International Journal of Advanced Technology and Engineering Exploration, Vol 12(123)
ISSN (Print): 2394-5443 ISSN (Online): 2394-7454
http://dx.doi.org/10.19101/IJATEE.2023.10102083

Research Article

Optimizing integrated circuit testing: a comprehensive approach to testability and efficiency

Praveen K[1*], Rajanna G S[1] and Shivakumara Swamy G M[2]
Srinivas University, Mukka, Surathkal, Mangaluru-574146, Karnataka, India[1]
Bapuji Institute of Engineering and Technology, Shamnur Road, Davangere-577004, Karnataka, India[2]

Received: 07-August-2023; Revised: 26-January-2025; Accepted: 11-February-2025

Abstract
Testing is a critical aspect of integrated circuit (IC) design, aimed at ensuring thorough evaluation of all nodes within the designs netlist to identify potential defects. The effectiveness of design for testability (DFT) relies on assessing the observability and controllability of each node within the architecture. These attributes form the foundation of DFT, enabling optimal test coverage (TC) with minimal test time (TT). The rigor of design testing is directly linked to the number of patterns used, with crucial parameters, such as multi-voltage conditions, temperature variations, and comprehensive testing methodologies playing key roles during the process. This methodology integrates techniques, such as scan insertion (SI), automatic test pattern generation (ATPG), embedded deterministic testing, and pattern simulation. The primary goal is to compress and minimize test data volume (TDV) and TT. ATPG plays a pivotal role in this approach by generating patterns tailored to meet compression requirements. Simulation validates the effectiveness of these patterns in reducing TT. The application of this methodology leads to significant improvements in testing efficiency. This study effectively determines the optimal multi-voltage and temperature parameters for IC testing. By leveraging automatic pattern generation and compression techniques, it achieves a substantial reduction in both test data and TT. The findings emphasize the importance of adhering to the DFT principles of observability and controllability to maximize TC. The proposed methodology demonstrates clear improvements in TDV and TT efficiency, contributing not only to the field of IC manufacturing but also emphasizing the value of a systematic and comprehensive approach to DFT in enhancing the reliability and performance of ICs.

Keywords
Design for testability (DFT), Test coverage (TC), Test time (TT), Test data volume (TDV), Automatic test pattern generation (ATPG), Observability and controllability.

1.Introduction
Scan insertion (SI) is a key technique in design for testability (DFT). It enhances the controllability and observability of integrated circuits (ICs) during testing [1, 2]. Controllability refers to the ability to provide input signals, while observability indicates the ability to monitor output based on the given input. Functional testing assesses the design functionality, whereas DFT detects defects or faults arising from design issues, manufacturing errors, or processing flaws of IC. Such defects may include improper silicon structures, missing molecules, and connecting pads.

The SI technique integrates flip-flops into the design by adding a multiplexer before each flip-flop to enhance controllability and observability. These flip-flops are connected sequentially to form a scan chain [1]. The scan chains are loaded serially, capture the combinational logic, and then unload the data. A dedicated scan clock is used for shifting, capturing, and unloading operations. The evaluation is conducted by comparing the actual input with simulated values. The scan chain length is determined based on edge mixing, clock mixing, and power domain considerations.

Manually writing input and output patterns for any design is a time-consuming process that requires handling large volumes of data. To address this, automatic test pattern generation (ATPG) was introduced [3]. This technology improves test efficiency, particularly in circuits with deep logic testing architectures. ATPG detects stuck-at faults and transition delay faults. Stuck-at faults testing identifies logic faults for 0 and 1 states, while

*Author for correspondence

transition delay fault testing focuses on timing errors. Fault detection uses the D-algorithm, which involves fault activation, fault propagation, and fault justification. Transition delay fault testing utilizes two vectors: initialization and finalization. Additionally, the launch-on-capture (LOC) ATPG method exhibits performance comparable to that of the launch-on-shift (LOS) method for at-speed or transition delay testing. ATPG is effective in generating comprehensive test patterns to identify various design faults.

Test point insertion is introduced to enhance test coverage (TC) by enabling control over specific nodes. If the desired coverage improvements are not achieved, test points are added to these nodes by applying constraints to the hierarchal structure of the design. This process offers an additional channel for a certain value to propagate, thereby enhancing the observability of logic signals.

As the size of designs increases, their complexity increases, leading to an exponential rise in test data volume (TDV) and escalating test costs. Embedded deterministic testing (EDT) is an emerging compression technology [4] designed to address challenges. It operates at both the register-transfer level (RTL) and synthesized design logic levels by incorporating decompressor and compactor logic. By applying EDT [5], the impact of expanding TDV on test costs is significantly reduced.

The background highlights the significance of SI in DFT, the effectiveness of reconvergence-aware testability measures, the comparison between LOC and LOS ATPG methods [6], the role of control points in enhancing TC, the structure and importance of scan chains, and the application of EDT in managing TDV [7]. The subsequent sections explore the details of each of these components. Experiments indicate its effectiveness when combined with different algorithms, paving the way for exploration with more advanced algorithms [8]. Additionally, the LOC ATPG method has shown performance comparable to the LOS method for at-speed testing, with the potential to replace LOS in various designs. ATPG's strength lies in its ability to detect a wide range of faults and generate comprehensive test patterns to identify all possible design faults. Furthermore, the application of EDT [9] effectively mitigates the impact of expanding TDV on test costs.

DFT technology plays a vital role in the testability features of the IC design. The process is conducted systematically, starting with the addition of testable features to the design, known as the SI process. During SI, any violations found need to be addressed by adding test point insertions. The SI-synthesized netlist also referred to as the gate-level netlist, is used to generate test patterns. EDT can be added to the process to reduce the TDV and test time (TT). The tool creates the test bench and test patterns to identify the design's faults, errors, and failures. ATPG patterns [10, 11] are validated, and faulty reports are generated for the design. The design is tested for stuck-at and at-speed patterns across different frequencies. The simulation tool validates the generated patterns against the timing fixtures in the design.

Challenges in digital design testing include balancing scan chains and incorporating DFT features, such as memory tests or built-in self-test (BIST) features, as these may impact chip area and performance [12]. The design validates the generated pattern with a low-cost configuration of the design [13]. A key consideration is managing asynchronous clocks and resets during the testing process. It is essential to balance testability requirements while minimizing area and ensuring high-speed performance operations. Additionally, specific DFT methodologies, such as scan chains and joint test action groups (JTAG), can elevate power consumption during testing, facilitating careful management to adhere to power budgets for battery-operated devices [14]. Embedding DFT functionalities substantially increases design complexity, leading to extended design cycles, greater testing efforts, and a higher likelihood of design errors, particularly in large designs where testing vector data is generated, requiring robust data management strategies [15]. This leads to increased costs and decreased throughput, highlighting the importance of test compression. Hierarchical testability challenges arise due to design hierarchies, complicating test pattern generation, fault localization insertion, and planning for comprehensive coverage in hierarchical test strategies. Additionally, integration challenges with electronic design automation (EDA) tools may occur when DFT tools are integrated with those from other EDA vendors.

The benefit of testing IC is to improve the quality of IC production and fault detection in semiconductor devices. This enables the early and accurate identification of faults during fabrication, enhancing overall product quality and reducing manufacturing

costs [16]. Enhanced testability and streamlined test pattern generation facilitate quicker identification and resolution of issues, resulting in increased production yield and faster time-to-market. Optimized use of test resources maximizes effectiveness while managing power consumption, ensuring adherence to power budgets. Additionally, system-level testing facilitation and compliance with industry standards promote compatibility and reliability.

The objective of this paper are as under:

1. To insert SI into the provided input design with a maximum chain length (MCL) of 72 and a chain count of 10.
2. To achieve a compression ratio (CR) of 4.19x and analyze the design performance before and after compression logic insertion.
3. To generate a statistical report for stuck-at and transition faults for a design rule check (DRC)-compliant design, along with the corresponding test patterns, and produce LOC patterns for fault modes, including scan and chain patterns for fault model and transition fault simulation.

The remaining part of the manuscript is organized as follows: Section 2 explains the research conducted on IC testing and fabrication testing technology. Section 3 discusses the methodologies involved in the design testing process, supported by a flowchart. Section 4 presents the results obtained from the tool concerning compression. Section 5 explains the limits of IC design process technology and its design implications, while Section 6 addresses the limitations and future scope.

2.Literature review

Related work has been discussed in this section to explore the advantages and disadvantages. Ramasamy and Samiappan [17] described testing as a vital step in the product delivery process, providing essential data for identifying and rectifying faults, which ultimately contributes to design and manufacturing improvements and better returns. Delay, power consumption, and enhancement of faulty coverage are key factors considered in these methods. The effectiveness of the proposed linear feedback shift register (LFSR) architecture is demonstrated in achieving improvements in fault coverage and power consumption. The implications of this approach extend to the broader field of very large-scale integration (VLSI) testing, providing a novel solution to address inherent challenges. Ye et al. [18] addressed the critical issue of determining the optimal input and output channel counts for test compressing hardware, focusing on minimizing TDV. The study presented efficient methods for estimating pattern count (PC) across a diverse set of input or output configurations, emphasizing the importance of on-chip test compression hardware in alleviating test costs. The efficient methods developed for estimating TT count across diverse configurations are underscored as pivotal contributions. The practical implications of these estimation results are significant, both in optimizing on-chip hardware and in guiding configurations in system on chip (SOC) systems. Tsai et al. [19] presented two approaches for predicting test patterns and volumes while considering different types of input channels. These approaches offered valuable insights for optimizing scan test configurations during the DFT planning phase. Furthermore, the error data rates for the smallest expected data volumes were reported to be less than 3.5%, demonstrating the accuracy and reliability of these prediction methods. Rapid Single Flux Quantum (RSFQ) logic [20], which leverages Josephson Junctions (JJ), is emerging as a high-performance alternative to Complementary Metal-Oxide-Semiconductor (CMOS) technology, where physical scaling has reached its limits. RSFQ offers unique advantages, including a lower defect density compared to CMOS due to its larger feature sizes. However, process variations and RSFQ-specific nonidealities present significant challenges, making timing verification and delay testing essential for ensuring chip performance. RISC circuits have introduced delays and tools, along with a new ATPG paradigm [21], demonstrating promising results in addressing RSFQ-specific challenges. The findings also suggest potential future directions for further refinement and application of these methodologies in the rapidly evolving landscape of RSFQ technology. Kung et al. [22] presented a methodology for generating test patterns for various stuck-at faults, categorizing them as either 0 or 1. Transition faults, including slow-to-rise (STR) and slow-to-fall (STF), were also addressed, along with path delay faults, which account for delays between nodes. Additionally, static bridging faults—occurring within node faults—and transistor stuck defect faults were considered. Test pattern generation was performed for all fault types, with the ability to impose different constraints to control detectable faults. The reconvergence-aware testability measure was designed to overcome the limitations of existing ATPG methodologies. This approach aimed to enhance the justification process by specifically minimizing backtracks caused by reconvergence

[23]. Its effectiveness was validated through comprehensive experiments, demonstrating a significant reduction in ATPG execution time. Notably, circuits with higher design logic levels achieved a 76% improvement, marking a substantial advancement in ATPG efficiency and performance. The primary objective of scan-cell design was to reduce power consumption during testing [24]. The focus was on minimizing switching power dissipation in both scan chains and logical ATPG when the chip was in test mode. The review emphasized the significance of this objective in enhancing the overall efficiency and power profile of the IC. The scan-based at-speed test aimed to overcome the limitations of existing methods by improving controllability and enhancing result quality. The benefits of this approach, such as its flexibility in transition launching and the introduction of local scan enable generator (LSEG), were discussed as key advantages in achieving more effective and reliable testing. The role of EDA methods in TT for semiconductor devices was emphasized. The scan-based at-speed test was presented as a promising solution to address challenges in the transition fault model, providing flexibility and improved controllability [25]. Kongala et al. [26] highlighted a significant bottleneck encountered in the verification process of multi-billion gate system-on-chips: the time required to verify various operation modes through simulation. Through extensive analysis of numerous test benches, they observed that certain tasks, such as compiling and elaborating the design under test, simulating power-on or initialization sequences, and similar processes, were repetitive and time-consuming. A novel hybrid approach [27] was designed to address the challenges of test case-based selection in regression testing. This approach combined a modified greedy technique for test case selection with genetic algorithm methods. The modified greedy approach handled the test case selection process, while the genetic algorithm optimized the test suite using key parameters, such as initial population, test case combinations, fitness value, and the processes of test case mutation and crossover. A low-power random access memory (RAM) was developed based on a modified gate diffusion input and a test pattern generator (TPG). By leveraging techniques such as the LFSR and novel RAM architecture designs, the aim was to reduce energy consumption during testing, a common challenge in VLSI systems [28]. Additionally, the implementation of a low-transition TPG, such as the LFSR with a low-density modified low-transition low-power design, further minimized power dissipation by optimizing the switching activity between test vectors. The need for a low-cost pattern generator in scan-based BIST schemes was highlighted. The objective was to enhance the efficiency of test pattern generation by utilizing vectors, specifically targeting the reduction of hardware overhead [29]. The proposed broadcast-based multiple single input change-test pattern generator (BMSIC-TPG) algorithm successfully achieved a notable reduction in hardware overhead while maintaining lower power consumption and substantial fault coverage. The broadcasting approach further enhanced the efficiency of test pattern generation in scan-based BIST schemes. Srivastava and Abraham [30] addressed power-induced over-testing in ICs, where fully functional chips failed due to excessive IR drop or power supply noise during scan-based at-speed tests. Traditional solutions involved ATPG improvements or low-power DFT techniques with hardware modifications. The paper proposed a novel partition-based adaptive test method that applied local hotspot power constraints to ATPG, reducing over-testing and minimizing pattern count. Experimental results revealed improved reliability and efficiency in transition delay fault testing. Ni et al. [31] discussed the rising concerns about testability and reliability in secure hardware, which were exacerbated by global manufacturing practices that increased the risk of tampering. Hardware security extended beyond individual components to encompass entire systems, requiring a full-system approach. By focusing on testability, reliability, and security across all design levels, improved system performance was achieved. As new technologies continued to emerge, balancing these factors became crucial for both researchers and industry. Pomeranz [32] developed a novel method for selecting paths to test path delay faults. Traditional methods often focused on long paths, which were frequently untestable. Their approach, called “negative path selection,” excluded certain paths to prioritize those that were testable. The method efficiently eliminated shorter paths and leveraged fault information to improve the process. Testing on circuits demonstrated its effectiveness in identifying long-path delay faults. Li et al. [33] explored RSFQ logic, which used Josephson Junctions (JJ) as a high-performance alternative to CMOS technology limitations. While RSFQ featured a lower defect density, it faced challenges related to process variations and specific non-idealities. Its pulse-based operation underscored the importance of timing verification and delay testing. The paper introduced single-pattern delay tests and proposed a novel ATPG method designed to address worst-case

delays by evaluating multiple pipeline stages. Monte Carlo simulations confirmed the effectiveness of this approach in mitigating the impact of process variations.

3.Methodology

DFT is an integrated feature used to test the netlist or RTL design. *Figure 1* presents the flowchart of the DFT methodology. The methodology involves the SI processes, where EDT can be inserted either before or after SI. It involves performing LOC and LOS operations during the capture procedure, generating stuck-at and transition patterns, and saving the patterns in formats such as Verilog, standard test interface language (STIL), waveform generating language (WGL), and American Standard Code for Information Interchange (ASCII), suitable for simulation purposes.

DFT is a critical step in digital design, built upon RTL design to ensure that digital circuits are testable and faults can be efficiently diagnosed. During the manufacturing phase, designers must incorporate DFT techniques to facilitate testing. These techniques include inserting scan chains, implementing memory BIST structures, and adding other testability features.

For example, scan chains enable the serial shifting of TT into and out of the circuit, simplifying the testing process. Designers must ensure that RTL code includes features that enhance testability, such as the ability to control and observe internal signals during testing. Additionally, DFT must be compatible with the clock domains in the design. The insertion of scan chains must be carefully synchronized with the circuit's clocking strategy to ensure proper functionality.

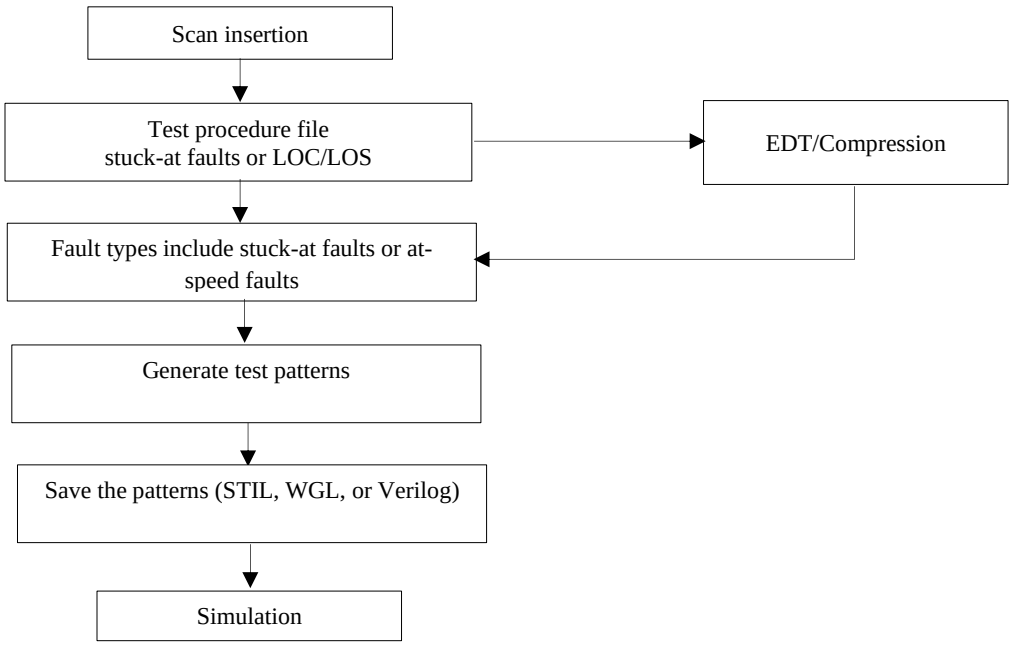


Figure 1 DFT methodology

3.1Libraries in DFT

Standard cells libraries

DFT features, such as scan cells, are often integrated into standard cell libraries provided by semiconductor foundries. These libraries contain specialized cells designed specifically for scan testing, enabling the efficient implementation of DFT features in the design.

Scan cells

DFT libraries often include specialized scan cells optimized for scan chain insertion. These cells have additional control inputs and outputs to support scan functionality.

BIST structures

Libraries may also include cells and structures optimized for BIST implementation. These structures typically include TPGs, response analyzers, and control logic.

3.2SI process

SI is a critical step in the DFT process, aimed at enhancing the testability of digital ICs. It involves

inserting scan chains into the design to simplify circuit testing during manufacturing. The primary goal of SI is to facilitate the application of TT and the observation of responses during the testing phase. A scan chain comprises a series of flip-flops connected in sequence, allowing the serial shifting of TT into and out of the circuit. Each flip-flop in the scan chain has a dedicated scan input and output. During SI, normal flip-flops are replaced with scan flip-flops by adding multiplexers to the flip-flops. These scan flip-flops have additional inputs and outputs for scan data, enabling the serial loading and unloading of TT [34]. Scan flip-flops operate in three modes:

- Shift Mode
- Capture Mode
- Unload mode

Figure 2 reveals how normal D flip-flops are converted into scanned flip-flops by adding a multiplexer to the D-pin. The multiplexer includes a functional pin, a scan input, and a scan enable pin.

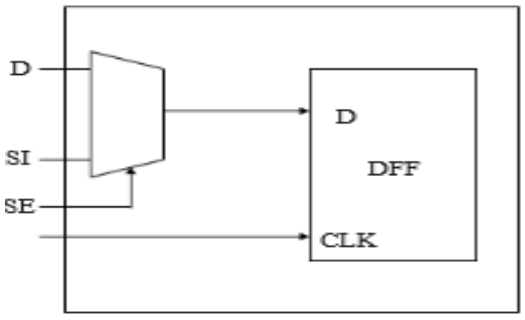


Figure 2 Normal flip-flop is converted into the scanned flop

When SE = 0, the functional input, shift data, and unloading of the data occur from the D-pin. When SE = 1, the input is from the scan-in pin, and the capture process occurs, as demonstrated in *Figure 3*. Fault models can be tested in different methods, including stuck-at and transition-delay faults. Stuck-at faults are detected when the structure of the design is connected directly to logic 0 and logic 1, whereas transition faults are detected for small delays in switching from logic 1 to logic 0.

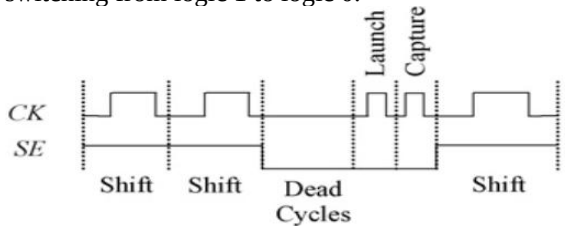


Figure 3 Procedures for shift and capture for LOC

3.3Test procedures

Figure 3 presents LOC, a testing methodology within the DFT framework that plays a vital role in identifying and diagnosing faults in digital ICs. Unlike traditional scan-based testing, LOC focuses on launching the TT directly into the functional portions of the circuit, avoiding the need for dedicated scan chains. This approach offers advantages, including reduced area overhead and potentially faster TT application. LOC leverages the inherent logic within the design to initiate the desired transitions, making it particularly useful for identifying at-speed faults and detecting subtle timing-related defects. By dynamically controlling the launch and capture of TT within the functional logic, LOC provides powerful and flexible methods for ensuring the reliability and quality of digital circuits during the testing phase of the semiconductor manufacturing process.

LOS is a test methodology used within the DFT framework, specifically for at-speed testing of digital ICs. It operates by launching TT during the shifting phase of the scan operation, utilizing scan chains to direct test vectors to the functional parts of the circuit, as shown in *Figure 4.*

This technique addresses the challenges associated with transition and path delay faults. LOS offers benefits in fault detection, especially in scenarios where traditional scan-based methods may have limitations. By carefully managing the launch timing, LOS provides a comprehensive approach to identifying and diagnosing timing-related issues. As an integral part of DFT, LOS ensures the robustness and reliability of digital circuits during the testing and manufacturing phases.

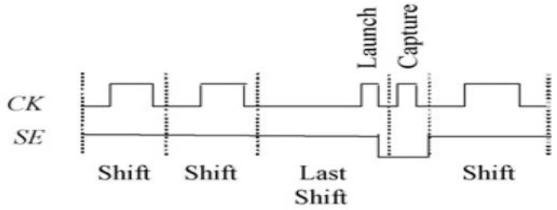


Figure 4 Procedures for shift and capture for LOS

3.4EDT architecture

EDT architecture is a design approach in electronics testing technology. This methodology integrates deterministic test structures directly into the digital circuitry of an IC. Unlike traditional external testing methods, EDT incorporates testing mechanisms within the chip, enhancing its ability to self-diagnose

and identify faults. This architecture typically incorporates on-chip instruments, such as BIST modules and scan chains, enabling the efficient application of deterministic TT. The key advantage of EDT is its ability to perform testing at speed, addressing potential faults related to circuit timing. This approach is particularly beneficial in high-performance and complex ICs, ensuring a thorough and reliable testing process without heavily relying on external test equipment. The EDT architecture plays a crucial role in achieving high fault coverage and ensuring the overall reliability and quality of digital designs during the manufacturing and operational phases [35]. EDT architecture comprises a decompressor, internal design, compactor, external channels, internal channels, and ATE, as shown in *Figure 5*.

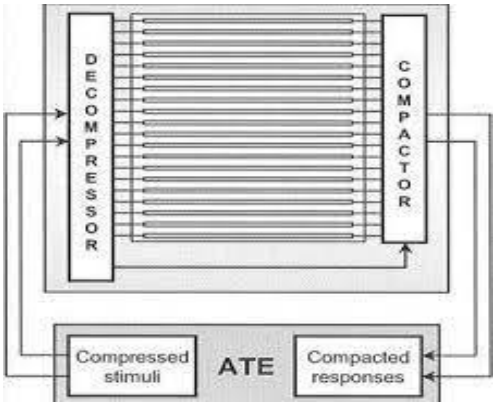


Figure 5 EDT architecture

The LFSR operates as a critical element within the decompressor of an EDT compression system, facilitating both compression and decompression processes during semiconductor testing. Functioning as a shift register, the LFSR comprises interconnected flip-flops with a feedback mechanism, generating a pseudo-random sequence of bits as test data is shifted [36]. This pseudo-random sequence is XORed with incoming test data during compression, yielding a compressed output. In the decompaction phase, the same LFSR configuration is utilized to reverse the compression, XORing the compressed data with the generated pseudo-random sequence to reconstruct the original test data [37].

The LFSR-generated sequence serves as a signature for error detection and correction, ensuring data integrity. Overall, the LFSR's role in the decompressor significantly enhances the efficiency of the compression and decompression processes, reducing test data while maintaining necessary fault coverage and optimizing semiconductor testing resources.

3.5Fault detection

Fault detection is a critical aspect of testing designs in electronic design. DFT techniques enhance the testability of ICs, making it easier to detect and diagnose faults during the manufacturing testing phase. Below are some common fault detection mechanisms used in DFT.

BIST: BIST is a DFT technique in which dedicated hardware is integrated into the design to perform self-testing. BIST modules generate TT, apply it to the circuit, and analyze the responses to detect faults.

Scan chains: Scan chains are sequences of flip-flops connected in sequence. TT is serially shifted into these chains during testing, enabling easy observation of internal nodes and the identification of faults.

Boundary scan or JTAG: The JTAG standard defines a boundary scan architecture that provides a standardized method for testing interconnects and detecting faults at the boundary of an IC.

Functional testing: In addition to specialized DFT techniques, functional testing is crucial. This involves applying input stimuli to the circuit and observing the output behavior. Any deviation from the expected behavior may indicate a fault.

Transition fault testing: This technique focuses on detecting faults related to the timing of signal transitions. It ensures that signals switch at the correct time and do not violate specified timing constraints.

Path delay testing: Path delay faults can impact the timing of critical paths in a design. Path delay testing aims to identify faults related to delays along specific paths within the circuit.

Stuck-at fault testing: This classic model simulates a fault where a signal is permanently stuck at logic 0 or 1. Stuck-at-fault testing is efficient and widely used to identify manufacturing defects.

Transition delay fault testing: This technique identifies faults related to signal transitions and delays, making it crucial for detecting timing issues that can affect the overall performance of the circuit.

3.6ATPG

ATPG is a technique used in digital design testing to generate test patterns that help detect faults or defects in a circuit. The goal is to ensure that manufactured ICs or digital systems meet their specifications and are free from defects. Below is a basic overview of how ATPG works:

Fault modeling: The first step in ATPG is to model potential faults that may occur in the digital circuit.

Common fault models include stuck-at faults (where a signal is stuck at 0 or 1), transition faults, and bridging faults (where two signals are shorted together).

Test generation: Once the fault models are defined, the ATPG tool generates test patterns to detect these faults. The tool aims to create a set of test vectors that, when applied to the circuit, activate or propagate the faults to observable points in the circuit.

Fault simulation: The generated TT is then simulated using a fault simulator. This process involves applying the TT to a version of the circuit that includes the modeled faults. The simulator evaluates whether the faults can be detected in the test patterns.

Test application: The final step involves applying the generated test patterns to the actual hardware (or its physical representation) for testing. The results are then analyzed to identify any detected faults.

3.7Pattern validation

ATPG involves creating and simulating test patterns to systematically test digital circuits. These patterns are specifically designed to detect faults, such as stuck-at faults or bridging faults. Once generated, the patterns are simulated using specialized tools. During simulation, the patterns are applied to a version of the circuit that includes modeled faults, and the circuit's behavior is analyzed. This process ensures that the patterns effectively activate or propagate faults to observable points in the circuit. It is crucial to validate the reliability and fault coverage of the patterns before they are used on the actual hardware. By leveraging simulations, ATPG enables thorough testing and verification of digital circuits, helping ensure the production of high-quality, defect-free ICs.

4.Results and discussion

4.1SI process

According to the first objective, SI should be inserted into the provided input design with an MCL of 72 or a chain count of 10. The DFT design process starts with the RTL (synthesized netlist). MCL is defined as the total number of chains present in a single chain.

The synthesized netlist can be generated using the design compiler(DC) tool. In this phase, a basic RTL design incorporating 718 flip-flops was selected, and the corresponding libraries were applied to all parameters used in the scan-based design, including SI, clocks, and necessary files such as the do-file and test procedure file, which must be available. During the SI process, all flip-flops are converted into scan flip-flops using the insert_test_logic switch or command.

A synthesized netlist in the context of DFT refers to a Verilog netlist that has been modified and enhanced to support testing requirements. Key DFT techniques include the insertion of scan chains, the addition of test points, and the incorporation of other features to facilitate the application of TT and the extraction of test results [38]. This design contains 718 flip-flops, with a chain length of 72 or a chain count of 10. As demonstrated in *Figure 6*, the chain count is 72 for chains 0 to 7, and the length of chains 8 and 9 is reduced by 1 count, resulting in a total of 718. These chains are created using the insert_test_logic command. *Figure 7* shows the scan chain configurations after resolving all the DRC violations, including details about the sequential library cells used to configure the scan chain with a length of 72. By default, the tool generates the test procedural file, do-file, and scan-inserted netlist.

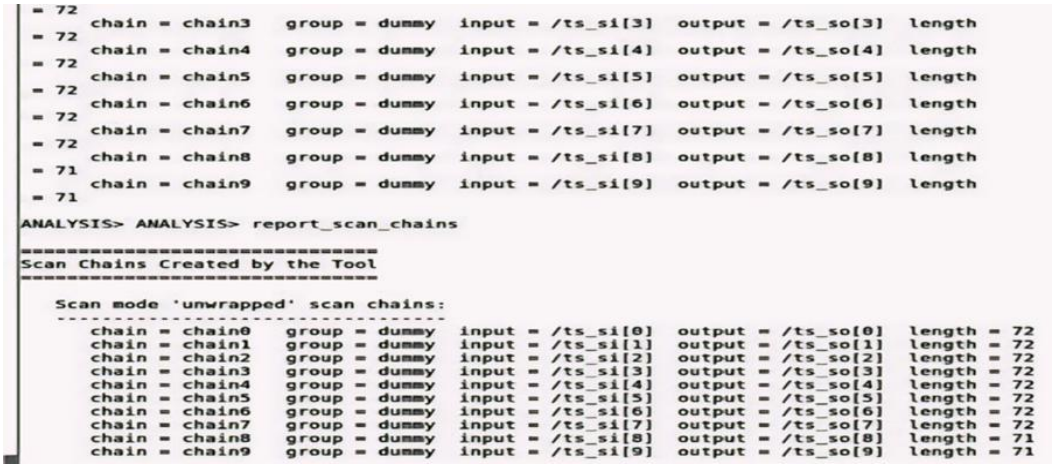


Figure 6 The creation of chain length count is 72

Figure 7 Scan chain configurations

4.2EDT configurations

The second objective is to achieve a CR of 4.19x. This can be achieved by various factors, such as the number of shift cycles, external scan channels, internal scan chain, decompressor size, and total number of scan cells. *Figure 8* provides complete details of the EDT configurations, including the number of internal and external channels, CR, number of shift cycles, number of X-masking bits, scan cells, and other reports.

Figure 9 presents the pin report for the EDT, showing 10 scan channels configured for both input and output.

EDT clock: The EDT logic comprises both combinational logic and flip-flops, functioning on a clock separate from the scan clock. It is crucial to avoid sharing this clock with any other clock source or the RAM control pin. Sharing the clock with scan cells could potentially cause disruptions when the edt_clk is pulsed during the load_unload process [39]. Strict adherence to the constraint of not sharing the clock is essential to avoid a decrease in TC. This is particularly evident during the capture phase, where the edt_clk must remain in the off-state. Moreover, the feasibility of implementing RAM sequential patterns and multiple load patterns may pose challenges under these specific circumstances.

EDT update: The pin edt_update serves the purpose of resetting or updating two distinct sets of registers within the decompressor and compactor logics. In the decompressor, the ring generator registers are affected, while in the compactor, the mask hold registers are modified. During the loading procedure, activating edt_update results in specific actions: the ring generator registers are reset to 0, and the mask_hold_register is loaded with the value from the mask_shift_register. This process is crucial because the ring generator requires a predictable state before generating random values, and setting all bits to 0 achieves this predictability. Meanwhile, the mask_hold_register is tailored for per-pattern masking, where each pattern has a specific masking sequence.

EDT bypass: It could be related to techniques or methods in EDT for bypassing certain functionalities during testing. However, without more context, it is challenging to provide specific information. A comparative analysis of the EDT before and after compression. The CR can be conducted. The CR can be computed based on the TDV and TT. Initial parameters include 718 flip-flops, 10 primary inputs, a frequency of 5 MHz, a PC set to 10, and a CR of 4.19x.

Pre-compression:

Pre and post compression is shown through Equation 1 to 7.

MCL = Total flops/Number of primary input
= 718/10=71.8 (1)
==72(It should not be a fraction)

TP= 1/F (2)
$=20\times10^{-6}$ sec

TT= (PC + 1)× MCL×TP (3)
=15.84 ms

TDV= Number_of_Chains×MCL×(PC×2) (4)
=14400 bits

Post compression:
CR=4.19x, Internal Chains =?, External Channels=10, (5)
CR=Internal chain/External Channels
4.19x=Internal chain /10
Internal Channel= 41.9==42(Roundoff)

MCL=718/42 =18
TT= (PC+1)× MCL×TP (6)
= 11×18×20×10^{-6} = 3.9 ms

TDV = Number of Chains×MCL×PC×2 (7)
= 10×18×10×2= 3600 bits.

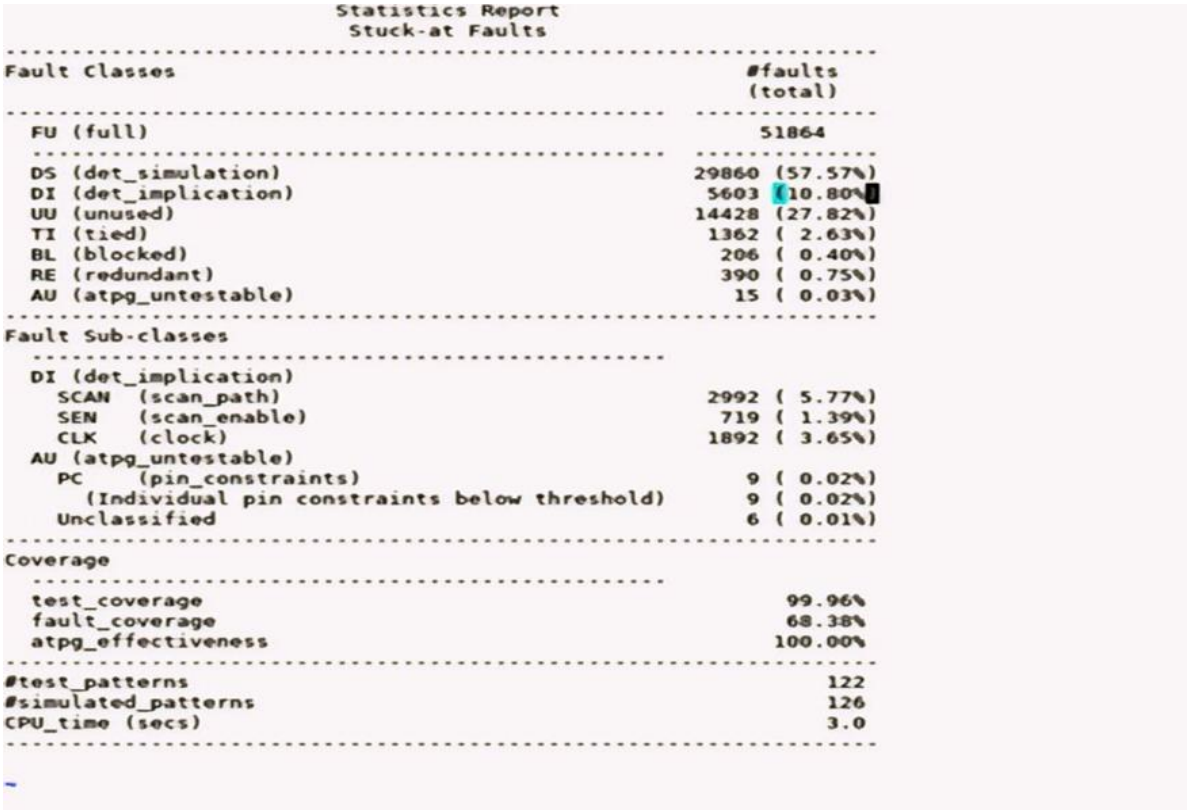
Simply compare the pre-compression and post-compression outcomes: TT reduced from 15.84 ms to 3.9 ms, and TDV decreased from 14,400 bits to 3,600 bits. The third objective is to provide a statistical report for stuck-at-pattern generation. *Figure 10* displays a statistical report of fault classes, showcasing the detected faults presented on the screen in the aforementioned format. Stuck-at faults, common issues in digital design, occur when a node or wire remains stuck at a specific logic value (0 or 1). Identifying these faults involves leveraging scan chains to generate TT, facilitating their detection and display during the testing process. TC is the ratio of the total number of detected faults to the total number of detectable faults, achieving 99.96%. TC should always be higher than fault coverage to ensure optimal test efficiency. Fault coverage is defined as the number of detected faults relative to the total number of possible faults. Fault coverage is the metric that gauges the tool's effectiveness in identifying a specific category of faults.

```
//
//  compactor
//    Setting: edt_compactor_i
//    Path   : NOT FOUND
//
ANALYSIS> report_edt_configurations
//
//  IP version:                 8
//  Shift cycles:               86,  72 (internal scan length) + 14 (additional cycles)
//  External scan channels:     2
//  Internal scan chains:       10
//  Masking bits:               11
//  Decompressor size:          16
//  Compactor type:             Xpress
//  Scan cells:                 718
//  Compression per pattern:    4.19x (ATPG bypass = 2 x 360, EDT = 2 x 86)
//  Bypass logic:               On
//  Lockup cells:               On
//  Clocking:                   edge-sensitive
//  Compactor pipelining:       Off
```

Figure 8 EDT configurations

```
satyanarayanaccdft4@GLS:/workarea/crashcourse/ccdft4/satyanarayanaccdft4/Project/compression
ANALYSIS> report_edt_pins
//
//  Pin description       Pin name            Inversion  Change edge  Channel pipeline stages (mask register
)
//  --------------        --------            ---------  -----------  -------------------------------------
-
//  Clock                 'edt_clock'             -           -
//  Update                'edt_update'            -           -
//  Bypass mode           'edt_bypass'            -           -
//  Scan channel 1 input  'edt_channels_in1'      -           -                          6
//   "      "    " output  'edt_channels_out1'     -          TE
//  Scan channel 2 input  'edt_channels_in2'      -           -                          5
//   "      "    " output  'edt_channels_out2'     -          TE
//
```

Figure 9 EDT pins report



```
                        Statistics Report
                         Stuck-at Faults
-------------------------------------------------------------------
Fault Classes                                          #faults
                                                       (total)
-------------------------------------------------------  ----------
  FU (full)                                             51864
  -----------------------------------------------------  ----------
  DS (det_simulation)                                  29860 (57.57%)
  DI (det_implication)                                  5603 (10.80%
  UU (unused)                                          14428 (27.82%)
  TI (tied)                                             1362 ( 2.63%)
  BL (blocked)                                           206 ( 0.40%)
  RE (redundant)                                         390 ( 0.75%)
  AU (atpg_untestable)                                    15 ( 0.03%)
-------------------------------------------------------------------
Fault Sub-classes
  -----------------------------------------------
  DI (det_implication)
    SCAN   (scan_path)                                 2992 ( 5.77%)
    SEN    (scan_enable)                                719 ( 1.39%)
    CLK    (clock)                                     1892 ( 3.65%)
  AU (atpg_untestable)
    PC     (pin_constraints)                              9 ( 0.02%)
      (Individual pin constraints below threshold)        9 ( 0.02%)
    Unclassified                                           6 ( 0.01%)
-------------------------------------------------------------------
Coverage
  -----------------------------------------------
  test_coverage                                        99.96%
  fault_coverage                                       68.38%
  atpg_effectiveness                                  100.00%
-------------------------------------------------------------------
#test_patterns                                           122
#simulated_patterns                                      126
CPU_time (secs)                                          3.0
-------------------------------------------------------------------
~
```

Figure 10 Statistical report for stuck-at fault

4.3ATPG and fault detections

In this scenario, the majority of faults are encompassed by the detectable simulation faults. To enhance coverage, it is possible to augment it by adjusting the sequential depth, which can be varied between 2 and 5, and by altering the abort limit, which can range from 100 to 5000.

Figure 11 reports on transition delay faults, which manifest in two formats: STF and STR. STR refers to the transition delay from logic 0 to 1, representing the time taken for the transition for the logic. Conversely, STF pertains to the delay from logic 1 to 0. Transition faults are a distinct category in digital circuits. These faults occur when the logic circuit fails to transition between logic levels as expected, influenced by factors, such as defects, delay, stuck-at faults, or other issues impacting timing and signal integrity.

The third objective is to represent the architectural diagram also or tree view of the EDT configurations of the design [40]. The primary purpose of a compactor is to reduce the amount of data that needs to be tested or transmitted after a test is applied to a digital circuit. The response data generated may be quite extensive, especially in the case of scan-based testing, where multiple scan cells capture the logic state of the circuit. Compactors help manage this data by compressing the response information without losing critical fault detection information.

A decompressor is a circuit or tool designed to reverse the compression process applied to response data during testing. Its primary function is to restore the compressed response data to its original form, enabling a detailed analysis of the test results. The Tessent tool creates the channels internally between the decompaction and compressor, depending on the required CR for the design. External output channels are determined by the channels provided by the designer.

Transition faults are another significant category of issues in digital circuits, particularly in DFT methodologies. These faults occur when the circuit fails to transition as expected between logic levels 1 and 0. Multiple factors, including faults, defects, or other issues impacting the circuit's timing and signal integrity, can contribute to transmission faults. Addressing these factors in DFT strategies is crucial

to ensuring robust testing and the reliability of digital circuits.

Patterns are created for stuck-at, transition, and path delay scenarios, designed for parallel and serial applications suitable for chain and scan capture. The simulation of these patterns depends on the chains, with each chain contributing information to the chain report. During this simulation, all chains are merged into a single chain to validate connectivity. ATPG produces patterns in various formats, such as ASCII, WGL, STIL, Verilog, System-Verilog, binary, and more. Designers often favor the STIL format due to its compatibility with different formats. The simulated results are obtained using patterns generated through the ATPG tool, a widely used methodology for crafting TT vital in uncovering and diagnosing faults within digital circuits. The ATPG algorithms analyze the circuit design to identify potential faults and produce TT capable of detecting these faults.

Issues arise in the verification of TT when there is a discrepancy between the results produced by the ATPG tool and the simulated outcomes, especially when considering timing information. This discrepancy is commonly known as simulated mismatch. After inserting the test logic into the design, fault detections are performed using the command "create_patterns," Once the patterns are created, the command "write_patterns" is used to write them.
An example of a command is as follows:

write_patterns /home/Praveen/5.ATPG/coreaparallelscan.v -Verilog -parallel - Patterns_set -scan -parameter_list {SIM_TOP_NAME parallel_TB SIM_KEEP_PATH 1 SIM_STATUS_MSG 1} -verbose -replace. This command writes the patterns along with the specified parameters. The "– Verilog" option indicates that the pattern will be written in Verilog, though other formats, such as WGL, STIL, and SystemVerilog can also be used. The "–parallel" option specifies that the patterns are for parallel or serial tests. The "patterns_set" switch is used to create patterns for chain or scan patterns. The "–parameter_list" switch is used to generate the required parameters for writing the patterns, such as the simulation's top-level name, parallel test bench, simulation path, and status message. write_patterns /home/Praveen/5.ATPG/coreaparallelscan.v -Verilog -parallel - Patterns_set -scan -parameter_list {SIM_TOP_NAME parallel_TB SIM_KEEP_PATH 1 SIM_STATUS_MSG 1} -verbose -replace. This command writes the patterns along with the specified parameters. The "– Verilog" option indicates that the pattern will be written in Verilog, though other formats, such as WGL, STIL, and SystemVerilog can also be used. The "–parallel" option specifies that the patterns are for parallel or serial tests. The "patterns_set" switch is used to create patterns for chain or scan patterns. The "–parameter_list" switch is used to generate the required parameters for writing the patterns, such as the simulation's top-level name, parallel test bench, simulation path, and status message.

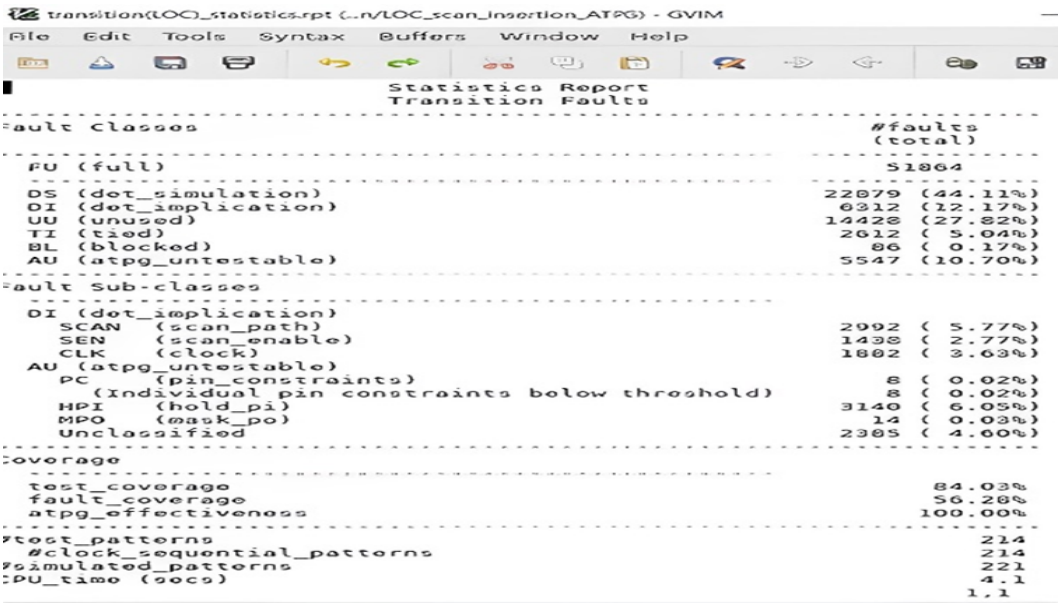


Figure 11 Statistical report of transition faults

4.4Patterns simulations

After verifying the circuit's functionality and generating the test vectors using the tool, proceed to execute the tool-generated test vectors in both no-timing and timing-based simulators. Compare the outcomes with the anticipated behavior projected by the ATPG tools. If no discrepancies are detected between the expected values from the ATPG tool and the results from the simulator, a message affirming the absence of errors between simulated and expected patterns is reported.

Simulation mismatches occur when the results from simulations do not align with the expected or predicted outcomes. These mismatches may arise during various phases of the design and testing process, such as when comparing tool-generated test vectors and simulated results. Addressing and resolving these mismatches is crucial for ensuring the accuracy and reliability of the testing procedures in digital circuit design, as displayed in *Figure 12.*

In the event of a mismatch in the ATPG pattern, it is crucial to identify the specific chain where the mismatch occurs and debug the specific flip-flop causing the failure. The command encompasses the series of simulation steps followed during the simulation process, including the elaborating and compilation of library files and netlists.

```
# Loading work.srff_10(fast)
# Loading work.srff_9(fast)
# Loading work.srff_8(fast)
# Loading work.srff_7(fast)
# Loading work.srff_6(fast)
# Loading work.srff_5(fast)
# Loading work.srff_4(fast)
# Loading work.srff_3(fast)
# Loading work.srff_2(fast)
# Loading work.srff_1(fast)
# Loading work.ao22(fast)
# Loading work.or04(fast)
# Loading work.clock_or02(fast)
# Loading work.mux21(fast)
# run -all
# Loading LOC_chain_serial.v.0.vec
#
# Begin chain test
#
# End chain test
#
# No error between simulated and expected patterns
#
# ** Note: $finish    : LOC_chain_serial.v(1163)
#    Time: 5962 ns  Iteration: 0  Instance: /ChipTop_LOC_chain_serial_v_ctl
# End time: 11:27:17 on Nov 24,2021, Elapsed time: 0:00:01
# Errors: 0, Warnings: 5
```

Figure 12 No mismatches in serial simulation

If an "error between simulated and expected patterns" occurs, it is advisable to inspect the mismatch flip-flop in the simulator. Simulation tools such as NcSim and XCelium (Cadence), Verilog Compiler Simulator, QuestaSim, and ModelSim (Mentor Graphics) can be used for this purpose. Identifying and correcting discrepancies between the simulated and expected patterns is crucial for ensuring accurate and reliable testing results, as shown in *Figure 13.*

Following this, serial and parallel test benches are generated from the ATPG tool to conduct the simulationsThe simulation is executed in two modes: unified command-line interface in command mode, denoted by the "vsim -c" command, and graphical mode, denoted by the "vsim -gui" command. When operating in command mode, it is crucial to include "-do run -all" to execute the entire hierarchy. Debugging involves two methods. The first is a divide-and-conquer algorithm, where an issue is recursively broken down into related sub-issues until they are manageable enough for direct resolution. The second method is an iterative approach, where flip-flops are divided into segments within a chain, and each segment undergoes testing, with "error-free" segments disregarded. The remaining portions are iteratively divided and tested until the root cause of the error is identified. Loading the hierarchy based on

the specified chain and flip-flop number allows to highlight of the failing chain. An additional signal, “fail to compare,” is used, where a transition from 1 to 0 or 0 to 1 signifies a change. By identifying the level at which the failure occurs, the corresponding flip-flop can be loaded, and examine parameters, such as reset, clock, and scan_en pins. This detailed inspection helps diagnose and address the issue, as shown in *Figure 14.*

```
9570ns: Expected  response xxxxxxx00 pattern          6 cycle        239
9570ns: Mismatch at pin          1 name scan_out1, Simulated 1, Expected 0
9570ns: Mismatch on chain: chain1 cell:          3
9610ns: Simulated response 0000111011 pattern          6 cycle        240
9610ns: Expected  response xxxxxxx01 pattern          6 cycle        240
9610ns: Mismatch at pin          1 name scan_out1, Simulated 1, Expected 0
9610ns: Mismatch on chain: chain1 cell:          4
simulated and expected patterns
```

Figure 13 Simulated mismatches in serial patterns

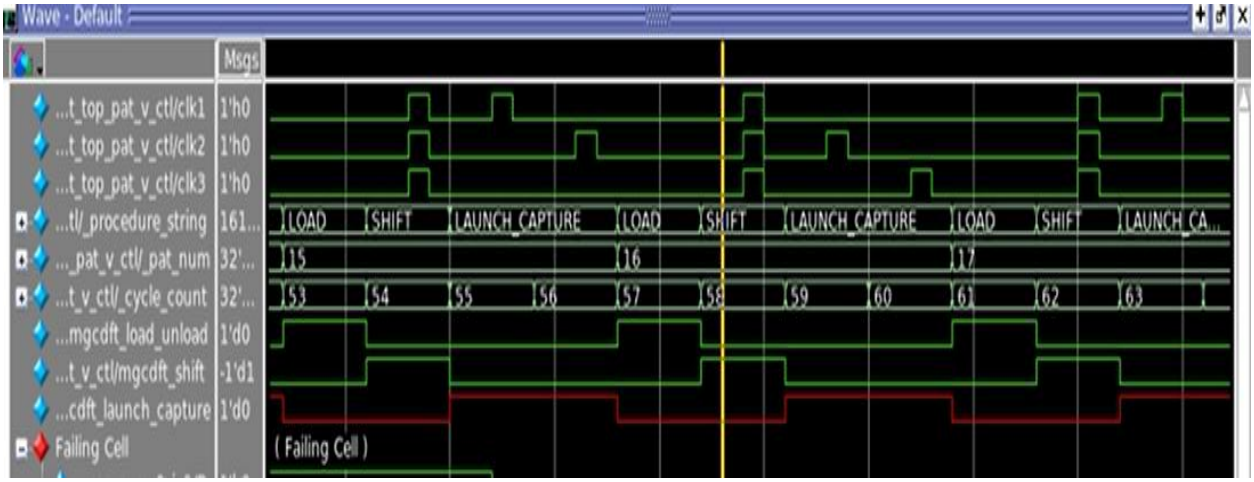


Figure 14 Basic requirements of signal loading in serial simulation

Figure 15 illustrates the anticipated output signals on the serial bus, derived from both the simulated and expected buses. A comprehensive comparison is conducted between these two buses. Similarly, the same procedure needs to be executed for the serial scan (capture) patterns. During this phase, the focus is on the flip-flops operating in capture mode, where the scan_en signal is set to 1. The entire dataset is conveyed through the SI pin, exclusively originating from this specific input.

Scan-based testing is a widely approach used for evaluating complex digital circuits. In this approach, additional flip-flops, referred to as scan cells, are embedded into the circuit. These scan cells facilitate testing by introducing TT and capturing the resulting outcomes at the circuit's output. Serial scan capture patterns play a pivotal role in this procedure, as they seize the internal state of the circuit within the scan cells. These patterns usually comprise a sequence of 0’s and 1’s, which are sequentially shifted into the scan chain. Subsequently, the values in the scan cells are captured and thoroughly analyzed.

The careful selection of scan capture patterns is crucial for achieving enhanced fault coverage and minimizing testing time. These patterns must be designed to detect all potential faults in the circuit while reducing the number of patterns required for efficient testing. ATPG tools can be used to generate these patterns, utilizing advanced algorithms to create transition delays that meet specified fault coverage and testing time criteria.

The simulation of serial scan patterns involves examining the chain and scan patterns. The message "No error between the simulated and expected patterns," indicates a flawlessly designed circuit, free from any issues or defects, as illustrated in *Figure 16.*

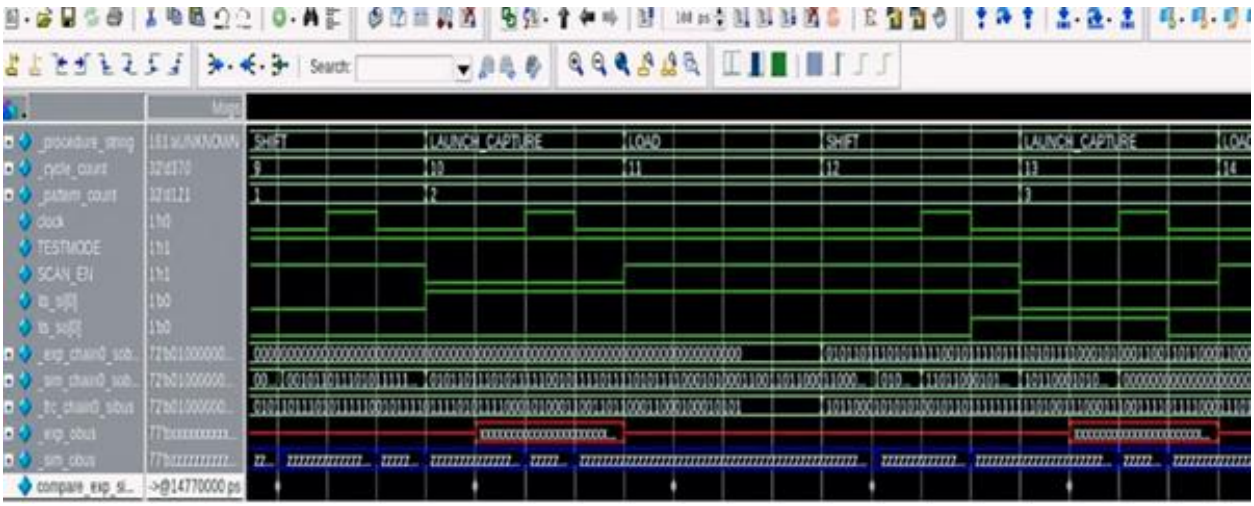


Figure 15 loading of pattern number and compare fail signal

```
# Loading work.CLA_4Bits_2(fast)
# Loading work.CLA_4Bits_1(fast)
# Loading work.CLA_Gen_4Bits_1(fast)
# Loading work.CLA_Gen_2Bits(fast)
# Loading work.power_controller(fast)
# Loading work.srff_0(fast)
# Loading work.srff_12(fast)
# Loading work.srff_11(fast)
# Loading work.srff_10(fast)
# Loading work.srff_9(fast)
# Loading work.srff_8(fast)
# Loading work.srff_7(fast)
# Loading work.srff_6(fast)
# Loading work.srff_5(fast)
# Loading work.srff_4(fast)
# Loading work.srff_3(fast)
# Loading work.srff_2(fast)
# Loading work.srff_1(fast)
# Loading work.ao22(fast)
# Loading work.or04(fast)
# Loading work.clock_or02(fast)
# Loading work.mux21(fast)
# run -all
# Loading stuck_scan_serial.v.0.vec
#
# No error between simulated and expected patterns
#
# ** Note: $finish    : ./stuck_scan_serial.v(1201)
#    Time: 364122 ns  Iteration: 0  Instance: /ChipTop_stuck_scan_serial_v_ctl
# End time: 11:31:00 on Nov 25,2021, Elapsed time: 0:00:03
# Errors: 0, Warnings: 0
[satyanarayanaccdft4@GLS Stuckat_Sim_debug]$ vsim -gui -debugDB -voptargs="+acc" Chi
pTop_stuck_scan_serial_v_ctl    -l sims_scan_serial.log   +nospecify +nowarmTSCALE
```

Figure 16 Scan or capture patterns simulation result

Parallel simulation and debugging represent a robust strategy within ATPG for ICs. This method involves running multiple simulations simultaneously on a multi-core processor, significantly accelerating the testing and debugging processes. The subsequent step involves simulating TT, where generated test vectors are applied to the circuit's inputs, and the outputs are analyzed. If errors or faults are detected during simulation, debugging is initiated to analyze the circuit's behavior and identify the root cause of the problem. Parallel simulation is employed to accelerate the debugging process by running multiple simulations simultaneously on a multi-core processor, thereby significantly reducing debugging

time. Once errors or faults are identified, corrective actions are implemented by modifying the circuit design or adjusting the TT. This iterative process is repeated until the circuit passes all tests and meets the specified requirements.

Figure 17 displays the loaded waveforms from parallel simulation, which is inherently simpler than serial simulation. In this approach, specifications for a particular hierarchy are designated, making it easier to identify mismatches. If a mismatch is detected, loading the flip-flop allows for checking parameters, such as reset, clock, and test mode (enable). After verifying the correctness of all pins, proceed with tracing the input pin hierarchies, either backward or forward, to identify faults in the simulation. A comparative analysis can be conducted using the open_visualizer tool. If mismatches persist, imposing constraints may be necessary to resolve discrepancies.

Debugging parallel scan pattern failures is a systematic approach designed to identify and resolve issues within the circuit design. When mismatches occur between simulated and expected patterns, the process begins with a detailed analysis of the failing patterns, as shown in *Figure 18.* This involves isolating specific chains or flip-flops affected by the failure. Signal tracing plays a vital role in focusing on key signals, such as scan_en, clock, and reset. Hierarchical debugging facilitates a targeted investigation, effectively narrowing the scope for efficient analysis. Techniques such as backward tracing from failing points and forward tracing from the inputs help reveal the root causes and interactions leading to the observed failure. Simulation tools, including waveform viewers and debuggers, are essential for visualizing signal behavior. Constraint analysis ensures that the applied constraints align with the intended testing conditions, allowing necessary adjustments to refine the process. Techniques such as force and release, which involves manipulating signals during simulation, aid isolate issues and assess their impact on the circuit. Iterative testing plays a major role, involving incremental changes to patterns or the design, systematically narrowing the root cause of the failure. Collaboration and documentation play crucial roles, with team input and comprehensive documentation ensuring insights are shared and findings are well-documented for future reference. Through this systematic approach, debugging parallel scan patterns becomes a structured and effective process, enhancing the reliability of the testing methodology.

Figure 19 presents the outcome after debugging the mismatches identified in the design. If no discrepancies are found between the simulated and expected patterns, it confirms that the design is free of issues.

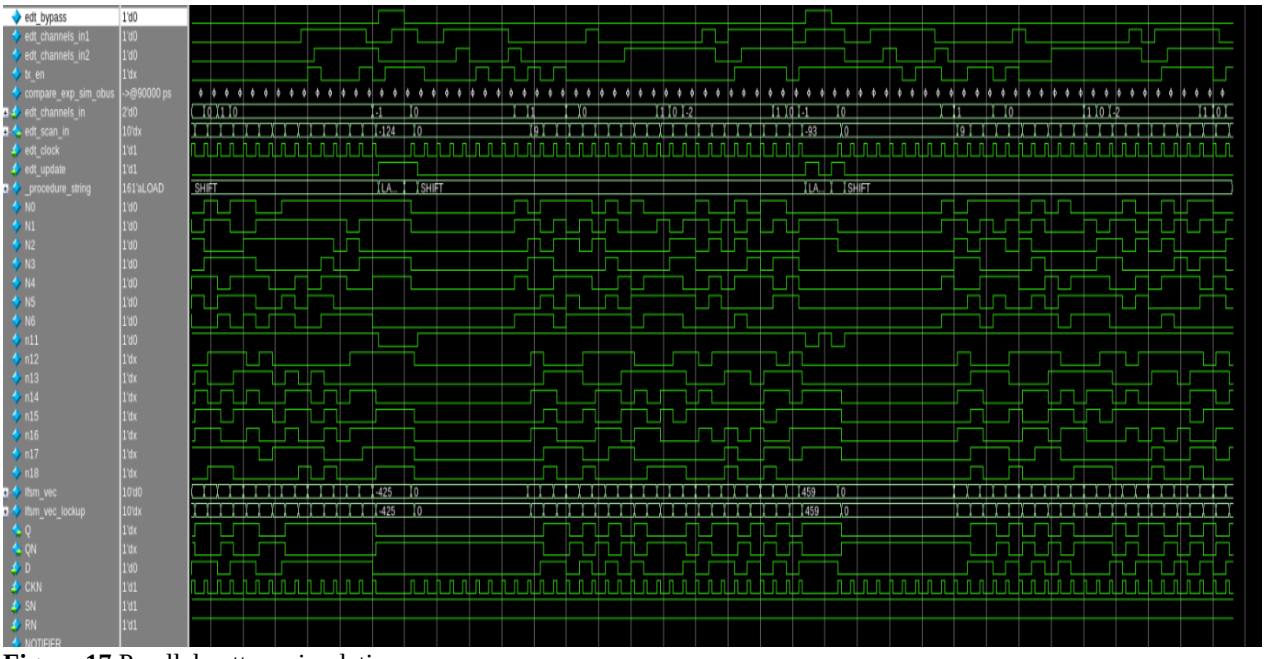


Figure 17 Parallel pattern simulations

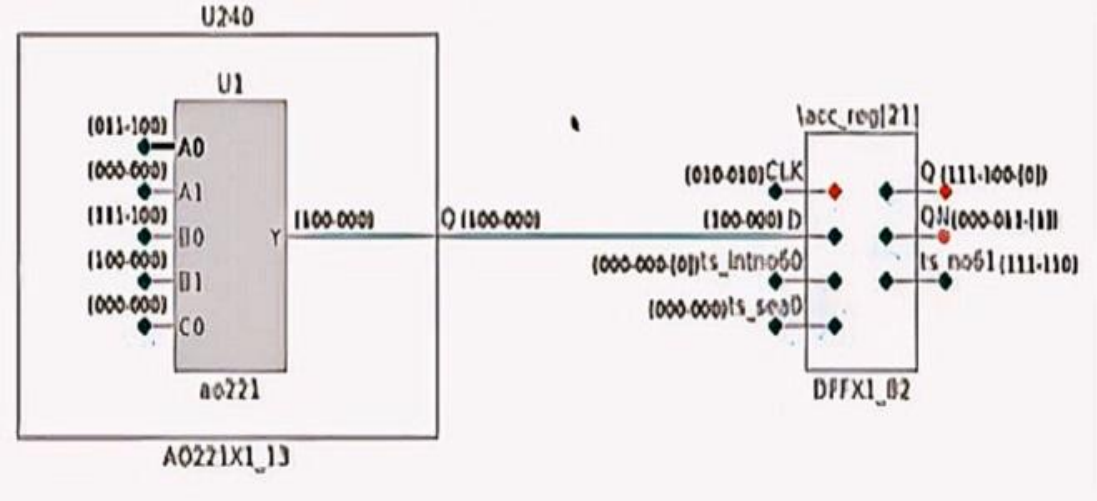


Figure 18 Schematic loading of the failure flip-flop

```
# Loading work.srff_7(fast)
# Loading work.srff_6(fast)
# Loading work.srff_5(fast)
# Loading work.srff_4(fast)
# Loading work.srff_3(fast)
# Loading work.srff_2(fast)
# Loading work.srff_1(fast)
# Loading work.ao22(fast)
# Loading work.or04(fast)
# Loading work.clock_or02(fast)
# Loading work.mux21(fast)
# run -all
# Loading stuck_scan_parallel.v.0.vec
#
# No error between simulated and expected patterns
#
# ** Note: $finish    : ./stuck_scan_parallel.v(4195)
#    Time: 14802 ns  Iteration: 0  Instance: /ChipTop_stuck_scan_parallel_v_ctl
# End time: 14:13:21 on Nov 25,2021, Elapsed time: 0:00:01
# Errors: 0, Warnings: 0
[satyanarayanaccdft4@GLS Stuckat_Sim_debug]$ vsim -gui -debugDB -voptargs="+acc"
 ChipTop_stuck_scan_parallel_v_ctl    -l sims_scan_parallel.log   +nospecify +no
warmTSCALE
```

Figure 19 Parallel patterns simulated output

5.Discussions and limitations

5.1Discussion

As outlined in this research, a theoretical and practical experiment was conducted to compare the pre-compression and post-compression analysis of test data and TDV, with the results detailed in the respective section. The EDT tool proved to be a powerful software solution, significantly enhancing efficiency and reducing workload. Given the scale of application involving millions of transistors, manual testing of a single chip becomes impractical. EDT emerges as an optimal solution for generating ATPG patterns and facilitating comparative analyses [41].

5.2Key findings of compression in DFT

Compression techniques in DFT offer several key findings that significantly impact chip testing processes. A primary advantage is a substantial reduction in TDV, streamlining testing procedures, and decreasing TT. This efficiency improvement facilitates quicker time-to-market for electronic products. Additionally, compression enhances test access efficiency by optimizing the utilization of test access mechanisms, such as scan chains, leading to more thorough testing of electronic systems. However, implementing compression in DFT introduces added design complexity due to the incorporation of compactor and decompressor circuitry along with the associated control logic. Furthermore, trade-offs between CR and overhead

must be carefully managed, as achieving higher CR often leads to increased design complexity and resource requirements. Despite these challenges, compression offers significant benefits in testing efficiency and cost reduction. To fully leverage its advantages, it is crucial to balance these trade-offs and address the complexities during the chip design process.

5.3Design implications

The integration of compression techniques in DFT brings significant implications for chip design. Primarily, compression aims to reduce the volume of test data needed for chip testing, thereby impacting the design of the test architecture. This reduction in data volume leads to improved TT efficiency, influencing the optimization of test scheduling and execution mechanisms. Additionally, compression enhances the efficiency of test access mechanisms, such as scan chains, which may require adjustments in their layout and routing within the chip [42]. Moreover, the improved testability provided by compression may require considerations for fault tolerance and error recovery mechanisms in the design. However, implementing compression introduces complexities, particularly in the form of additional circuitry and control logic. Designers must balance CR and overhead to optimize chip performance and efficiency. Additionally, the interaction between compression techniques and existing test methodologies requires careful consideration to ensure seamless integration and effectiveness in chip testing. Overall, integrating compression into DFT demands a comprehensive design approach, weighing various trade-offs to maximize benefits while minimizing drawbacks.

5.4Comparative analysis of the compression

Figure 20 presents a comparative analysis of the design when the internal scan chains are modified. As the internal scan chains are altered, the CR stages change while keeping the external channels fixed. The compression data volume also varies depending on the design complexity. Fixing the CR and the number of shift cycles results in variations based on the internal scan chains, as illustrated in *Figure 21*.

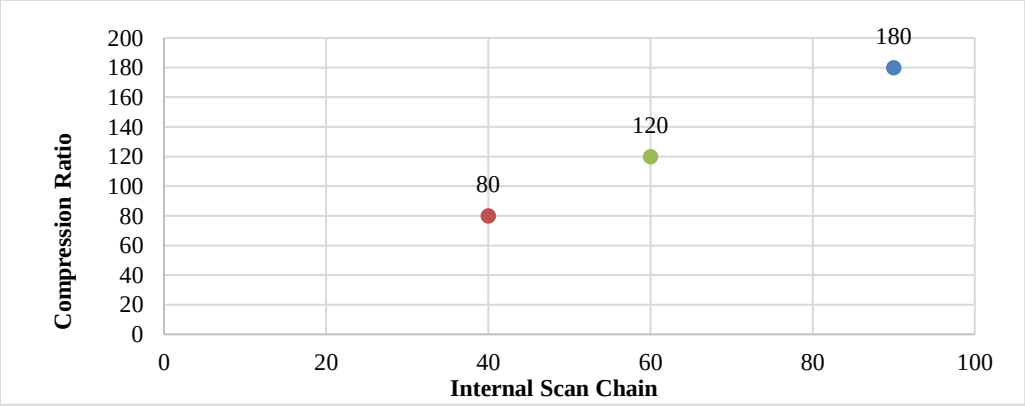


Figure 20 Graph for comparative analysis of compression for changing internal scan chains

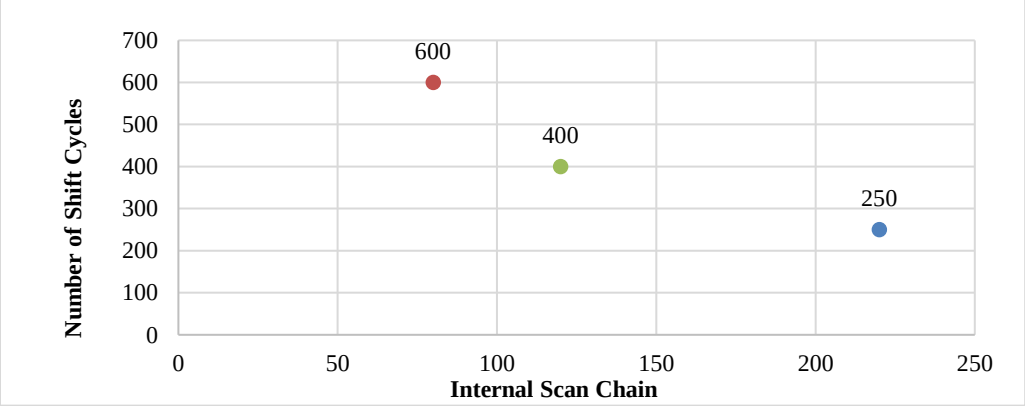


Figure 21 Variation of the number of shift cycles by changing internal scan chains

5.5Limitations of the compression

Compression techniques in chip testing face certain limitations within the testing environment. These are as under:

Limited fault coverage: Compression in fault coverage can result in coverage loss. Fault coverage depends on the CR; higher CR typically leads to lower fault coverage, while lower CR improves coverage.

Area overhead: Implementing EDT structures in the design requires extra area, which can affect the overall chip size in resource-constrained designs. This, in turn, may lead to higher manufacturing costs. Furthermore, the design introduces additional constraints and logic.

Impact on performance: The insertion of embedded test structures may impact the circuit's performance, particularly in applications where high-speed performance is essential. In such cases, any added delay caused by the testing structures is undesirable [43].

Dependency on scan chains: When EDT is inserted into the design, the scan chain structure is modified to integrate with the internal design. Adding more scan chains enhances fault coverage, improving the overall test efficiency.

Complexity in debugging: Debugging EDT-related issues can be more challenging than traditional methods. Loading patterns into serial simulations without hierarchy can complicate the process. Identifying the root cause of a fault or diagnosing issues within embedded test structures often requires specialized knowledge.

Complexity in fabrications: The testing process in design manufacturing is complex and involves multiple stages. These processes use different finite-state machines to test various aspects of the design [44].

Dependency on DFT tools: The successful implementation of EDT often relies on the capabilities of DFT tools. The effectiveness of EDT is, to some extent, dependent on the accuracy, efficiency, and integration capabilities of the tools used.

Some of the future suggestions are as under:

1. With the increasing complexity of digital circuits, driven by higher integration of components and functionalities, the demand for advanced DFT techniques is expected to grow significantly. Future research should focus on addressing the challenges of efficient and scalable testing methodologies for intricate designs, ensuring enhanced fault detection and reduced test overhead.
2. Continuous advancements in test compression are essential for managing TDV and TT efficiently. Future research in DFT can explore innovative compression algorithms and hardware-efficient techniques to minimize TDV and TT while maintaining high fault coverage and test accuracy.
3. As digital systems become more vulnerable to security threats, DFT methodologies must evolve to incorporate hardware security mechanisms. Future research should focus on integrating advanced techniques to detect and mitigate hardware Trojans, counterfeit components, and other security vulnerabilities, ensuring the trustworthiness and integrity of digital circuits [45].

A complete list of abbreviations is listed in *Appendix I.*

6.Conclusion

This research highlighted the significance of DFT methodologies in improving test efficiency, fault coverage, and test data compression for complex digital circuits. By integrating SI, ATPG, EDT, and pattern simulation, the study effectively reduces TDV and TT while maintaining high fault detection capabilities. The compression analysis conducted in this study demonstrates a substantial improvement, with TT reduced from 15.84 ms to 3.9 ms and TDV minimized from 14,400 bits to 3,600 bits. The application of ATPG and EDT enhances fault detection by optimizing test patterns for stuck-at and transition faults. Additionally, simulation and debugging techniques, including parallel and serial pattern analysis, validate the effectiveness of generated test patterns, ensuring accurate and efficient fault diagnosis. With the growing complexity of modern ICs, the role of DFT methodologies continues to expand. The findings of this study reinforce the importance of DFT strategies in enhancing IC reliability, reducing manufacturing costs, and accelerating time-to-market.

Acknowledgment

None.

Conflicts of interest

The authors have no conflicts of interest to declare.

Data availability

The data for this study is derived from the confidential CPU_TOP project, focusing on the minimum number of flip-flops required. Waveforms and figures were obtained

using Mentor's Tessent tool, which is available upon reasonable request from the corresponding author.

Author's contribution statement

Praveen K: Conception of the DFT methodology, design, investigation of challenges, writing the original draft, analysis and interpretation of results, and writing, reviewing, and editing. **Dr. Rajanna G S:** Results verification, supervision, addressing challenges, as well as reviewing and editing. **Dr. Shivakumara Swamy G M:** Data modifications, issue resolution, verification of the DFT flow, draft preparation, and results review.

References

[1] Mojtabavi NM, Ooi CY. A novel scan architecture for low power scan-based testing. VLSI Design. 2015; 2015(1):1-13.

[2] Trivedi R, Dhariwal S, Kumar A. Comparison of various ATPG techniques to determine optimal BIST. In international conference on intelligent circuits and systems 2018 (pp. 93-8). IEEE.

[3] Borda P, Prajapati P. LOC, lOS and LOES at-speed testing methodologies for automatic test pattern generation using transition delay fault model. International Journal of Research in Engineering and Technology. 2014; 3(3):273-7.

[4] Shantagiri PV, Kapur R. Aggressive exclusion of scan flip-flops from compression architecture for better coverage and reduced TDV: a hybrid approach. Journal of Low Power Electronics and Applications. 2019; 9(2):1-25.

[5] Patel P, Rajawat A, Jain P. Comparative analysis of simulation techniques: scan compression and internal scan. International Journal on Cybernetics & Informatics. 2023; 12:15-22.

[6] Xiang D. Test compression for launch-on-capture transition fault testing. ACM Transactions on Design Automation of Electronic Systems. 2023; 29(1):1-20.

[7] Shukla NK, Mayet AM, Raja MR, Parayangat M, Usman M, Verma R, et al. A unified test data volume compression scheme for circular scan architecture using hosted cuckoo optimization. The Journal of Supercomputing. 2024; 80(5):6411-34.

[8] Pandey K. A critical engineering dissection of LOS and LOC at-speed test approaches. In IEEE international test conference India 2020 (pp. 1-7). IEEE.

[9] Murugan SK, Prabha MM. Launch off shift and capture power reduction in transition fault test based on design for testability methods. International Journal of Research in Computer Applications and Robotics, India. 2015; 3(2):40-9.

[10] Thobhani P, Mehta U. Development of ATPG using basic algorithm processes. International Engineering Journal for Research & Development. 2021; 6:1-8.

[11] Yang KC, Lee MT, Wu CH, Li JC. ATPG and test compression for probabilistic circuits. In international symposium on VLSI design, automation and test (VLSI-DAT) 2019 (pp. 1-4). IEEE.

[12] Chandrashekar C, Neelgar BI. Pattern generation techniques for BIST. International Journal of Research and Analytical Reviews. 2020; 7(3):355-9.

[13] Das KS, Zala A. Optimizing cell-aware ATPG pattern volume to keep test cost competitive. In international conference on electronics, computing and communication technologies 2020 (pp. 1-6). IEEE.

[14] Singh VK, Sarkar T, Pradhan SN. Power-aware testing for maximum fault coverage in analog and digital circuits simultaneously. IETE Technical Review. 2022; 39(6):1395-409.

[15] Kumar YG, Kariyappa BS, Kurian MZ. Design, implementation and performance analysis of test pattern generator for built-in self-test using m-GDI technology. Indian Journal of Science and Technology. 2022; 15(5):221-6.

[16] Liang ZJ, Wu YT, Yang YF, Li JC, Chang N, Kumar A, et al. High-speed, low-storage power and thermal predictions for ATPG test patterns. In international test conference 2023 (pp. 206-15). IEEE.

[17] Ramasamy J, Samiappan D. Design and Analysis of Non-vulnerable PRBS Generation with Internal Shift and XOR of LFSR. In international conference on electrical and electronics engineering 2023 (pp. 573-582). Singapore: Springer Nature Singapore.

[18] Ye CS, Zheng SX, Tsai FJ, Wang C, Lee KJ, Cheng WT, et al. Efficient test compression configuration selection. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems. 2021; 41(7):2323-36.

[19] Tsai FJ, Ye CS, Lee KJ, Zheng SX, Huang Y, Cheng WT, et al. Prediction of test pattern count and test data volume for scan architectures under different input channel configurations. In international test conference 2020 (pp. 1-10). IEEE.

[20] Wang F, Gupta SK. An effective and efficient automatic test pattern generation (ATPG) paradigm for certifying performance of RSFQ circuits. IEEE Transactions on Applied Superconductivity. 2020; 30(5):1-11.

[21] Wang F, Gupta S. Automatic test pattern generation for timing verification and delay testing of RSFQ circuits. In 37th VLSI test symposium 2019 (pp. 1-6). IEEE.

[22] Kung YC, Lee KJ, Reddy SM. Generating single-and double-pattern tests for multiple CMOS fault models in one ATPG run. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems. 2019; 39(6):1340-5.

[23] Chen KH, Chen CY, Huang JL. Testability measures considering circuit reconvergence to reduce ATPG runtime. In 22nd international symposium on design and diagnostics of electronic circuits & systems 2019 (pp. 1-12). IEEE.

[24] Nekar H, Siddamal SV. Design and implementation of novel scan architecture for test power reduction. Procedia Computer Science. 2020; 171:2556-62.

[25] Suteddy W, Adiwilaga A, Atmanto DA. Fault coverage testing on the ISCAS'89 S1423 sequential circuit using scan based design and synopsis tetramax.

Journal of Computer Engineering, Electronics and Information Technology. 2022; 1(2):103-16.
[26] Kongala S, Gupta A, Walia Y, Jain S. Novel methodology to optimize TAT and resource utilization for ATPG simulations for large SoCs. In international test conference 2023 (pp. 60-4). IEEE.
[27] Mohan M, Shrimali T. Hybrid data approach for selecting effective test cases during the regression testing. International Journal on Smart Sensing and Intelligent Systems. 2017; 10(5):1-24.
[28] Dannina SSR, Kumar S. Efficient GDI RAM design and low power clock splitting based ATPG for BIST. International Journal of Advanced Science and Technology. 2019; 28(19):1152-63.
[29] Zhang G, Yuan Y, Liang F, Wei S, Yang CF. Low cost test pattern generation in scan-based BIST schemes. Electronics. 2019; 8(3):1-11.
[30] Srivastava A, Abraham J. Low capture power at-speed test with local hot spot analysis to reduce over-test. In international test conference 2022 (pp. 446-55). IEEE.
[31] Ni T, Wen X, Amrouch H, Zhuo C, Song P. Introduction to the special issue on design for testability and reliability of security-aware hardware. ACM Transactions on Design Automation of Electronic Systems. 2023; 29(1):1-3.
[32] Pomeranz I. Path unselection for path delay fault test generation. IEEE Transactions on Very Large Scale Integration (VLSI) Systems. 2022; 31(2):267-75.
[33] Li M, Lin Y, Gupta S. Design for testability (DFT) for RSFQ circuits. In 41st VLSI test symposium 2023 (pp. 1-7). IEEE.
[34] Chauhan J, Panchal C, Suthar H. Scan methodology and ATPG DFT techniques at lower technology node. In international conference on computing methodologies and communication 2017 (pp. 508-14). IEEE.
[35] Gopikrishna K, Jindal P. Test coverage analysis of DFT with EDT and without EDT architecture. In 12th international conference on computing communication and networking technologies 2021 (pp. 1-4). IEEE.
[36] Cai S, Zhou Y, Liu P, Yu F, Wang W. A novel test data compression approach based on bit reversion. IEICE Electronics Express. 2017; 14(13):1-11.
[37] Roy S, Stiene B, Millican SK, Agrawal VD. Improved pseudo-random fault coverage through inversions: a study on test point architectures. Journal of Electronic Testing. 2020; 36:123-33.
[38] Javvaji PK, Tragoudas S. Test pattern generation and critical path selection in the presence of statistical delays. IEEE Transactions on Very Large Scale Integration (VLSI) Systems. 2019; 28(1):163-73.
[39] Kundu S, Abraham J. Revisiting test compression configuration in context of multi-core testing using packetized scan network. In 37th international conference on VLSI design and 23rd international conference on embedded systems (VLSID) 2024 (pp. 724-9). IEEE.
[40] Talatule SD, Zode P, Zode P. A secure architecture for the design for testability structures. In 19th international symposium on VLSI design and test 2015 (pp. 1-6). IEEE.
[41] Iwata H, Maeda Y, Matsushima J, Laouamri O, Khanna N, Mayer J, et al. A new framework for RTL test points insertion facilitating a “Shift-Left DFT” strategy. In international test conference 2023 (pp. 1-10). IEEE.
[42] Chao Z, Zhang X, Huang J, Ye J, Cai S, Li H, et al. A fast test compaction method for commercial DFT flow using dedicated pure-MaxSAT solver. In 29th Asia and South Pacific design automation conference 2024 (pp. 503-8). IEEE.
[43] Ahmed N, Tehranipoor M, Ravikumar CP. Enhanced launch-off-capture transition fault testing. In international conference on test 2005. IEEE.
[44] Ansari R, Aghdasi P, Shahnazari A. DFT-based finite element analysis of compressive response in armchair phosphorene nanotubes. Journal of Molecular Graphics and Modelling. 2024; 129:108751.
[45] Pomeranz I. Sharing of topped-off compressed test sets among logic blocks. IEEE Access. 2024; 12:49895-903.

Praveen K is a DFT Engineer at Siliciom Technologies Pvt Ltd. He holds a BE in Electronics and Communication Engineering and an M.Tech in Digital Electronics. He is currently a Research Scholar affiliated with Srinivas University, Mangaluru. With extensive experience in both academia and industry, he previously served as an Assistant Professor in the Electronics and Communication Engineering Department at BIET, Davangere, for six years and at Bheema Institute of Technology and Science, Adoni, Andhra Pradesh, for three years. In the industry, he has worked on projects for esteemed clients such as Siemens, Intel, Google, and Tech Mahindra Cerium. His research interests include VLSI Design, Digital System Design, DFT, RTL design, as well as Front-End and Back-End Design.
Email: Praveen.kuntanahal@gmail.com

Dr. Rajanna G S is currently a Research Professor in the Department of Electronics and Communication Engineering at the Institute of Engineering and Technology, affiliated with Srinivas University, Mangaluru, Karnataka, India. His teaching career began in 1984 at SJMIT, Chitradurga, Karnataka, and he has since held various academic positions. Dr. Rajanna earned his bachelor's degree from M. S. Ramaiah Institute of Technology, Bangalore University, Karnataka, his M.Tech from Walchand College of Engineering, Sangli, Shivaji University, and his Ph.D. from JNTU Hyderabad. With 35 years of teaching experience, his research interests include Power

Electronics, Control Systems, Artificial Intelligence, Digital System Design, and Soft Computing.
Email: kgsrajanna@gmail.com

Dr. Shivakumara Swamy G M is an Assistant Professor in the Department of Electrical and Electronics Engineering at Bapuji Institute of Engineering and Technology, affiliated with Visvesvaraya Technological University, Davangere, Karnataka, India. He has been working at Bapuji Institute of Engineering and Technology since 2004. Dr. Shivakumar earned his bachelor's degree from Kuvempu University, Shivamogga, in 2000, his Master's degree from Visvesvaraya Technological University, Belagavi, in 2005, and completed his PhD at Srinivas University, Mangaluru in 2023. With over 19 years of teaching experience, his research interests include Privacy Preserving-Data, Artificial Intelligence, Power Systems, Renewable Energy, and Digital System Design.
Email: gms20mar@gmail.com

Appendix I

S. No.	Abbreviation	Description
1	ATPG	Automatic Test Pattern Generation
2	ASCII	American Standard Code for Information Interchange
3	BIST	Built-In Self-Test
4	CMOS	Metal-Oxide-Semiconductor
5	DC	Design Compiler
6	DFT	Design for Testability
7	DRC	Design Rule Check
8	EDA	Electronic Design Automation
9	EDT	Embedded Deterministic Testing
10	IC	Integrated Circuit
11	JTAG	Joint Test Action Group
12	JJ	Josephson Junctions
13	LOC	Launch-On-Capture
14	LOS	Launch -On- Shift
15	LFSR	Linear Feedback Shift Register
16	MGDI	Modified Gate Diffusion Input
17	MCL	Maximum Chain Length
18	PC	Pattern Count
19	RAM	Random Access Memory
20	RSFQ	Rapid Single Flux Quantum
21	RTL	Register-Transfer Level
22	SI	Scan Insertion
23	SOC	System on Chip
24	STIL	Standard Test Interface Language
25	STF	Slow to Fall
26	TC	Test Coverage
27	TDV	Test Data Volume
28	TPC	Test Pattern Count
29	TT	Test Time
30	VCS	Verilog Compiler Simulator
31	VLSI	Very Large-Scale Integration
32	WGL	Waveform Generating Language