Journal of Electronic Testing (2024) 40:691–705
https://doi.org/10.1007/s10836-024-06153-3

RESEARCH

Advancing Low Power BIST Architecture with GAN-Driven Test Pattern Optimization

C. Thangam[1] · **R. Manjith**[2]

Received: 7 March 2024 / Accepted: 4 November 2024 / Published online: 13 November 2024

Abstract
A novel approach to achieve low power consumption during Built-In Self-Test (BIST) operations in Very Large Scale Integrated (VLSI) circuits through the integration of Generative Adversarial Networks (GANs) in the test pattern generation process is presented in this work. The escalating need for energy-efficient VLSI designs necessitates novel approaches to reduce power consumption without compromising fault coverage. This research capitalizes on GANs to create an adaptive and optimized test pattern generator for low-power BIST architectures. GANs comprise a generator and discriminator, with the generator learning to create authentic data and the discriminator distinguishing real from generated data. By employing GANs, the BIST framework captures intricate patterns, enabling custom test patterns for specific circuit attributes. Data collection from circuits under test is followed by pre-processing for GAN training. The trained GAN generates diverse test patterns, optimized for both low power consumption and heightened fault coverage. This GAN-based test pattern generator seamlessly integrates into the BIST architecture, including Ternary Parallel Prefix Tree Adder (TPPTA). Comprehensive evaluations validate the superiority of the GAN-enhanced BIST, showcasing significant power reduction, elevated fault detection, and reduced testing duration.

Keywords Very large scale integrated (VLSI) · Built-in self-test (BIST) · Generative adversarial networks (GANs) · Ternary parallel prefix tree adder (TPPTA) · Circuit under test (CUT)

1 Introduction

The landscape of VLSI circuits has experienced rapid and dynamic growth, driving a compelling need for designs that effectively combine energy efficiency with the ability to detect faults reliably [1]. As these circuits evolve to accommodate greater complexity and functionality, the traditional methods of testing them have encountered significant challenges in finding the right equilibrium between minimizing power consumption and maintaining a high level of fault detection accuracy. At the core of this issue lies a fundamental tension: the aspiration to create VLSI designs that consume minimal power must coexist with the imperative to identify faults within these circuits comprehensively [2–5]. Traditional testing approaches have often struggled to reconcile these two objectives, which lead to suboptimal trade-offs or compromises in terms of energy efficiency and fault coverage.

Addressing this crucial challenge, the proposed work presents an innovative and ground breaking approach that seeks to redefine the background of BIST operations [6–8]. BIST not only streamlines testing processes but also offers opportunities to optimize energy consumption during testing while ensuring robust fault coverage. BIST architectures facilitate self-contained test pattern generation, application, response collection, and analysis, reducing the reliance on external testing equipment and lowering the need for excessive energy consumption [9]. However, traditional BIST methods face their own set of challenges. Static test patterns,

Responsible Editor: E. Amyeen.

✉ C. Thangam
cthangam2001@gmail.com

R. Manjith
manjithkmr@gmail.com

[1] Department of Electronics and Communication Engineering, St. Mother Theresa Engineering College, Vagaikulam, Thoothukudi, Tamilnadu, India

[2] Department of Electronics and Communication Engineering, Dr. Sivanthi Aditanar College of Engineering, Tiruchendur, Thoothukudi, Tamilnadu, India

while simple to implement, might not effectively capture the complexity of modern VLSI circuits, resulting in lower fault coverage [10]. Moreover, generating tailored test patterns that strike the right balance between energy efficiency and fault coverage remains a difficult task.

Logic-BIST [11] employs pseudo-random test patterns via LSFR or cellular automaton to reduce test costs. However, this approach may miss random-pattern-resistant faults, leading to extended test times. Solutions to this include enhancing Logic-BIST through LFSR-reseeding [12], which introduces a level of determinism into Logic-BIST process by periodically changing the seed value of LFSR [13]. However, this approach still necessitate additional hardware for LFSR management and control. Similarly, test set embedding [14], offers another avenue for enhancing Logic-BIST by strategically embedding specific test patterns within the pseudo-random pattern sequence while maintaining the benefits of pseudo-random testing [15]. This approach strikes a balance between determinism and randomness, contributing to more efficient fault detection.

The bit-flipping [16, 17], provide further sophistication in deterministic test pattern generation within Logic-BIST by selectively flipping bits in generated patterns, thereby which the design tests that target specific circuit behaviors or faults. While this approach enhances test effectiveness, it require additional storage and decoding components to manage the complex pattern information effectively. Pseudo random pattern generator in [18], thoroughly exercise the circuit's functionalities and uncover potential faults. While they offer better fault coverage than deterministic patterns, they miss restrained faults and complex interactions.

The TPG has various output modes, including pseudorandom-weighted, which enhances fault coverage. It produces random yet repeatable patterns in each clock cycle. Over 'n' scan chain cycles, one seed bit is needed for each test pattern. Recent research [19] focused on reducing switching activity during scan shifts and automatic parameter selection for low power. However, prior methods, as implemented in [20, 21], adds XOR transitions between shift registers, consuming more power and area. These limitations are addressed in the proposed design by machine learning techniques.

In [22] for online self-testing TPG is performed using DNN accelerators. It uses pseudorandom and structured test patterns to finds faults but has limitations in detecting certain types of non-functional faults. The existing technique [23] for Static Random Access Memory, combining Deep Q-learning for fault injection and MBIST for fault detection. It uses bit swapping LFSR-based BISR for cell repair. While promising, this approach have computational requirements for Deep Q-learning and face limitations when there are numerous faulty cells to repair. In [24] a method using PSO and DNN for efficient fault coverage prediction and testing is proposed. While effective, its performance vary depending on the circuit characteristics, making it less universally applicable. In [25] a novel framework for designing reliable SoCs that adapt to aging-induced degradation. It uses LBIST and Machine Learning to activate countermeasures in the field. While effective with low overhead, it may have limitations in complex or variable SoC designs. As a consequence, the proposed work is designed with the goal of overwhelming the limitations of exiting literatures. The contributions of proposed framework involves:

- Implementation of GAN capture intricate patterns within circuits and enhanced fault detection capabilities, particularly for complex and unintended faults that are missed.
- Optimization of GAN-generated patterns tackles the challenge of minimizing energy consumption while ensuring thorough fault coverage.
- Integration of GAN with TPPTA facilitates efficient translation of GAN patterns into circuit-compatible input signals while minimizing power consumption for rapid confidence test.

Section II provides a description of the proposed system, followed by Section III on modeling, Section IV on the results of the proposed work, and Section V on conclusion.

2 Description of Proposed System

The envisioned BIST architecture represents a pioneering advancement in VLSI circuit testing, leveraging the integration of innovative components: the GAN-based test pattern generator with TPPTA. Figure 1 illustrates the proposed BIST Architecture.

The BIST process begins with initialization phase, at which the circuit is prepared for testing. Registers, memory elements, and other test-related components are configured to enable BIST operation. In pattern generation phase, the architecture generates test patterns that will be applied to CUT. The proposed GAN-based TPG, in harmony with TPPTA, generates diverse test patterns optimized for both low power consumption and increased fault coverage. This shapes a testing background that not only aligns with modern VLSI complexities for effective circuit testing methodologies. Afterwards, CUT is applied with the generated test patterns. These patterns stimulate the circuit's various components, such as logic gates, memory cells, and interconnects. As the patterns propagate through circuit, they reveal its behavior and highlight any deviations from expected outcomes. As the test patterns traverse the circuit, responses

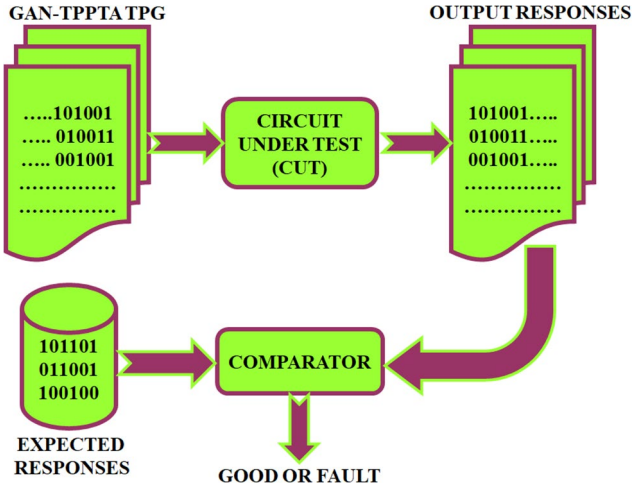

Fig. 1 Proposed system architecture

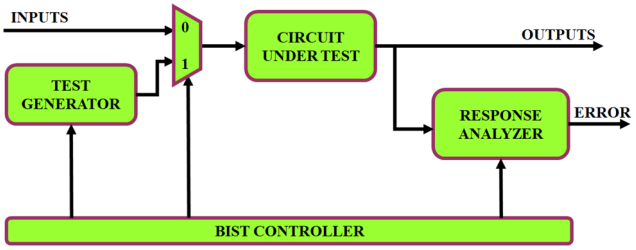

Fig. 2 BIST architecture

generated by circuit are collected. The BIST architecture records these responses for subsequent analysis. In Output Response Analyzer (ORA) phase, the collected responses are compared against expected outcomes. The expected responses are determined based on the design specifications of circuit. Any discrepancies between observed and expected responses are indicative of faults or defects within the circuit. Based on the response analysis, BIST architecture identifies potential faults or defects within the circuit. Different patterns might trigger different fault types, enabling the architecture to pinpoint the locations and nature of the faults. Through interaction between GAN and TPPTA, the generated patterns are efficiently translated into input signals for testing circuits. Once faults are identified, the BIST architecture generates a report detailing the types and locations of detected faults. This information is crucial for subsequent repair and maintenance processes.

3 Modelling of Proposed Work

3.1 BIST Approach

BIST is a method in electronics to internally test and diagnose the health of circuits and systems, eliminating the need for external testing equipment. Figure 2 illustrates the BIST architecture. It's widely used in manufacturing and for maintenance and troubleshooting of electronic components and systems.

3.1.1 BIST Controller

A key component in a BIST system, responsible for managing and coordinating the self-testing process within an integrated circuit. The BIST controller is typically implemented as a dedicated hardware block on the chip and is responsible for several crucial functions.

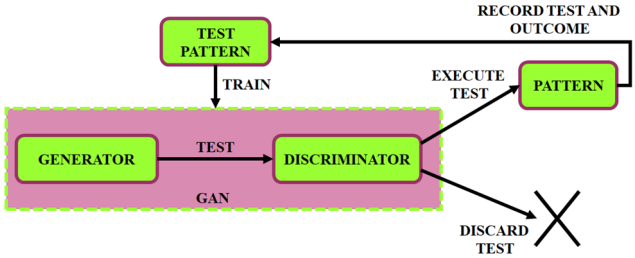

Fig. 3 Test pattern generation using GAN

3.2 GP Using Generative Adversarial Network (GAN)

The generation of test patterns using a GAN is a novel and advanced approach to create test patterns that effectively stimulate the circuit's components and detect faults or defects. A GAN consisting of two neural networks, generator, and discriminator, which are trained instantaneously through a competitive process, as shown in Fig. 3. The generator network generates data, in this case of test patterns, while discriminator network tries to distinguish between real and generated patterns. The two networks engage in a continual feedback loop, with generator trying to improve its capacity to produce realistic data, and discriminator trying to get better at distinguishing real from generated data.

3.2.1 Training GAN

Training a GAN for test pattern generation involves a process of adversarial learning. Initially, the generator produces random or noise-based test patterns based on an initial input seed or noise vector. These patterns are usually far from resembling actual test patterns. These generated patterns are then fed into discriminator along with real test patterns from training set. The discriminator's role is to classify whether each pattern is real or fake. The generator receives feedback from the discriminator's performance and adjusts its parameters to produce test patterns that are more difficult for the discriminator to distinguish as fake. The primary optimization goal of the GAN during training is to maintain reduced

power consumption. However, a balance is maintained between fault coverage and lower power consumption by the inclusion of both objectives in the training process. This indicates that an iterative approach is used where initial patterns are optimized for one objective and then refined in subsequent iterations to address the other objective.

The discriminator network is simultaneously trained to improve its ability to differentiate between real and generated test patterns. It acts like a binary classifier that outputs a probability score, indicating whether a given pattern is real or generated. It receives the real test patterns and the generated test patterns as the inputs. It also receives feedback from the generator's performance. The discriminator typically uses standard neural network architecture. The hidden layers of the network extract features from the input pattern. These layers also reduce the high-dimensional input to a lower-dimensional space where meaningful distinctions between real and fake patterns are made. Here, a binary cross-entropy loss function is used for comparing the predicted output with the actual label. The label is 1 for real pattern and 0 for fake pattern. Each layer applies an activation function that introduces non-linearity, enabling the discriminator to model complex patterns and relationships between features in the input data. The last layer of the discriminator outputs a probability score between 0 and 1. The training process remains iteratively, with generator and discriminator competing to improve their performance. Over time, the generator gets better at creating test patterns that closely resemble real ones.

3.2.2 Test Pattern Generation by GAN

To generate a test pattern, an initial input seed or noise vector is provided to the generator. This vector is typically random or semi-random. The generator takes the input seed and produces a test pattern. Each layer of the neural network applies transformations to the input noise vector using a combination of learned weights, biases and activation functions. The ReLU activation function introduces non-linearity, allowing the generator to model complex patterns and relationships in the data. As the noise vector propagates through the layers of the generator network, it is transformed into progressively more structured data, resembling test patterns. This pattern is essentially a set of inputs that is applied to CUT. After applying the generated test patterns to the circuit, responses are collected. These responses are compared to the expected outcomes based on design specifications of VLSI circuit. Any discrepancies between observed and expected responses indicate potential faults or defects in the circuit.

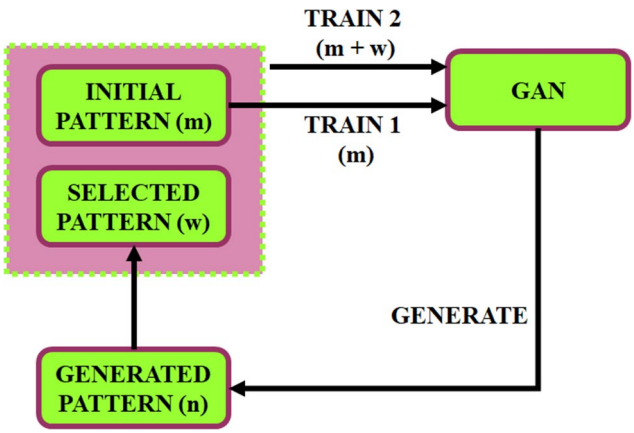

Fig. 4 Generation of test pattern using software testing

GAN is used in the framework to generate test inputs for software testing iteratively. The framework's structure, which consists of the following four processes in order, appears in Fig. 4:

3.2.2.1 Step 1 Initial training of data is done by selecting m test inputs, each with its corresponding execution paths.

3.2.2.2 Step 2 Train GAN using collected training data from Step 1, which generates n additional data points.

3.2.2.3 Step 3 Select $w(w < n)$ inputs data from n data sets that encompass not-executed branches in training data based on path information of generated data.

3.2.2.4 Step 4 Incorporate w selected data points from Step 3 into training data along with their corresponding executed paths. Repeat the process by returning to Step 2 to train GAN further with the expanded training data.

3.3 Ternary Parallel Prefix Tree Adder (TPPTA)

The TPPTA being proposed features a four-stage structure, distinct from the three-stage structure in a typical prefix adder. This structure enables the summation of three binary input operands through several stages, encompassing bit-addition, base, PG (propagate and generate) and sum logic. These four steps have the following logical expressions for each:

In the first stage, known as the Bit Addition Logic:

The sum (S_i') for each bit (i) is calculated as follows:

$$S_i' = a_i \oplus b_i \oplus c_i \tag{1}$$

The carry-out (cy_i) for each bit (i) is determined by:

$$cy_i = a_i.b_i + b_i.c_i + c_i.a_i \tag{2}$$

In this phase, the binary addition of three input operands is performed using full adders, producing sum and carry signals.

In second stage, referred to as Base Logic:

The generate ($G_{i:i}$) and propagate ($P_{i:i}$) signals are computed as follows:

$$G_{i:i} = G_i = S_i'.cy_{i-1}\,,\ \ G_{0:0} = G_0 = S_0'.C_{in} \tag{3}$$

$$P_{i:i} = P_i = S_i' \oplus cy_{i-1},\ \ P_{0:0} = P_0 = S_0' \oplus C_{in} \tag{4}$$

This stage involves the use of sum (S_i') and carry-out (cy_i) signals from Bit Addition Logic stage to generate (G_i) and propagate (P_i) signals, incorporating the external carry input (Cin) for the initial saltire-cell computation.

In the third stage, known as PG (Generate and Propagate) Logic:

The generate ($G_{i:j}$) signal is determined by:

$$G_{i:j} = G_{i:k} + P_{i:k}.G_{k-1:j} \tag{5}$$

The propagate ($P_{i:j}$) signal is calculated as:

$$P_{i:j} = P_{i:k}.P_{k-1:j} \tag{6}$$

This stage involves the computation of generate and propagate signals using a amalgamation of black and grey cell. The logical relationships capture the interaction between adjacent stages in the adder architecture.

In fourth stage, designated as Sum Logic:

The sum (S_i) is calculated as:

$$S_i = (P_i \oplus G_{i-1:0}) \ \ where\ \ S_0 = P_0 \tag{7}$$

The carry-out (C_{out}) is directly acquired from carry generate bit:

$$C_{out} = G_{n:0} \tag{8}$$

This stage finalizes the addition process, computing the sum bits based on propagate and generate signals from previous stages, and the carry-out is derived directly from last generate bit.

The proposed VLSI architecture for TPPTA is illustrated in Fig. 5. This architecture is designed to handle three n-bit binary inputs in a distinctive manner, utilizing a four-stage process.

During the initial stage (bit-addition logic), the n-bit binary input operands undergo bitwise addition by an array of full adders. Each calculates the "sum" and "carry" signals, as emphasized in Fig. 5. Stage-1 defines expressions for sum and carryin logical form with corresponding bit-addition logic depicted in logical diagram presented in Fig. 6.

In preliminary stage, "sum (S_i')" output from current full adder and "carry (cy_i)" output from its right-adjacent full adder collaborate to estimate generate(G_i) and propagate (P_i) signals in second stage, referred to as base logic. Calculation of these signals is symbolized by "squared saltire-cell," exemplified in Fig. 6, with a total of $n+1$ saltire-cells in base logic stage. Logical diagram of saltire-cell, depicted in Fig. 6, is recognized through

$$G_{i:i} = G_i = S_i'.cy_{i-1} \tag{9}$$

$$P_{i:i} = P_i = S_i' \oplus cy_{i-1} \tag{10}$$

Proposed TPPTA technique incorporates the external carry-input signal (C_{in}). This extra carry-input signal, denoted asC_{in}, is factored into the computation of $G_0\,(S_i'.C_{in})$ within first saltire-cell of base. The third stage, referred to as "generate and propagate logic" (PG), the carry bit is pre-computed through a mixture logics of black and grey cell. Logic illustration of black and grey cells is presented in Fig. 6. The expressions become:

$$G_{i:j} = G_{i:k} + P_{i:k}.G_{k-1:j} \tag{11}$$

$$P_{i:j} = P_{i:k}.P_{k-1:j} \tag{12}$$

The proposed adder involves a total of $(log_2 n + 1)$ prefix computation stages. As a consequence, critical path delay of adder is predominantly impacted by progression of carry propagate chain. In concluding stage, known as sum logic, "sum (S_i')" bits are calculated based on carry generate $G_{i:j}$ and carry propagate P_i signals by:

$$S_i = (P_i ? G_{i-1:0}) \tag{13}$$

Furthermore, carry-out signal (C_{out}) is directly derived from final carry generate bit ($G_{n:0}$). The overall enactment of adder is primarily influenced by number of prefix stages in the PG logic. Consequently, maximum propagation gate delay (T_{prop}) of TPPTA architecture is assessed as follows:

$$T_{prop} = T_{bitadd} + T_{base} + T_{PG} + T_{sum} \tag{14}$$

$$\approx\ 2T_X + T_X + 2\,[log_2 n' + 1]\,T_G + T_X \tag{15}$$

$$\approx\ 4T_X + 2\,[log_2 n' + 1]\,T_G \tag{16}$$

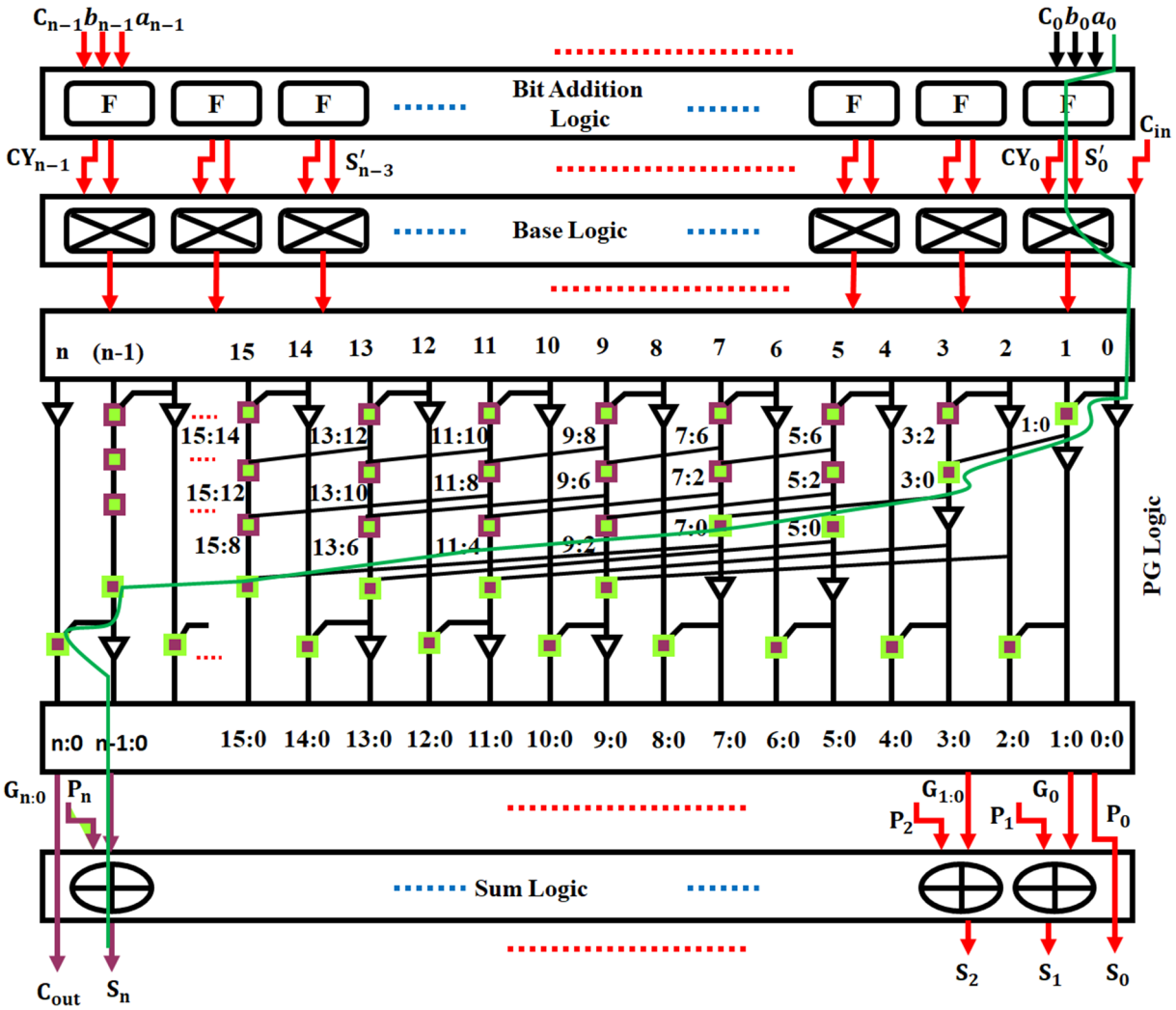

Fig. 5 First order of proposed TPPTA VLSI architecture

Likewise, the hardware area (A_{propo}) for proposed TPPTA is approximated as:

$$A_{prop} = A_{bitadd} + A_{base} + A_{PG} + A_{sum} \tag{17}$$

$$\approx (2nA_X + 3nA_G) + (n+1)(A_X + A_G) + \left[2n + 3s\left[\frac{n}{2}\right] - 3 \times 2^s + 3\right] A_G + nA_X \tag{18}$$

$$A_{prop} \approx (4n+1)\,A_X + \left[6n + 3s\left[\frac{n}{2}\right] - 3 \times 2^s + 4\right] A_G \tag{19}$$

In the context, where $s = [log_2 n - 1]\, and n' = n - 1$.

3.4 Optimization of TPPTA by GAN

The integration of GANs into TPPTA architecture introduces a novel approach for enhancing its performance. GANs, known for their capability to generate authentic and intricate data patterns, are employed to optimize various aspects of the TPPTA, including power consumption, fault coverage, and overall efficiency in fault detection within digital circuits. The optimization process involves leveraging GANs to generate test patterns that extend beyond conventional pseudo-random patterns. The goal is to adapt these patterns for specific testing requirements, optimizing for low power consumption and high fault coverage. Iterative training of GANs facilitates creation of realistic test patterns by continuously refining the generator's ability to produce data that challenges discriminator's capacity to distinguish between real and generated patterns. The GAN-optimized TPPTA combination proves beneficial in addressing the increasing demand for energy-efficient VLSI circuit designs. The generated test patterns strike a balance between power efficiency and fault detection capabilities, showcasing the effectiveness of this approach in semiconductor testing and optimization.

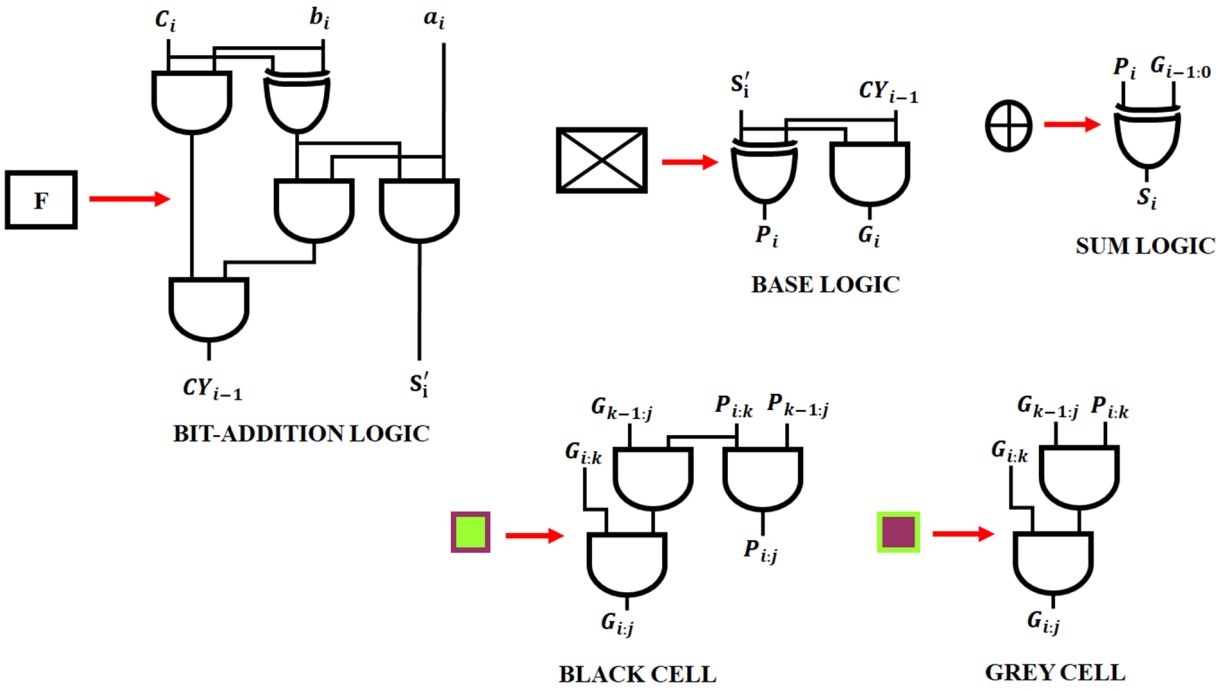

Fig. 6 Logical representation of bit addition, base logic, sum logic, black-cell and grey-cell

This approach enhances the testing process, efficiently addressing the growing need for energy-efficient VLSI circuit designs. The GAN-optimized TPPTA combination enables the generation of highly customized and efficient test patterns that balance power efficiency and fault detection capabilities. This makes it a valuable asset in contemporary semiconductor testing and optimization. A key requirement for training GANs is access to a substantial and high-quality dataset of real test patterns. These patterns are needed to train the discriminator, which helps the generator learn how to produce realistic patterns. In this GAN-TPPTA BIST setup, the generator needs to produce test patterns that not only maximize fault coverage but also minimize power consumption. The GAN must learn to generate patterns that explore different circuit paths and trigger potential faults, ensuring the test coverage of hard-to-detect faults. Simultaneously, the generator must learn to avoid patterns that cause excessive power usage, adding complexity to the training process. Also, the generator must learn to create patterns that effectively test the TPPTA, ensuring that the adder's performance under ternary conditions is well covered.

3.4.1 Response Analyzer

The response analyzer is a critical component responsible for assessing the behavior and correctness of a CUT based on the responses it produces during testing. The BIST architecture generates a set of test patterns for CUT, in which the patterns are designed to detect faults or defects within the circuit. The responses results in the form of digital values, analog signals, or other forms of output, depending on the nature of the circuit. The response analyzer captures and stores these responses, often in a designated memory or storage area.

The primary function of the response analyzer is to compare the actual responses with the expected responses. Any discrepancies or differences between the two indicate potential faults or defects in the circuit. The response analyzer identifies and flags these discrepancies for further analysis. Based on the comparisons, the response analyzer makes a pass/fail decision regarding CUT. If the actual responses match the expected responses within acceptable tolerance levels, the circuit is considered to have passed the test. Otherwise, it is marked as failed, and further diagnostic steps may be taken to pinpoint the exact fault location.

4 Results and Discussion

Proposed BIST architecture is tested on FPGA platform by Xilinx ISE Design Suite 13.4 software for comprehensive evaluation, to assess its functionality and performance. The FPGA underwent rigorous testing to verify the effectiveness of the embedded self-test mechanisms, ensuring accurate fault detection and diagnosis. To assess power consumption, X Power tool is utilized, which provided insights into power usage based on the simulation files. These results are crucial for understanding the energy efficiency of our design and

optimizing power management strategies for the proposed FPGA device. In this proposed design, 2000 patterns with 50,000 cycles are used.

The proposed BIST architecture illustrated in Fig. 7 consists of TPG, response analyzer and Controller. The GAN-TPPTA generated patterns are fed into FPGA, while the RA compares actual responses to expected ones, detecting faults. The controller manages the entire process, making this architecture self-contained and effective for FPGA testing.

The RTL (Register-Transfer Level) view of GAN optimized TPPTA based Test Pattern Generator illustrates the interconnections of registers, multiplexers, and logic gates necessary to generate optimized test patterns for efficient fault detection in digital systems, shown in Fig. 8. The GAN optimization likely introduces additional components to the conventional TPPTA, enhancing its capability to generate low-power and effective test patterns.

The an in-depth depiction of proposed BIST architecture's control process is illustrated with RTL view in Fig. 9, This demonstrates, the components, such as state machines, control logic, and interfacing elements, which manage the execution of test patterns, coordination between the TPG and Response Analyzer, and overall control flow. The BIST Controller is pivotal in ensuring the seamless operation of BIST architecture and plays a crucial role in initiating, monitoring, and concluding the testing procedures.

The Fig. 10 typically illustrates the specific bit patterns that are fed into the FPGA system during testing process. This input pattern defines the test conditions and scenarios used to evaluate system's performance, functionality, or fault tolerance. It acts as a key reference point for assessing the system's behavior and responses under various test cases.

This visual representation in Fig. 11 likely includes the interactions between GAN and TPPTA components, showcasing the GAN-driven optimization influences TPPTA's output. The ground noise in input pattern is reduces in this stage. It demonstrates the integration of machine learning techniques, like GANs, with test pattern generation create efficient and low-power test patterns for fault detection and system evaluation.

The schematic representation in Fig. 12 shows a clear and expected pattern generated by FPGA. It serves as a reference to showcase the ideal response when the system operates without any issues or faults. The output shown in

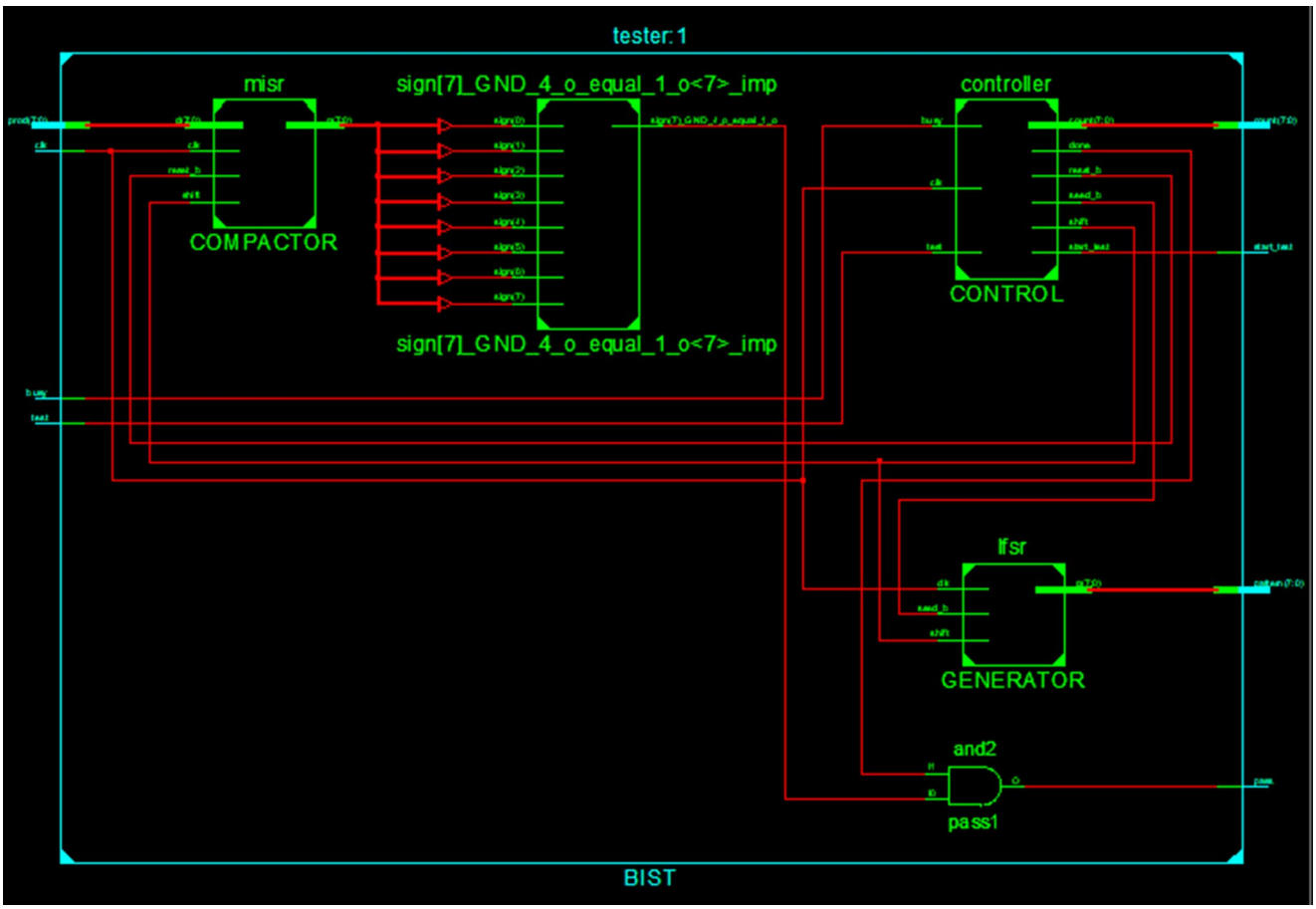

Fig. 7 Schematic representation of proposed BIST architecture

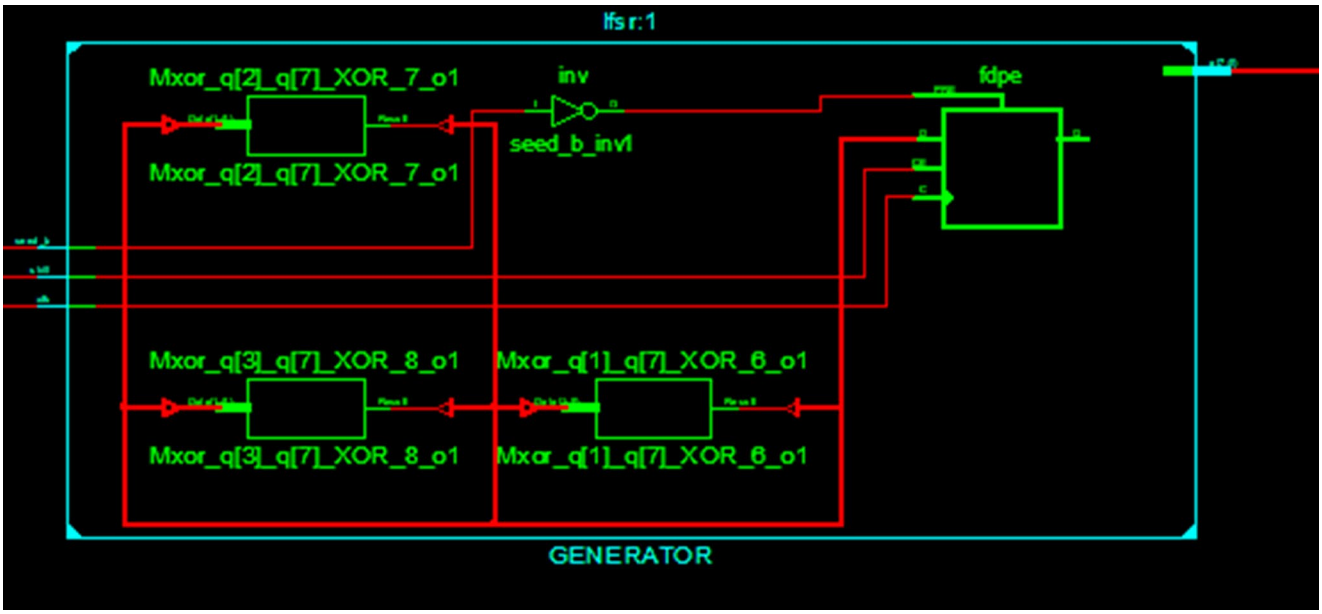

Fig. 8 RTL view of GAN optimized TPPTA based TPG

figure illustrates that there are no faults or errors detected during the testing process.

The hardware setup of the proposed work is provided in Fig. 13. The hardware implementation of the proposed low-power BIST architecture, incorporating GANs, consists of key integrated components. The core of the architecture includes a GAN-based TPG, embedded within the traditional BIST framework, specifically optimized for low-power consumption and enhanced fault detection. The generator module within the GAN is responsible for producing optimized test patterns based on pre-processed data from the circuits under test, while the discriminator evaluates the quality and authenticity of these patterns to ensure high fault coverage. These test patterns are applied to the CUT through existing test access mechanisms. A TPPTA is utilized for the efficient management of signal propagation during test operations, ensuring minimal power usage while maintaining test accuracy. Additionally, the system includes memory buffers for storing test patterns, a test controller for managing the overall test sequence, and feedback loops for dynamic pattern adaptation based on GAN outputs. The seamless integration of the GAN components with the TPPTA hardware ensures the generation of highly optimized patterns for testing, reducing both power consumption and testing time, while achieving high fault coverage in VLSI circuit designs.

The Table 1 presents a comparative analysis of different Low Power BIST methods, focusing on their power consumption and delay characteristics. Among the methods evaluated, Decompressor MPC [10] demonstrates a moderate power consumption of $110.64mW$ but exhibits a relatively high delay of$7.528ns$, making it suitable for scenarios where power efficiency takes precedence over latency concerns. In contrast, BS approach of [26] consumes significantly more power at $1114.30mW$ but offers a low delay of$0.904ns$, prioritizing fast testing over power conservation. MISC [27] strikes a balance between power consumption and delay, with a consumption of $215mW$ and a relatively low delay of$0.170ns$, rendering it a versatile choice for both low-power and efficient testing. MPC [28] showcases moderate power consumption$129.75mW$, necessitating further assessment. Among all existing techniques, the proposed GAN optimized method presents an appealing proposition with a power consumption of $94.11mW$ and an impressively low delay of$0.019ns$, positioning it as an excellent option for applications where both low power consumption and minimal delay are paramount.

Table 2 offers a comprehensive comparative analysis of fault coverage percentages for several fault detection techniques. The technique denoted on [17] demonstrates a good fault coverage of 80.95%, signifying its ability to identify a substantial portion of faults within the tested system. Building on this [16], exhibits an even higher fault coverage of 91.63%, indicating an enhanced fault detection capability compared to [17]. The approach in [5] takes a further step forward with a fault coverage of 96%, marking significant progress in fault detection effectiveness. However, the proposed GAN optimized technique outstand with an exceptional fault coverage of 98.21%. This remarkable achievement suggests that the proposed method excels in fault detection and diagnosis, making it a compelling choice

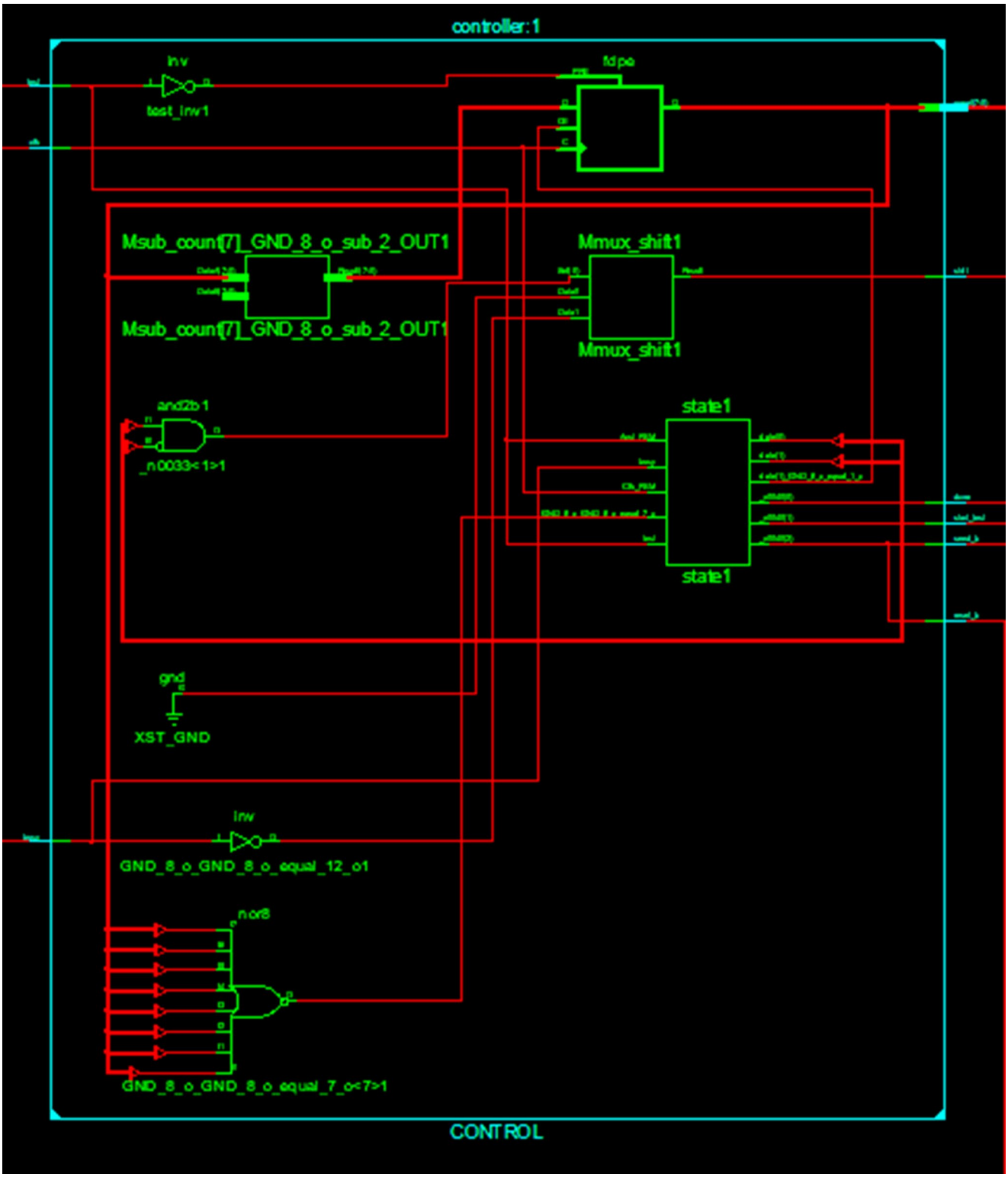

Fig. 9 RTL view of BIST controller

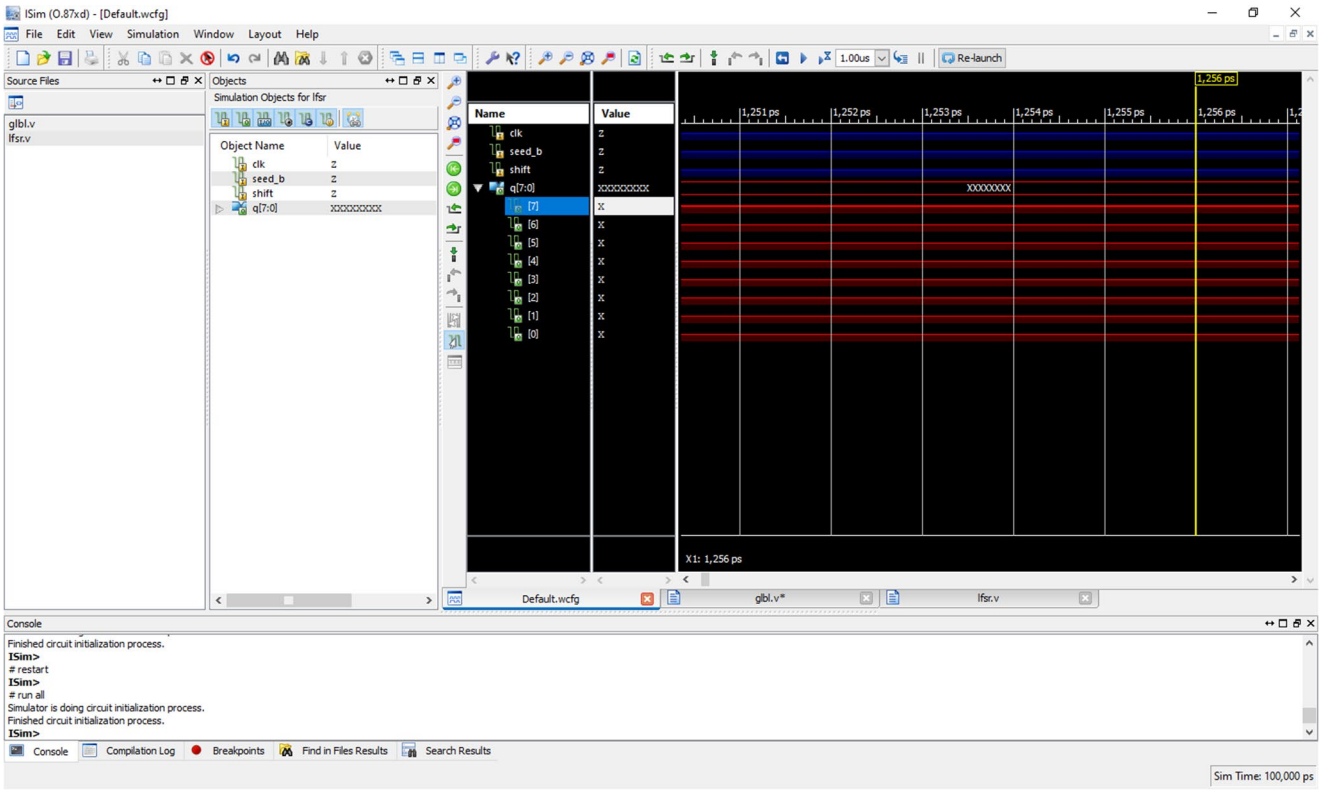

Fig. 10 Input pattern for testing

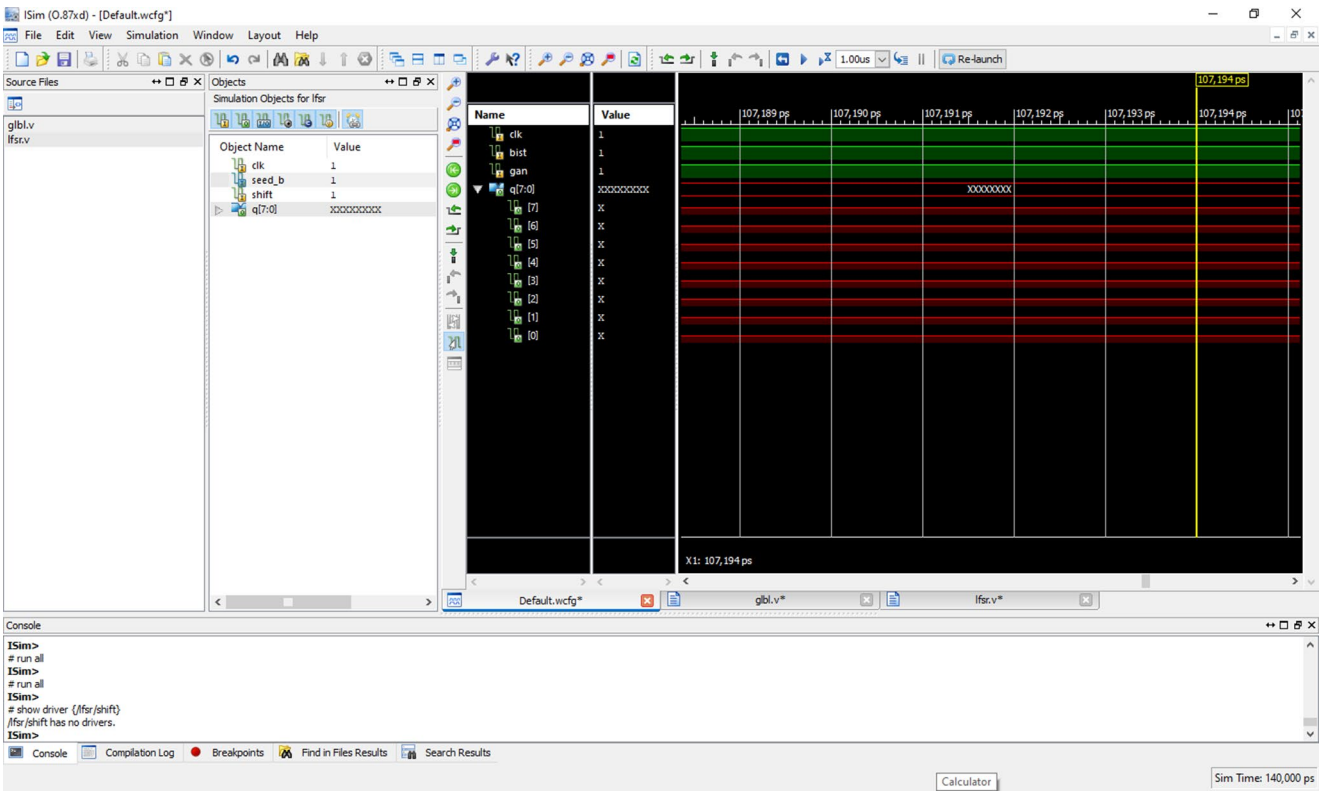

Fig. 11 Generation of test pattern using GAN-TPPTA

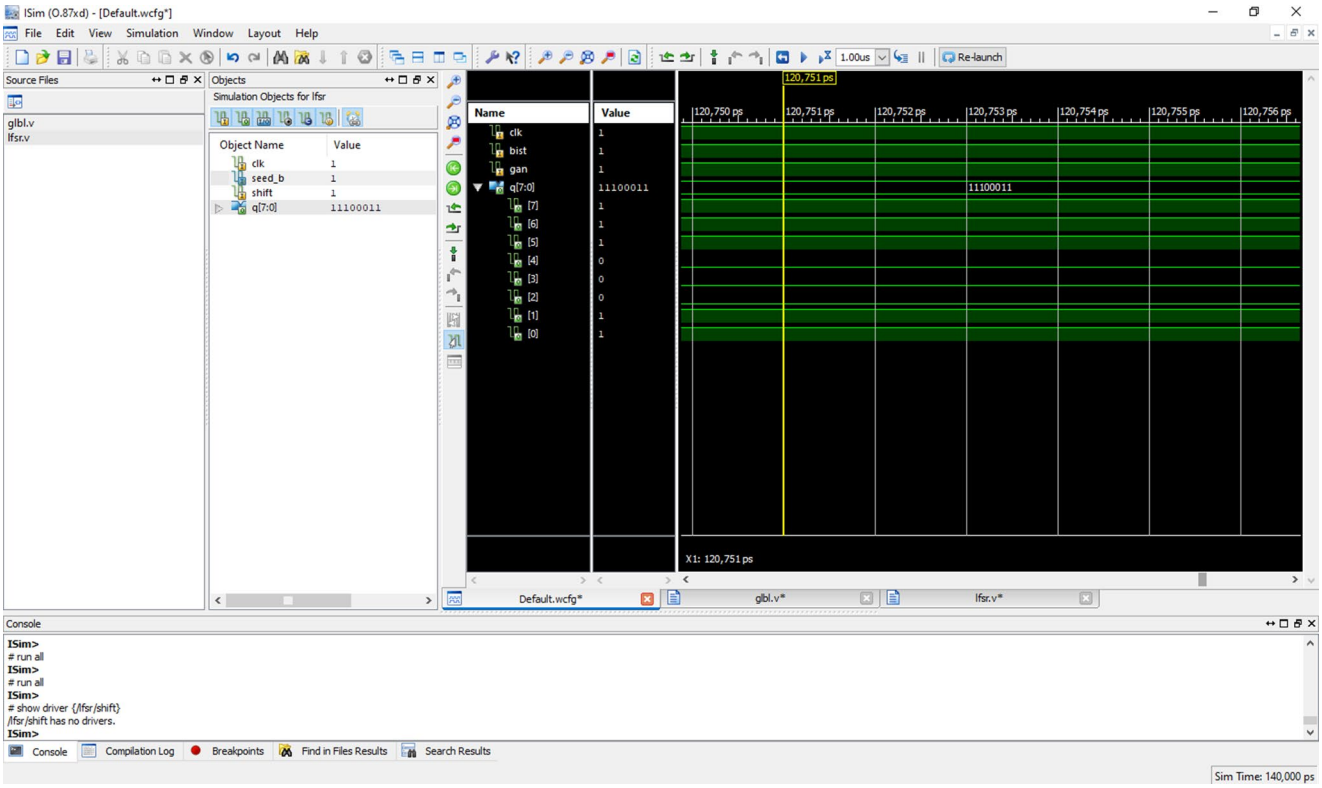

Fig. 12 Simulated output pattern

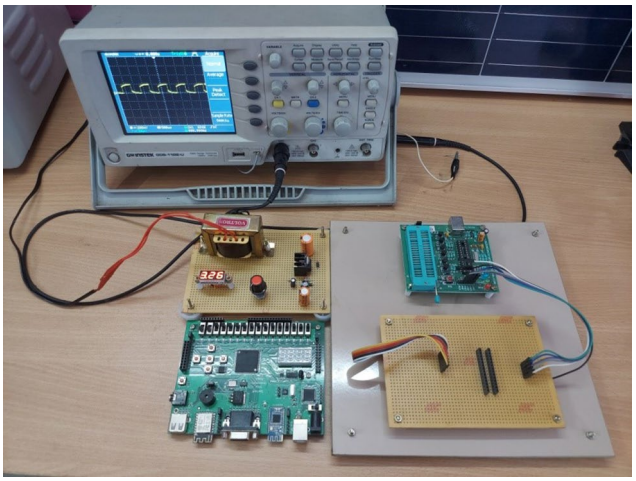

Fig. 13 Experimental setup

Table 1 Comparison of power consumption

Methods	Power Consumption (mW)	Delay (ns)
Decompressor MPC [10]	110.64	7.528
BS [26]	1114.30	0.904
MISC [27]	215	0.170
MPC [28]	129.75	-
Proposed	94.11	0.019

Table 2 Comparative analysis of fault coverage

Techniques	Fault Coverage (%)
[17]	80.95
[16]	91.63
[5]	96
Proposed	98.21

Table 3 Comparison of adder performance

Approaches	Width	Area (μm^2)	Delay (n)
CSLA [29]	8b	163.8	0.75
	16b	342	1.12
	32b	745.2	2.06
BEC-based CSLA [30]	8b	99.68	0.87
	16b	199.36	1.42
	32b	398.72	2.24
PGAN-TPPTA	8b	96.21	0.71
	16b	160.34	1.10
	32b	192.4	2.03

for applications where comprehensive fault coverage is critical.

Table 3 presents a comprehensive comparison of area considerations for different approaches to adder design. The CSLA [29] architecture is utilized with varying widths of 8, 16, and 32 bits. The corresponding areas are reported as $163.8\mu\, m^2$,$342\,\mu\, m^2$, and $745.2\,\mu\, m^2$, with delays of $0.75\,ns$, $1.12\,ns$, $and\, 2.06\,ns$, respectively. The second

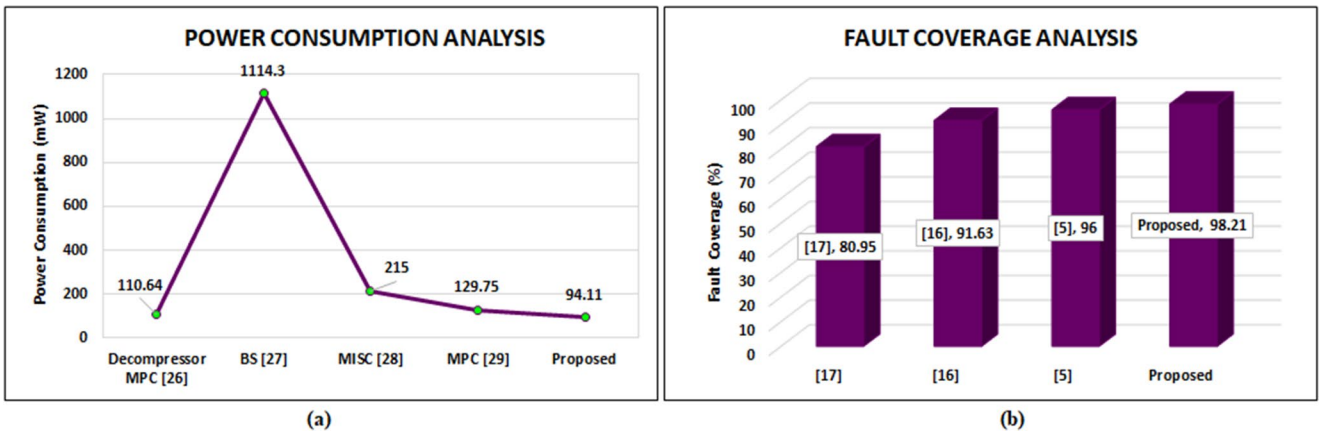

Fig. 14 Graphical representation of (**a**) Power consumption and (**b**) Fault coverage

approach introduces an improvement through BEC-based CSLA [30], demonstrating reduced areas of $99.68\,\mu\,m^2$,$199.36\,\mu\,m^2$, and $398.72\,\mu\,m^2$ for 8-bit, 16-bit, and 32-bit widths, with delays of $0.87\,ns,\ 1.42\,ns,\ and\ 2.24\,ns$, respectively. The proposed GAN-TPPTA approach stands out with competitive area considerations, showcasing areas of $96.21\,\mu\,m^2$,$160.34\,\mu\,m^2$, and $192.4\,\mu\,m^2$ for 8-bit, 16-bit, and 32-bit widths, coupled with delays of $0.71\,ns,\ 1.10\,ns,\ and\ 2.03\,ns$, respectively. The results underscore the efficiency of GAN-TPPTA approach in achieving comparable or even superior area performance, particularly for smaller bit widths, while maintaining competitive delay characteristics.

Figure 14(a), visually depicts the power consumption values for different BIST techniques, with the proposed optimized method showcasing the lowest power consumption with a value of$94.11mW$. Consequently, in Fig. 14(b), the graph illustrates fault coverage percentages, where the proposed technique excels with the highest fault coverage with98.21%. This figure neatly summarizes the comparative analysis, highlighting the balance between power efficiency and fault detection effectiveness, making the proposed GAN optimized approach as a favourable choice for achieving an optimal balance.

5 Conclusion

This paper proposes a transformative approach that melds the power of GANs with realm of BIST operations in VLSI circuits. This innovative integration addresses the intricate challenge of reconciling energy efficiency and fault coverage in the face of escalating circuit complexity. By infusing GANs into the BIST architecture, empowers the architecture to adapt and tailor test patterns to the unique attributes of individual circuits, enhancing fault detection capabilities. The TPPTA seamlessly collaborates with the GAN-generated patterns, ensuring efficient translation into circuit-compatible input signals while minimizing power consumption. GAN-TPPTA test vectors is applied to verify proper BIST operation, and results demonstrated the architecture's ability to efficiently identify and isolate faults, making it a promising solution for FPGA testing and maintenance. The analysis of proposed system results in reduced power consumption of $94.11mW$ with delay of$0.019ns$. Furthermore, the fault coverage capacity of proposed optimized system results in 98.21% respectively. The comprehensive evaluations undertaken in this work affirm the superiority of the GAN-enhanced BIST architecture over conventional methods. The successful integration of GANs optimized TPPTA underscores the potential for reshaping testing paradigms, presenting a realistic solution to the contemporary challenges posed by intricate circuit designs.

Funding The authors received no specific funding for this study.

Data Availability Not applicable.

Declarations

Conflict of Interest An authors have no conflicts of interest to declare that are relevant to the content of this article.

References

1. Zhang G, Yuan Y, Liang F, Wei S, Yang CF (2019) Low cost test pattern generation in scan-based BIST schemes. Electronics 8(3):314. https://doi.org/10.3390/electronics8030314
2. Panda AK, Ray KC (2018) Modified dual-CLCG method and its VLSI architecture for pseudorandom bit generation. IEEE Trans Circuits Syst I Regul Pap 66(3):989–1002. https://doi.org/10.1109/TCSI.2018.2876787
3. Chen P, Yin J, Zhang F, Mak PI, Martins RP, Staszewski, RB (2021) Mismatch analysis of DTCs with an improved BIST-TDC in 28-nm CMOS. IEEE Trans Circuits Syst I Regul Pap 69(1):196–206. https://doi.org/10.1109/TCSI.2021.3105451

4. Liang H, Xiangsheng FANG, Maoxiang YI, Huang Z, Yingchun LU (2018))a novel BIST scheme for circuit aging measurement of aerospace chips. Chin J Aeronaut 31(7):1594–1601. https://doi.org/10.1016/j.cja.2018.04.013
5. Shivakumar V, Senthilpari C, Yusoff Z (2021) A low-power and area-efficient design of a weighted pseudorandom test-pattern generator for a test-per-scan built-in self-test architecture. IEEE Access 9:29366–29379. https://doi.org/10.1109/ACCESS.2021.3059171
6. Menbari A, Jahanirad H (2022) A low-cost BIST design supporting offline and online tests. J Electron Test 38(1):107–123. https://doi.org/10.1007/s10836-022-05986-0
7. Harutyunyan G, Shoukourian S, Zorian Y (2018) Fault awareness for memory BIST architecture shaped by multidimensional prediction mechanism. IEEE Trans Comput Aided Des Integr Circuits Syst 38(3):562–575. https://doi.org/10.1109/TCAD.2018.2818688
8. Pavlidis A, Louërat MM, Faehn E, Kumar A, Stratigopoulos HG (2021) SymBIST: symmetry-based analog and mixed-signal built-in self-test for functional safety. IEEE Trans Circuits Syst I Regul Pap 68(6):2580–2593. https://doi.org/10.1109/TCSI.2021.3067180
9. Jamal K, Chari KM, Srihari P (2019) Test pattern generation using thermometer code counter in TPC technique for BIST implementation. Microprocess Microsyst 71:102890. https://doi.org/10.1016/j.micpro.2019.102890
10. Warade NS, Ravi T (2022) Design of Model Predictive Control Pseudo Random Pattern Generator for Low Power BIST. J Eng Sci Technol 17(1):0207–0224
11. Moghaddam E, Mukherjee N, Rajski J, Solecki J, Tyszer J, Zawada J (2018) Logic BIST with capture-per-clock hybrid test points. IEEE Trans Comput Aided Des Integr Circuits Syst 38(6):1028–1041. https://doi.org/10.1109/TCAD.2018.2834441
12. Sowmiya G, Malarvizhi S (2023) Memory-efficient LFSR encoding and weightage driven bit transition for improved fault coverage. IETE J Res 69(4):1783–1788. https://doi.org/10.1080/03772063.2021.1958072
13. Ying JC, Tseng WD, Tsai WJ (2018) Asymmetry dual-LFSR reseeding for low power BIST. Integration 60:272–276. https://doi.org/10.1016/j.vlsi.2017.10.012
14. Kaczmarek B, Mrugalski G, Mukherjee N, Pogiel A, Rajski J, Rybak Ł, Tyszer J (2021) LBIST for Automotive ICs with enhanced test generation. IEEE Trans Comput Aided Des Integr Circuits Syst 41(7):2290–2300. https://doi.org/10.1109/TCAD.2021.3100741
15. Martínez LH, Khursheed S, Reddy SM (2020) IET computers Digit techniques 14(1):27–36. https://doi.org/10.1049/iet-cdt.2019.0042. LFSR generation for high test coverage and low hardware overhead
16. Lee K, Lee S, Park J, Lee I, Kang S (2021) A low-power BIST scheme using weight-aware scan grouping and scheduling for automotive ICs. IEEE Access 9:116115–116132. https://doi.org/10.1109/ACCESS.2021.3105429
17. Naveen Balaji G, Chenthur Pandian S (2019) Design of test pattern generator (TPG) by an optimized low power design for testability (DFT) for scan BIST circuits using transmission gates. Cluster Comput 22:15231–15244. https://doi.org/10.1007/s10586-018-2552-x
18. Garbolino T (2021) A new, fast pseudo-random pattern generator for advanced logic built-in self-test structures. Appl Sci 11(20):9476. https://doi.org/10.3390/app11209476
19. Hasler J, Natarajan A, Kim S (2018) Enabling energy-efficient physical computing through analog abstraction and IP reuse. J Low Power Electron Appl 8(4):47. https://doi.org/10.3390/jlpea8040047
20. Cui X, Zhang M, Lin Q, Cui X, Pang A (2019) Design and test of the in-array build-in self-test scheme for the embedded RRAM array. IEEE J Electron Devices Soc 7:1007–1012. https://doi.org/10.1109/JEDS.2019.2931757
21. Mittal P, Shah D (2020) Linear Feedback Shift Register-based test pattern generators: a comparative study. Int J Adv Res Electron Communication Eng 9(8):60–62
22. Mishra AK, Das AK, Kandasamy N (2022) Built-In functional testing of Analog In-Memory accelerators for deep neural networks. Electronics 11(16):2592. https://doi.org/10.3390/electronics11162592
23. Ahmed MA, Alnatheer S (2022) Deep Q-Learning with bit-swapping-based Linear Feedback Shift Register fostered Built-In self-test and Built-In self-repair for SRAM. Micromachines 13(6):971. https://doi.org/10.3390/mi13060971
24. Maity H, Khatua K, Chattopadhyay S, Sengupta I, Patankar G, Bhattacharya P (2019) Fault coverage enhancement via weighted random pattern generation in bist using a dnn-driven-pso approach. In 2019 International Conference on Information Technology (ICIT), 228–233. IEEE. https://doi.org/10.1109/ICIT48102.2019.00047
25. Sadi M, Contreras GK, Chen J, Winemberg L (2017) Tehranipoor MDesign of reliable SoCs with BIST hardware and machine learning. IEEE Trans Very Large Scale Integr VLSI Syst 25(11):3237–3250. https://doi.org/10.1109/TVLSI.2017.2734685
26. Gupta K, Raman A (2022) Design of Low-Power Bit Swapping BIST for IC Self-testing. In Innovations in Electronics and Communication Engineering: Proceedings of the 9th ICIECE 2021 569–577. Singapore: Springer Singapore, https://doi.org/10.1007/978-981-16-8512-5_60
27. Kolanchinathan VP, Dinesh Kumar TR, Jaishree P, Niranjana M, Sowmiya M, Thresha V (2023) Low-power and space-efficient built in Self-Test Architecture with MSIC Test Pattern Generator. J Popul Ther Clin Pharmacol 9308–314. https://doi.org/10.47750/jptcp.2023.30.09.031
28. Warade N, Ravi T (2020) Reference signatures and selective test pattern generator based Model Predictive control technique for test response prediction in low power BIST. Int J 7. https://doi.org/10.30534/ijeter/2020/84872020
29. Nam M, Choi Y, Cho K (2018) High-speed and energy efficient carry select adder (CSLA) dominated by carry generation logic. Microelectron J 79:70–78. https://doi.org/10.1016/j.mejo.2018.07.001
30. You H, Yuan J, Tang W, Qiao S (2019) An energy and area efficient carry select adder with dual carry adder cell. Electronics 8(10):1129. https://doi.org/10.3390/electronics8101129

C. Thangam received her B.E. degree in Electronics and Communication Engineering and M.E. degree in Applied Electronics from Ms University, Tirunelveli and Anna University, Chennai, Tamilnadu, India in 2004 and 2013 respectively. Her areas of interest are MIMO systems, VLSI design and Digital Systems.

R. Manjith received his B.E. degree in Electronics and Communication Engineering and M.E. degree in Communication Systems from Madurai Kamaraj University and Anna University, Tamilnadu, India in 2002 and 2008 respectively. He has received his PhD in Wireless Communication under the faculty of Information and Communication Engineering from Anna University, Chennai, Tamil Nadu, India during November 2014. His areas of interest are OFDM & MIMO systems, VLSI design and Digital Image Processing.