ☐ Fast analysis
- o Efficient fault detection

4.11 Algorithm of Proposed System

Step-by-Step Algorithm

Step 1: Initialize shift register and weight parameters

Step 2: Enable BIST control unit

Step 3: Generate Galois feedback using XOR logic

Step 4: Introduce true randomness

Step 5: Generate weight control signals

Step 6: Apply weighted selection using MUX

Step 7: Produce weighted output pattern (Yw)

Step 8: Apply pattern to CUT

Step 9: Capture output response

Step 10: Update MISR signature

Step 11: Repeat for N test cycles

Step 12: Compare final signature with golden signature

Step 13: Declare fault-free or faulty

4.12 Design Considerations

The proposed methodology is designed with the following objectives:

- ☐ High entropy
- ☐ Low correlation
- ☐ Optimal weighting
- ☐ Low hardware overhead
- ☐ Power efficiency
- ☐ Scalability

These factors ensure efficient testing performance.

4.13 Advantages of Proposed Methodology

The proposed WTPG-based BIST offers several advantages over conventional systems:

- o Higher randomness than LFSR-based generators
- o Improved fault coverage
 - Reduced test time
- o Better detection of complex faults
- o Lower power consumption
- o Reduced switching activity
- o Efficient hardware utilization

The design is suitable for modern large-scale VLSI systems.

4.14 BIST Architecture

A typical BIST architecture includes:

- o Test Pattern Generator (TPG)
- o Circuit Under Test (CUT)
- o Test Response Analyzer (TRA)
- o BIST Control Unit

Role of Each Block

TPG: Generates test patterns

CUT: Circuit being tested

TRA: Checks output responses

Control Unit: Controls overall operation

This architecture improves testability and reduces testing cost.

4.15 Summary

The proposed methodology introduces a Weighted True Random Pattern Generator

integrated with BIST architecture for efficient VLSI testing.

By combining:

- ☐ True randomness
- ☐ Galois feedback
- ☐ Weighted logic

The system achieves:

- ☐ Improved fault coverage
- ☐ Reduced test time
- ☐ Lower power consumption
- ☐ Better testing efficiency

CHAPTER 5 — VERILOG

5.1 Introduction

Verilog is a Hardware Description Language (HDL) used for modeling, designing, and verifying

digital systems. It plays a major role in VLSI design by allowing engineers to describe hardware

behavior and structure at different levels of abstraction.

In this project, Verilog is used to design and simulate the proposed Weighted True Random

Pattern Generator (WTPG) and BIST architecture.

Main uses of Verilog:

- ☐ Digital circuit design
- ☐ Functional simulation
- ☐ Hardware verification
- ☐ Logic synthesis

5.2 Features of Verilog

Important features of Verilog include:

- ☐ Simple and easy syntax
- ☐ Supports modular design
- ☐ Efficient simulation
- ☐ Suitable for synthesis
- ☐ Widely used in ASIC and FPGA design

These features make Verilog suitable for complex digital circuit implementation.

5.3 Basic Concepts

1. Modules

Modules are the fundamental building blocks in Verilog. Each module represents a hardware block.

Example:

```
module example(input a, input b, output y);  
    assign y = a & b;  
endmodule
```

2. Ports

Ports define the inputs and outputs of a module.

Types of ports:

- ☐ Input
- ☐ Output
- ☐ Inout

Ports enable communication between modules.

3. Data Types

Common data types in Verilog:

- ☐ wire → Represents physical connections
- ☐ reg → Stores values in procedural blocks

Example:

wire a;

reg b;

4. Operators

Verilog supports various operators:

- ☐ Arithmetic: +, -, *, /
- ☐ Logical: &, |, ^
- ☐ Relational: ==, !=, >, <

These operators are used for implementing digital logic.

5.4 Procedural Blocks

Procedural blocks define sequential behavior.

Main procedural blocks:

- ☐ always
- ☐ initial

Example:

always @(posedge clk)

begin

q <= d;

end

These blocks are useful for implementing sequential circuits such as flip-flops and counters.

5.5 Simulation and Verification

Simulation is used to verify whether the designed circuit behaves correctly before hardware

implementation.

Simulation helps in:

- ☐ Checking functionality
- ☐ Detecting errors
- ☐ Validating design behavior

A testbench is used to provide inputs and observe outputs.

Example tools for simulation:

- ☐ Xilinx Vivado
- ☐ ModelSim

5.6 Role of Verilog in Proposed Design

In this project, Verilog is used to implement:

- ☐ Weighted True Random Pattern Generator
- ☐ BIST Control Unit
- ☐ Test Response Analyzer
- ☐ Circuit Under Test

The complete architecture is modeled and verified using Verilog before synthesis.

Advantages in this project:

- ☐ Fast design implementation
- ☐ Easy debugging
- ☐ Accurate functional verification

5.7 Summary

Verilog is an essential HDL for digital design and verification. It enables efficient modeling,

simulation, and synthesis of complex VLSI circuits.

In this project, Verilog is used to implement and verify the proposed WTPG-based BIST

architecture, ensuring correct functionality before synthesis.

CHAPTER 6 — SOFTWARE USED FOR SYNTHESIS

6.1 Introduction

Xilinx Vivado is a design suite used for synthesis, simulation, implementation, and verification

of digital systems, especially FPGA-based designs. It provides an integrated environment for

converting HDL designs into hardware implementations.

In this project, Vivado is used to synthesize and verify the proposed Weighted True Random

Pattern Generator (WTPG) and BIST architecture designed in Verilog.

Main functions of Vivado:

- ☐ Design synthesis
- ☐ Simulation
- ☐ Implementation
- ☐ Timing analysis
- ☐ Power estimation

6.2 Design Flow in Vivado

The design flow in Vivado involves the following steps:

Step 1: Create Project

A new project is created by selecting project name, directory, and project type.

Project types include:

- ☐ RTL Project
- ☐ IP Integrator Project

Step 2: Select Target Device

The target FPGA device or board is selected based on design requirements.

Examples:

- ☐ Artix
- ☐ Kintex
- ☐ Zynq

This step ensures proper hardware mapping.

Step 3: Add Design Sources

Design source files are added to the project.

Files include:

- ☐ Verilog modules
- ☐ Testbench files
- ☐ Constraint files

These files define the hardware architecture and testing environment.

Step 4: Run Synthesis

Synthesis converts HDL code into gate-level representation.

During synthesis, Vivado:

- ☐ Analyzes logic
- ☐ Optimizes design
- ☐ Maps logic into hardware resources

Important outputs:

- ☐ LUT utilization
- ☐ Flip-flop count
- ☐ Resource usage
- ☐ Timing report

Step 5: Implementation

Implementation includes:

- ☐ Placement
- ☐ Routing
- ☐ Optimization

The synthesized design is mapped onto FPGA resources.

This stage ensures the design meets hardware constraints.

Step 6: Simulation and Verification

Simulation verifies design functionality.

The simulation process helps:

- ☐ Validate output behavior
- ☐ Detect functional errors
- ☐ Analyze waveforms

Waveforms are used to confirm correct operation of WTPG and BIST modules.

6.3 Advantages of Vivado

Major advantages of Vivado include:

- ☐ Easy design flow
- ☐ Fast synthesis
- ☐ Efficient resource optimization
- ☐ Accurate timing analysis
- ☐ Good simulation support

These features make Vivado suitable for FPGA-based VLSI design.

6.4 Role of Vivado in Proposed Work

In this project, Vivado is used to:

- ☐ Synthesize WTPG design
- ☐ Generate RTL schematic
- ☐ Analyze technology schematic
- ☐ Simulate waveform outputs
- ☐ Evaluate performance parameters

The main parameters analyzed are:

- ☐ Area
- ☐ Delay
- ☐ Frequency
- ☐ Power

These parameters are used to evaluate system performance.

6.5 Summary

Vivado is an efficient synthesis and simulation tool for digital system design. It provides

complete support for design implementation, verification, and performance analysis.

In this project, Vivado is used to validate the proposed WTPG-based BIST architecture and analyze its performance.

7.1 Introduction

CHAPTER 7 — RESULTS

This chapter presents the results obtained from the implementation and simulation of the

proposed Built-In Self-Test (BIST) system using the Weighted True Random Pattern Generator

(WTPG). The design was implemented using Verilog and synthesized using Xilinx Vivado.

The performance of the proposed design is evaluated using important VLSI parameters such

as:

Area

Delay

Frequency

Power

The obtained results demonstrate the effectiveness of the proposed WTPG-based BIST

architecture.

7.2 RTL Schematic

Register Transfer Level (RTL) schematic represents the logical architecture of the designed

system. It provides a high-level view of the internal modules and their interconnections.

The RTL schematic is used to:

Verify design architecture

Analyze module connectivity

Ensure correct logic implementation

RTL schematic views were generated for:

WTPG

BIST using WTPG

The RTL results confirm that all designed modules are properly interconnected and function

according to the proposed architecture.

7.3 Technology Schematic

Technology schematic represents the hardware implementation of the design in terms of

FPGA resources such as LUTs and logic blocks.

It helps analyze:

Area utilization

Logic mapping

Hardware efficiency

Technology schematic views were generated for:

WTPG

BIST using WTPG

The results show efficient resource utilization with minimal hardware overhead.

7.4 Simulation Results

Simulation verifies the functional correctness of the proposed design.

Simulation waveforms were generated for:

WTPG output

BIST without fault

BIST with fault

1. WTPG Output

The waveform confirms that the WTPG generates weighted random test patterns with

improved randomness and reduced correlation.

2. BIST Without Fault

When no fault is present, the output response matches the expected signature. This indicates

normal circuit behavior.

3. BIST With Fault

When faults are introduced, the output signature changes and mismatches the expected

result. This confirms successful fault detection.

These simulation results validate the functionality of the proposed system.

7.5 Performance Parameters

1. Area

2. Delay

3. Frequency

4. Power

Power consumption indicates energy efficiency. Lower power is desirable for VLSI systems.

The proposed WTPG-based BIST achieves improved performance across all these parameters.

7.6 Advantages of Proposed Design

The proposed WTPG-based BIST system provides several advantages:

1. High Randomness

2. Reduced Correlation

3. Improved Fault Coverage

4. Reduced Test Time

5. Lower Hardware Overhead

6. Power Efficiency

7. Scalability

7.7 Applications

The proposed WTPG-based BIST architecture can be applied in:

VLSI circuit testing

SoC testing

Memory testing

FPGA testing

Embedded systems

Automotive electronics

Medical devices

Safety-critical systems

Its efficient fault detection capability makes it suitable for high-reliability applications.

7.8 Conclusion

The results demonstrate that the proposed Weighted True Random Pattern Generator-based

BIST architecture significantly improves testing efficiency and fault detection performance.

By combining:

True randomness

Galois feedback

Weighted pattern generation

the system achieves:

Higher fault coverage

Lower test power

Reduced test time

Improved overall performance

The proposed design provides an efficient and reliable solution for modern VLSI testing

CONCLUSION

This work presents a Weighted True Random Pattern Generator (WTPG) integrated with a Built-In Self-Test (BIST) architecture to improve digital circuit testing. By combining true randomness with a controllable weighting mechanism, the proposed design generates high-quality test patterns that enhance fault coverage and detect complex faults more effectively than conventional methods. The system also reduces test time, hardware overhead, and dependence on external testing equipment. Its scalability and efficiency make it well suited for modern VLSI and System-on-Chip (SoC) applications, providing a reliable solution for next-generation digital circuit testing

ACKNOWLEDGMENT

We would like to thank Design Space lab, Department of Electronics and Communication Engineering at Nadimpalli Satyanarayana Raju Institute of Technology (NSRIT) for providing the resources and support required for this research work

REFERENCES

- [1] P. H. Bardell, W. H. McAnney, and J. Savir, *Built-In Test for VLSI: Pseudorandom Techniques*. New York, USA: Wiley, 1987.
- [2] M. L. Bushnell and V. D. Agrawal, *Essentials of Electronic Testing for Digital, Memory and Mixed-Signal VLSI Circuits*. Boston, MA, USA: Springer, 2000.
- [3] L.-T. Wang, C.-W. Wu, and X. Wen, *VLSI Test Principles and Architectures: Design for Testability*. San Francisco, CA, USA: Morgan Kaufmann, 2006.
- [4] J. Savir, "Survey of stochastic testing," *IEEE Transactions on Computers*, vol. 33, no. 10, pp. 933–940, Oct. 1984.
- [5] S. Wang and S. K. Gupta, "DS-LFSR: A new BIST TPG for low heat dissipation," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 21, no. 7, pp. 842–850, Jul. 2002.
- [6] X. Kavousianos, K. Chakrabarty, and D. Nikolos, "Test data compression for scan-based BIST using weighted random patterns," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 27, no. 11, pp. 1979–1992, Nov. 2008.
- [7] M. Tehranipoor and N. Ahmed, "Low-power test pattern generation for BIST," *IEEE Design & Test of Computers*, vol. 24, no. 6, pp. 542–553, Nov.–Dec. 2007.
- [8] B. Sunar, W. J. Martin, and D. R. Stinson, "A provably secure true random number generator with

built-in tolerance to active attacks,” *IEEE Transactions on Computers*, vol. 56, no. 1, pp. 109–119, Jan. 2007.

[9] M. Bucci and R. Luzzi, “Design of testable random bit generators,” in *Proc. IEEE Int. Symp. Circuits and Systems (ISCAS)*, 2005, pp. 2751–2754.

[10] A. T. Markettos and S. W. Moore, “The frequency injection attack on ring-oscillator-based true random number generators,” in *Proc. Workshop Cryptographic Hardware and Embedded Systems (CHES)*, 2009, pp. 317–331.

[11] K. Yang, D. Blaauw, and D. Sylvester, “An all-digital edge racing true random number generator robust against PVT variations,” *IEEE Journal of Solid-State Circuits*, vol. 51, no. 4, pp. 1022–1031, Apr. 2016.

[12] S. V. Kumar, C. Stroud, and J. Touba, “Weighted random pattern testing for BIST,” in *Proc. IEEE VLSI Test Symposium (VTS)*, 2003, pp. 255–260.

[13] J. Rajsiki and J. Tyszer, “Test responses compaction in multiple-input signature registers,” *IEEE Transactions on Computers*, vol. 42, no. 6, pp. 707–713, Jun. 1993.

[14] E. Dubrova, *Fault-Tolerant Design*. New York, USA: Springer, 2013.

[15] NIST, “A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications,” Special Publication 800-22, 2010.