Snigdha Jain

PrashantKumar_24vls04_Final_Thesis

Assignment1

Document Details

Submission ID
trn:oid:::27535:142311216

Submission Date
Jun 9, 2026, 6:32 PM GMT+5:30

Download Date
Jun 9, 2026, 6:36 PM GMT+5:30

File Name
PrashantKumar_24vls04_Final_Thesis.pdf

File Size
5.3 MB

43 Pages

9,188 Words

51,577 Characters

6% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

Exclusions

- 19 Excluded Matches

Match Groups

- **46** Not Cited or Quoted 5%
  Matches with neither in-text citation nor quotation marks
- **0** Missing Quotations 0%
  Matches that are still very similar to source material
- **2** Missing Citation 0%
  Matches that have quotation marks, but no in-text citation
- **3** Cited and Quoted 0%
  Matches with in-text citation present, but no quotation marks

Top Sources

2% Internet sources
4% Publications
4% Submitted works (Student Papers)

Match Groups

- **46** Not Cited or Quoted 5%
  Matches with neither in-text citation nor quotation marks
- **0** Missing Quotations 0%
  Matches that are still very similar to source material
- **2** Missing Citation 0%
  Matches that have quotation marks, but no in-text citation
- **3** Cited and Quoted 0%
  Matches with in-text citation present, but no quotation marks

Top Sources

2% Internet sources
4% Publications
4% Submitted works (Student Papers)

Top Sources

The sources with the highest number of matches within the submission. Overlapping sources will not be displayed.

1	Publication	Tieqiao Liu, Chao Qian, Yijun Song. "An Effective Test-per-Clock Test Scheme Base... <1%
2	Publication	Atharva Yawalkar, Vrinda Gupta. "Low Power Testing: A Study of Power Aware Te... <1%
3	Student papers	Amrita Vishwa Vidyapeetham on 2025-04-25 <1%
4	Publication	"Power-Aware Testing and Test Strategies for Low Power Devices", Springer Scien... <1%
5	Internet	beei.org <1%
6	Student papers	Universiti Teknologi Malaysia on 2024-07-09 <1%
7	Publication	Kamlesh Pandey. "A Critical Engineering Dissection of LOS and LOC At-speed Test ... <1%
8	Internet	www.amrita.edu <1%
9	Publication	B Nagesh, B S Nikhil Chandra. "Designof Efficient Scan Flip-Flop", 2021 Internatio... <1%
10	Student papers	Indian Institute of Information Technology, Design and Manufacturing - Kanchee... <1%

11 Internet

www.semanticscholar.org <1%

12 Student papers

Dr. B R Ambedkar National Institute of Technology, Jalandhar on 2026-06-04 <1%

13 Publication

Sarthak Singhal, Subahsish Mukherjee, Christos Papameletis, Dale Meehl et al. "P... <1%

14 Student papers

Delhi Technological University on 2026-05-25 <1%

15 Student papers

Universiti Teknologi Malaysia on 2019-12-03 <1%

16 Student papers

Amrita Vishwa Vidyapeetham on 2026-05-11 <1%

17 Student papers

Poornima University on 2026-03-23 <1%

18 Internet

pdfcookie.com <1%

19 Internet

researchr.org <1%

20 Student papers

Multimedia University on 2024-05-08 <1%

21 Publication

Heetae Kim, Hyunggoy Oh, Sangjun Lee, Sungho Kang. "Low Power Scan Chain Ar... <1%

22 Student papers

National Institute of Technology Delhi on 2026-05-21 <1%

23 Student papers

PEC University of Technology on 2022-05-22 <1%

24 Internet

ir.nctu.edu.tw <1%

25 Internet

www.freepatentsonline.com <1%

26 Publication

Alpaslan, Elif, Yu Huang, Xijiang Lin, Wu-Tung Cheng, and Jennifer Dworak. "On R... <1%

27 Publication

Efi Arvaniti, Yiorgos Tsiatouhas. "Low-Power Scan Testing: A Scan Chain Partitioni... <1%

28 Student papers

Indian Institute of Information Technology, Design and Manufacturing - Kanchee... <1%

29 Internet

dspace.dtu.ac.in:8080 <1%

30 Publication

Digital Logic Design Using Verilog, 2016. <1%

31 Student papers

Higher College of Technology on 2021-04-05 <1%

32 Publication

Dong Xiang, Zhen Chen, Laung-Terng Wang. "Scan Flip-Flop Grouping to Compres... <1%

33 Publication

Geethu Remadevi Somanathan, Ramesh Bhakthavathchalu, Krishnakumar M. "Ba... <1%

34 Publication

Sai Sandeep Annamareddy, Navya Mohan. "Improved Scan Chain Stitching for Re... <1%

Power Domain Aware DFT Implementation with Scan Chains, IEEE 1500 Wrappers, and Launch-off Shift for Digital Designs Testability

Thesis Submitted in Partial Fulfillment of the Requirements for the Degree of

MASTER OF TECHNOLOGY
in
VLSI AND EMBEDDED SYSTEMS

Submitted by

Prashant Kumar

(2K24/VLS/04)

Under the supervision of

Prof. Neeta Pandey

DTU, Delhi



Department of Electronics and Communication Engineering
DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi College of Engineering)
Shahbad Daulatpur,Bawana Road, Delhi-110042, India

MAY, 2026



Department of Electronics and Communication Engineering
DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi College of Engineering)
Shahbad Daulatpur,Bawana Road, Delhi-110042, India

CANDIDATE'S DECLARATION

I, Prashant Kumar, Roll No. 2K24/VLS/04 hereby certify that the work which is being presented in the thesis entitled **'Power Domain Aware DFT Implementation with Scan Chains, IEEE 1500 Wrappers, and Launch-off Shift for Digital Designs Testability'** in partial fulfilment of the requirements for the award of the Degree of Master of Technology (VLSI and Embedded Systems), submitted in the Department of Electronics and Communication Engineering, Delhi Technological University, is an authentic record of my own work carried out during the period from January 2026 to May 2026 under the supervision of **Prof. Neeta Pandey**.

I have not submitted the matter presented in the thesis for the award of any other degree of this or any other Institute.

Candidate's Signature

Prashant Kumar
(2K24/VLS/04)



Department of Electronics and Communication Engineering
DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi College of Engineering)
Shahbad Daulatpur,Bawana Road, Delhi-110042, India

CERTIFICATE

This is to certify that the work reported in this thesis entitled "Power Domain Aware DFT Implementation with Scan Chains, IEEE 1500 Wrappers, and Launch-off Shift for Digital Designs Testability" has been carried out by Mr. Prashant Kumar (2K24/VLS/04) under my supervision.

Supervisor's Signature
Prof. Neeta Pandey
ECE, DTU, Delhi



Department of Electronics and Communication Engineering
DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi College of Engineering)
Shahbad Daulatpur,Bawana Road, Delhi-110042, India

PLAGIARISM VERIFICATION

Title of the Thesis: Power Domain Aware DFT Implementation with Scan Chains, IEEE 1500 Wrappers, and Launch-off Shift for Digital Designs Testability

Total Pages: 45 **Name of the Scholar:** Prashant Kumar

Supervisor: Prof. Neeta Pandey

Department: Electronics and Communication Engineering

This is to report that the above thesis was scanned for similarity detection. Process and outcome is given below:

Software used: Turnitin **Similarity Index:** 11%

Total Word Count: 9,213 **Date:** May 27, 2026

Candidate's Signature **Signature of Supervisor**

ACKNOWLEDGEMENT

The successful completion of any task is incomplete and meaningless without giving any due credit to the people who made it possible without which the project would not have been successful and would have existed in theory. I am grateful to all faculty members of our department for their constant guidance and support, constant motivation and sincere support and gratitude for this project work. I would owe a lot of thanks to my supervisor, Prof. Neeta Pandey, Department of Electronics and Communication Engineering, Delhi Technological University for igniting and constantly motivating me and guiding me in the idea of a creatively and amazingly performed Major Project 1 in undertaking this endeavour and challenge and also for being there whenever I needed her guidance or assistance.

Date: 29.05.2026

Prashant Kumar
2K24/VLS/04

Abstract

Conventional scan-based DFT operates circuits at maximum switching activity during shift, causing test power to exceed functional power by two to three times. This induces IR-drop, thermal stress, and yield loss absent during normal operation. Addressing this requires a unified methodology that maintains high fault coverage while controlling power throughout the test process.

This thesis implements a power-aware DFT flow across nine ITC'99 benchmark circuits (b01–b21), covering designs from 5 to 432 flip-flops, using three techniques in combination. An IEEE 1801 UPF architecture divides each design into an always-on boundary domain and a switchable core domain, with isolation cells held transparent during ATPG by treating the power-gate control as a test-mode constant. The Integrated Clock Gating (ICG) cells are applied to registers that are written to rarely, gating 320 flip-flops in the dual-processor b21 and 35 flip-flops in the b13 sensor interface, with specific scan-cgc-en ports that preserve complete scan chain access during shift. Structured boundary access is provided by IEEE 1500 core wrapper cells, and at-speed transition coverage is provided on deep ALU and address-adder paths with Launch-Off-Shift (LOS) stages. The flow was done in Cadence Genus 27.10 and confirmed with Cadence Modus 25.10.

This work provides an efficient and scalable hardware design flow for power-aware testability. It is applicable to complex SoC designs, as well as low-power digital systems, where high fault coverage is needed without exceeding power limits during testing as a key production requirement.

Contents

18 Candidate's Declaration i

Certificate ii

PLAGIARISM VERIFICATION iv

Acknowledgement iv

Abstract v

Content vi

List of Tables vii

List of Figures viii

1 INTRODUCTION ix
1.1 Motivation and Background ix
1.1.1 Evolution of Design for Testability Architectures ix
1.1.2 Role of Design for Testability in VLSI Designs x
1.1.3 Need for Integrating Power-Aware DFT in Digital Designs x
1.2 Problem Statement 1
1.3 Project Objectives 1

2 LITERATURE REVIEW 2

3 Design Methodology 9
3.1 Overview of the proposed methodology 9
3.2 Implementation of Power-Aware Wrapper Cells 9
3.3 Scan Chains Insertion 13
3.4 Launch-off Shift 15

4 RESULTS and DISCUSSION 16
4.1 Simulation Environment 16
4.2 Stuck-At & Dynamic Fault Coverage 17

5 CONCLUSION 27

6 Future Scope 29

A Appendices 30

List of Tables

4.1 Tool Chain 16

5.1 ATPG Coverage, Test Patterns, and Runtime (Modus FULLSCAN) 27
5.2 Area and Power: Pre-DFT vs Post-DFT 28

List of Figures

2.1 Design example with multiple Power Domains 2
2.2 Correct DWC insertion avoiding additional PD crossing 3
2.3 Isolation Cell 3
2.4 Balanced scan chains 4
2.5 Test data Compaction flow 5
2.6 LOC Scheme Timing 5
2.7 LOS Scheme Timing 6
2.8 Selection of normal and scan path 6
2.9 Scan Flip Flop 7
2.10 Before Scan Stitching 8
2.11 After Scan Stitching 8

3.1 Sink/Source Logic in hierarchy 10
3.2 DWC Optimization 11
3.3 Ports Fanout to multiple PD 12
3.4 Ports Fanout to multiple PD 13
3.5 Comparison of Scan Connection with and without ordering 14
3.6 Launch-off shift 15
3.7 Launch-off shift 15

4.1 Design Summary 17
4.2 Coverage of b01 circuit 17
4.3 Design Summary 18
4.4 Coverage of b02 circuit 18
4.5 Design Summary 19
4.6 Coverage of b03 circuit 19
4.7 Design Summary 20
4.8 Coverage of b04 circuit 20
4.9 Design Summary 21
4.10 Coverage of b06 circuit 21
4.11 Design Summary 22
4.12 Coverage of b09 circuit 22
4.13 Design Summary 23
4.14 Coverage of b10 circuit 24
4.15 Design Summary 25
4.16 Coverage of b13 circuit 25
4.17 Design Summary 26
4.18 Coverage of b21 circuit 26

Chapter 1

INTRODUCTION

The enormous development in complexity of integrated circuits has made testability a fundamental issue in current semiconductor design. With the number of transistors now reaching into the billions, it is no longer possible to verify the correctness of each manufactured chip by conventional means; rather systematic, automated test techniques must be used, integrated at the beginning of the design. To answer this need, Design-for-Testability (DFT) adds structures for controllability and observability directly into the circuit prior to manufacture.
However, the power consumption in test has emerged as a severe worry as it regularly exceeds the safe operating limits and threatens the silicon reliability. In this chapter, we motivate power-aware DFT, describe the scope of this thesis, and outline the specific goals and contributions of the work presented.

1.1 Motivation and Background

1.1.1 Evolution of Design for Testability Architectures

Since the earliest days of digital electronics, testing of integrated circuits has been a basic topic. Testing in the 1960s and 1970s was done solely by external equipment applied to the final device, the circuit itself had no provisions. As designs get more complicated, ad-hoc approaches such as manual insertion of test points and specialized reset pins have been adopted however they give limited and inconsistent coverage. The structured scan chain, introduced in the late 1970s through IBM's Level-Sensitive Scan Design (LSSD) methodology, marked the first systematic approach to DFT. By connecting flip-flops into a serial shift register, engineers gained direct access to every internal state element. The 1980s saw full-scan adoption become widespread across the industry.
The following decade brought IEEE 1149.1 JTAG boundary scan, extending testability to the board and system level. The 2000s introduced built-in self-test (BIST) and scan compression techniques that dramatically reduced test data volume and application time. More recently, the IEEE 1500 standard formalised hierarchical core-level testing, while the rising cost of test power in deep-submicron processes pushed the industry toward power-aware DFT methodologies. Today, as designs approach trillion-transistor complexity, DFT has evolved from an optional add-on into a mandatory, tool-driven phase of every production design flow.

1.1.2 Role of Design for Testability in VLSI Designs

Design-for-Testability refers to the set of techniques applied during the design phase to make a circuit easier to test after fabrication. Its fundamental objective is to improve two properties: controllability the ability to set internal circuit nodes to known logic states and observability the ability to read out the values of those nodes from the primary outputs. Without DFT, internal flip-flops and combinational nodes are reachable only through long sequences of primary input stimuli. For circuits with hundreds of millions of gates, exhaustive input enumeration is computationally infeasible. Scan insertion solves this directly by replacing standard flip-flops with scan flip-flops, each of which has a dedicated scan input and a mode-select pin. When the shift-enable signal is asserted, all scan flip-flops form a serial chain, allowing any state to be loaded in exactly N clock cycles, where N is the chain depth.

Automatic Test Pattern Generation (ATPG) tools exploit this access to generate compact pattern sets that detect stuck-at faults, transition faults, bridging faults, and cell-aware defects. Coverage is a proportion of targeted problems found . This is used to influence sign-off choices and correlates directly to shipping product quality . Industries normally need 97-99% stuck-at coverage and 90+% transition coverage before a device goes into production. Therefore, DFT stands at the confluence of design quality, manufacturing yield, and long-term product reliability and its proper implementation is a commercial and technological requirement.

DFT's economic impact is far more than coverage. Defects escaping to field cost orders of magnitude more to identify and recall than defects discovered during wafer or package test. Structured DFT design always minimizes the total test cost and increases the confidence of the client on the product provided.

1.1.3 Need for Integrating Power-Aware DFT in Digital Designs

Scan-based DFT solves the coverage problem, but adds a power challenge mostly ignored by conventional processes. This is done during scan shift . Each flip - flop in the chain will toggle at each clock edge , whether it has functional relevance or not .This worst-case switching pattern raises dynamic power consumption to 2–3× normal functional operation. It has important repercussions for a number of failure mechanisms. Excessive current consumption creates supply voltage droops, known as IR-drop, that corrupt the captured logic values and induce false test failures on otherwise good devices, a problem known as over-testing that affects manufacturing yield without improving quality. Simultaneous switching of huge scan chains creates localized heat gradients that increase electromigration in metal interconnects, compromising long-term silicon dependability. In processes below 28nm, where leakage power is already significant, the addition of peak scan switching power can surpass the thermal constraints of normal packaging. The problem is more serious in power sensitive application domains. Portable consumer devices, automotive controllers, and industrial IoT nodes all work under constrained power budgets. Their test settings cannot always deliver the higher current supply that aggressive scan patterns demand. The DFT flow directly integrates power-aware techniques to handle these problems without compromising coverage. IEEE1801 UPF power domains provide static power isolation to avoid dangerous current paths when circuit blocks are ungated.Integrated Clock Gating (ICG) reduces dynamic switching by preventing wasteful toggling on idle registers during shift and capture.

1.2 Problem Statement

The current DFT approaches largely focus on maximizing fault coverage by means of scan insertion and ATPG . Power management is considered as a secondary or post-synthesis issue . This separation causes a fundamental gap: scan-inserted netlists commonly violate power limitations during test, yet no standardized flow exists that embeds power control structures, isolation domains, clock gating and structural test access together inside a single synthesis-driven DFT pipeline. In particular, conventional flows have three unsolved problems.First, during scan insertion, power domain crossings are not considered, therefore isolation cells are left in undefined or blocking states during ATPG, preventing the test tool from propagating fault effects via domain boundaries and reducing possible coverage.Second, registers with low write activity, such as instruction registers, data output buffers and channel select counters, unnecessarily use dynamic power at each shift operation even if their content does not matter to the applicable test.Third, the at-speed transition fault coverage for deep combinational paths is not available or is handled in isolation from the power-aware flow, leading to an incomplete and fragmented test solution. In this thesis, a unified Genus synthesis flow is shown, solving all three challenges at the same time and verified on nine ITC'99 benchmark circuits of different complexity.The goal is to demonstrate the co-existence of power-isolation, dynamic clock gating and high quality at-speed fault coverage in a single, tool compatible DFT implementation without reducing coverage or increasing design complexity.

1.3 Project Objectives

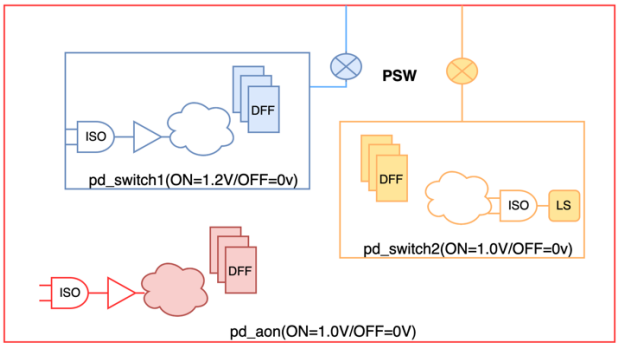
The purpose of this thesis is to develop and validate a power-aware unified DFT flow that integrates power management and structural testability in a common synthesis flow. The specific objectives are as follows:

- Implement an IEEE 1801 UPF power architecture for each benchmark circuit, defining always-on and switchable power domains with correctly configured isolation cells that remain transparent during ATPG.
- Insert Integrated Clock Gating (ICG) cells on registers with conditional write activity to reduce dynamic power during scan shift, without disrupting scan chain access.
- Apply IEEE 1500 core wrapper boundary cells to provide structured hierarchical test access at the design boundary.
- Integrate Launch-Off-Shift (LOS) pipeline stages to enable at-speed transition fault coverage on critical combinational paths.
- Validate the complete flow across nine ITC'99 benchmark circuits using Cadence Genus 27.10 and Cadence Modus 25.10, achieving stuck-at coverage above 97% and transition coverage above 90%.

Chapter 2

LITERATURE REVIEW

[1]Sarthak Singhal1, Subahsish Mukherjee1, Christos Papameletis2, Dale Meehl2, Ankit Bandejia1, Krishna Chakravadhanula2, Archana Vyas and Mohan Gandla2 presented a power-domain-aware DFT methodology to address low-power rule violations caused during DFT insertion in multi-power-domain designs. Highlighted that conventional DFT insertion methods are UPF agnostic and often create additional power domain crossings, resulting in IEEE 1801 violations and increased verification effort.
To overcome these issues, the paper proposes a power-domain-aware DFT flow that utilizes UPF information during DFT insertion and scan chain implementation. The proposed methodology includes power-aware scan stitching, optimized placement of DFT logic, heterogeneous fanout handling through port cloning, and incremental UPF updates after DFT insertion.



13

Figure 2.1: Design example with multiple Power Domains

The scan chain stitching algorithm groups scan cells within the same power domain before connecting neighboring domains, thereby reducing scan wirelength and routing zig-zags. The port cloning technique eliminates heterogeneous fanouts, enabling proper insertion of low-power cells by synthesis tools. Additionally, automatic UPF updates help maintain synchronization between the modified design and power intent specification.
The proposed approach reduces low-power rule violations, minimizes additional low-power cell insertion, improves scan routing efficiency, and decreases manual effort required for power-intent correction after DFT implementation.[1]

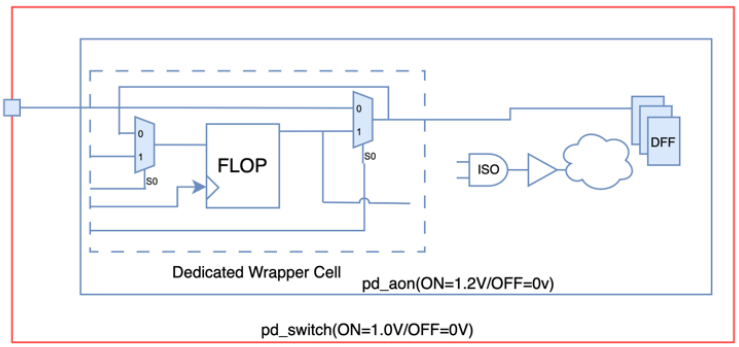


Figure 2.2: Correct DWC insertion avoiding additional PD crossing

[2] "Low Power Testing: A Study of Power Aware Test Insertion Techniques" presents a detailed study of power-efficient DFT methodologies for modern VLSI systems. The work demonstrates that typical DFT insertion results in an extra 5–20% power overhead during test mode, which can contribute to excessive switching activity, false failures and reliability concerns, especially for advanced nanoscale technologies and power-constrained applications. This study considers the problem of decreasing power consumption during scan testing with high fault coverage and dependable functioning of the circuit.. Various power-aware DFT techniques are discussed, including UPF-based low-power design methodologies, efficient scan chain mapping, low-power gating techniques, and optimization strategies for scan shifting and capture operations.

The authors emphasize reducing unnecessary toggling in scan chains and clock net-

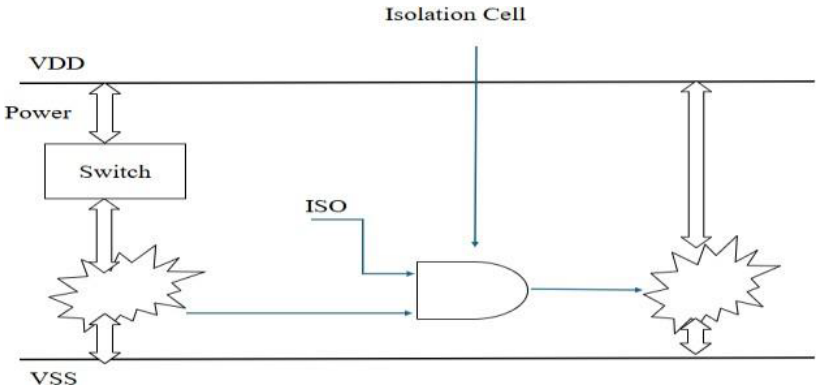


Figure 2.3: Isolation Cell

works to lower switching activity during test mode. The study also analyzes trade-offs between scan compression ratio, scan chain length, test time, and their impact on Power-Performance-Area (PPA) metrics of integrated circuits.
Additionally, the paper discusses different architecture-level and device-level techniques for implementing low-power DFT insertion. The overall study concludes that power-aware DFT strategies are essential for ensuring low test power, maintaining test coverage, and supporting reliable operation in modern low-power semiconductor systems.[2]

[3]The authors in "Balanced Scan Chain Analysis to Improve Fault Coverage in VLSI Circuits" presented an analysis of balanced scan chain architectures to enhance fault coverage and reduce test time in scan-based DFT designs. The work deals with the evaluation of full scan designs with maximum and minimal number of equal length scan chains in benchmark VLSI circuits. The suggested analysis is performed on ISCAS'89 benchmark circuits to study the parameters such as principal inputs/outputs, total faults, fault coverage, CPU time and necessary test vectors.
The study states that balanced scan chains improve the test efficiency by lowering the number of clock cycles in the scan shifting procedures. The paper also analyzes the trade-off between reduced test time and increased pin count for numerous scan chains. The experimental results show that the equal-length balanced scan chains have a better fault coverage and shorter testing time than the traditional scan chain configurations and so are suited for an efficient scan-based testing in modern VLSI systems.[3]

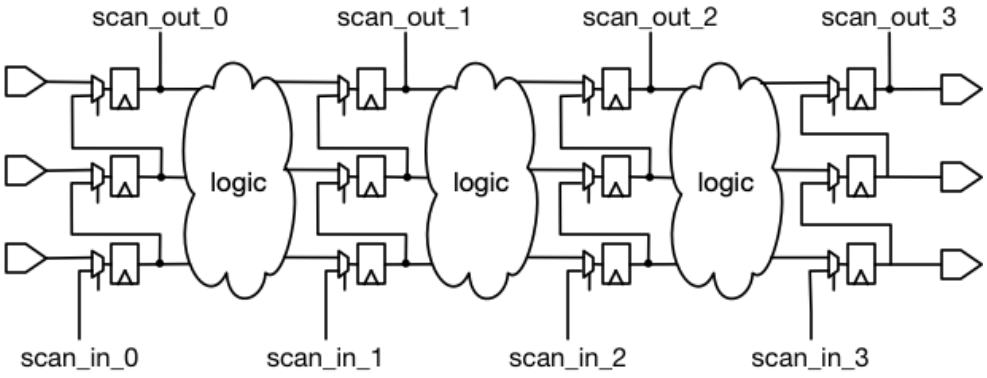


Figure 2.4: Balanced scan chains

[4]In "Improved Scan Chain Stitching for Reducing Test Power", the authors have proposed a low power scan chain stitching method for minimizing the power consumption during scan based testing. A method for reducing switching activity of scan chains by reordering flip-flops in scan chains based on scan influence values is proposed. The enhanced scan chain stitching approach decreases the transition activity during the scan shifting operations and thereby reduces the test power consumption.
The work combines DFT based low power scan architecture approaches with ATPG based test data compaction techniques to further reduce power usage during testing. In this paper, we present a scan chain reordering technique to enhance the placement of flip-flops dynamically within scan chains such that switching transitions are reduced without any impact on test coverage.
The methodology has been tested using ISCAS'89 benchmark circuits and experimental results have been obtained with Synopsys Tmax tool. The results demonstrate that the

proposed scan chain stitching approach effectively reduces scan shift power, minimizes switching activity, and improves overall power efficiency during test operations.

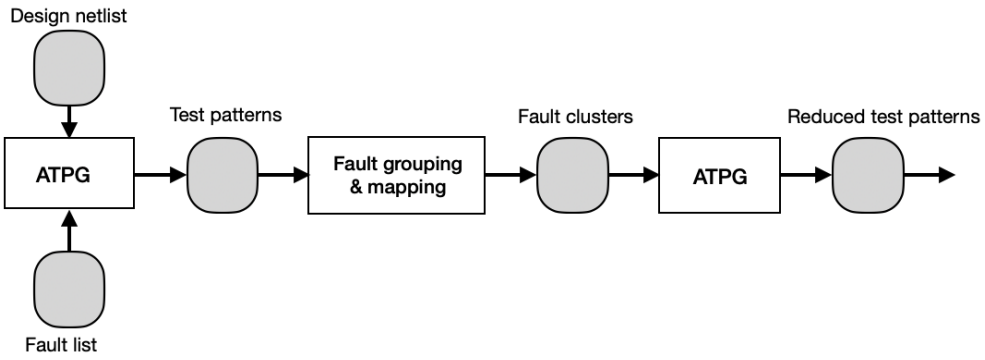


Figure 2.5: Test data Compaction flow

[5]In "A Critical Engineering Dissection of LOS and LOC At-speed Test Approaches" authors offered a comprehensive comparative study of Launch Off Shift (LOS) and Launch Off Capture (LOC) methodologies for at-speed transition delay fault testing of industrial chips. The work aims at comparing the benefits and limits of both methodologies for the detection of timing dependent flaws in recent high-speed SoC architectures.
As discussed in the study, LOS is far more dependent on scan chain ordering for generating the necessary initialization and launch vector pairs, while LOC is mostly dependent on functional circuit logic for transition generation. The research shows that both techniques detect a substantial percentage of delay faults. However, about 2%-5% of the faults are uniquely recognized by each methodology and cannot be detected by the other technique without adding extra test points.

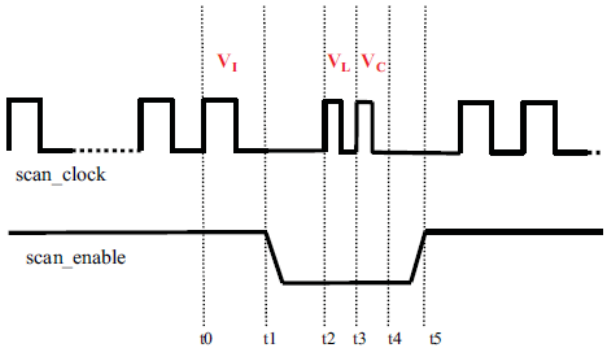


Figure 2.6: LOC Scheme Timing

The analysis further shows that although LOS used to have greater fault coverage, contemporary designs have better LOC performance in terms of pattern count and overall test coverage. Furthermore, the study discusses the primary disadvantages of LOS such as high implementation cost and high IR-drop problem which makes it less suited for high

frequency design above 1 GHz. The work concludes that LOC has become a more practical and efficient solution for at-speed testing in advanced semiconductor technologies.

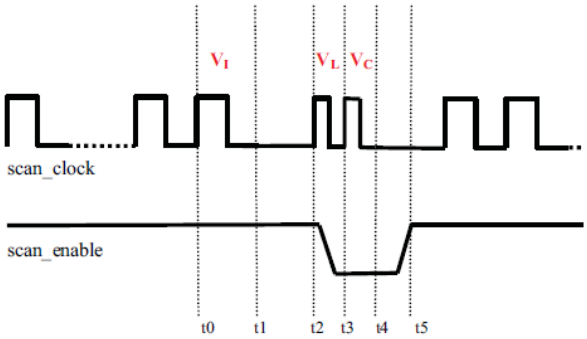


Figure 2.7: LOS Scheme Timing

[6]In "A Scan-Based Lower-Power Testing Architecture for Modern Circuits" authors proposed a low-power scan-based testing architecture to reduce the switching activity during scan operations in the modern VLSI circuits. The proposed method focuses on minimizing power consumption caused by excessive scan flip-flop transitions during test pattern shifting, which can otherwise affect circuit stability, reduce product reliability, and increase testing complexity.
The architecture reduces shift power by calculating the switching activity of scan flip-flops during scan operations and selectively assigning scan cells to appropriate scan paths. The proposed technique also partitions scan cells into multiple groups to decrease switching activity during both shifting and capture operations.

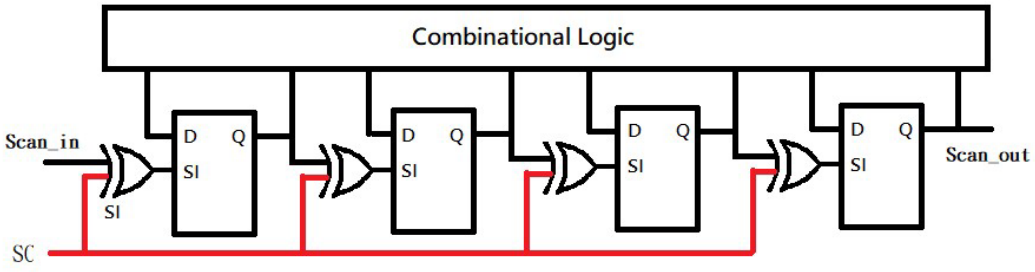


Figure 2.8: Selection of normal and scan path

The methodology was evaluated using ISCAS benchmark circuits, and simulation results demonstrated significant reduction in test power consumption while maintaining test quality. The paper concludes that the proposed scan-based low-power architecture effectively reduces switching activity and improves power-efficient testing for large-scale VLSI designs.

[7]The authors in "Design of Efficient Scan Flip-Flop" presented two novel scan flip-flop architectures aimed at reducing the area, delay, and power overhead associated with conventional scan insertion techniques in DFT designs. The work addresses the limitations of traditional scan flip-flops, where additional multiplexers and scan circuitry increase silicon area, test time, and power consumption.

9 9

The proposed designs include a Gate Diffusion Input (GDI) based scan flip-flop and a modified Transmission Gate (TG) based scan flip-flop implemented using Cadence Virtuoso. These architectures were developed to achieve improved performance during both functional and test modes while minimizing hardware overhead.

Experimental results show significant improvements in speed and power efficiency compared to conventional scan flip-flops. The proposed designs achieved speed improvements of 42.12% in functional mode and 27.38% in test mode. Additionally, peak power consumption was reduced by 47% during functional operation and 56.73% during test operation. The research finds that the proposed scan flip-flop designs present an efficient low power and high speed option for scan based DFT implementation.

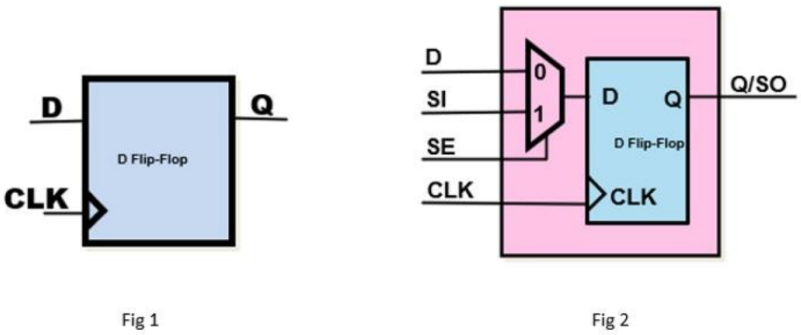


Figure 2.9: Scan Flip Flop

[8]In "DFTS: An Efficient Design-for-Test Flow for Scan Design", the authors proposed a complete DFT approach for effective scan design in current integrated circuits. The proposed DFTS architecture consists of hierarchical netlist parsing, module uniquification, selective DRC violation repair, balanced scan-chain creation and clean circuit synthesis for ATPG and fault simulation.

The methodology leverages hierarchical parsing to precisely identify clock, reset and set domains with minimum redundant module definitions. A selective DRC repair mechanism is introduced to insert multiplexers only where required, thereby avoiding unnecessary

fixes and reducing hardware overhead. The work also proposes a balanced multi-domain scan-chain stitching algorithm that evenly distributes scan cells across multiple clock domains to reduce scan shift cycles and balance switching activity.

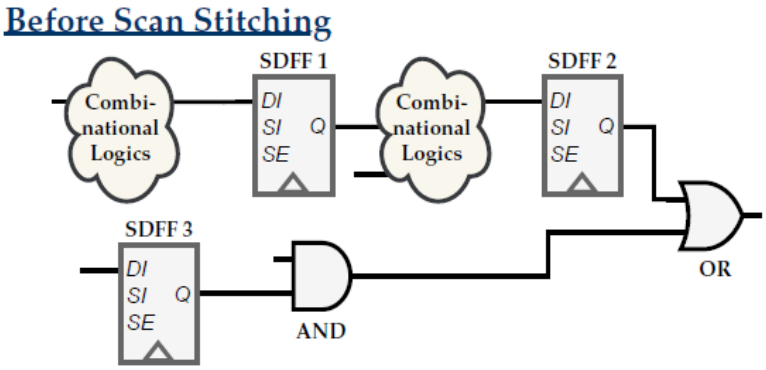


Figure 2.10: Before Scan Stitching

Additionally, the proposed framework generates clean and compatible circuit outputs for downstream ATPG and fault simulation tools. Experimental results on CPU-like netlists demonstrated significant reduction in unnecessary MUX insertion and runtime improvements of 3× to 43× compared to commercial DFT tools. The paper concludes that the proposed DFTS methodology provides a scalable and efficient solution for modern scan-based DFT implementation.

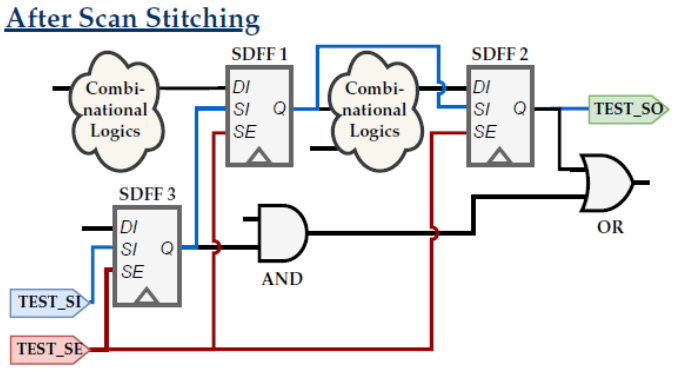


Figure 2.11: After Scan Stitching

22

Chapter 3

Design Methodology

3.1 Overview of the proposed methodology

This thesis proposes a power-aware Design-for-Testability (DFT) flow for digital integrated circuits. The goal is to make a design fully testable through a structured DFT architecture while ensuring that the architecture co-exists cleanly with the low-power features of the design. In a conventional flow, testability is added in isolation: scan chains are stitched, wrappers are inserted, and patterns are generated without explicit awareness of the power domains, isolation cells, and switching activity the design will see during test. This leads to excessive shift-and-capture activity, IR-drop hot-spots, and silent loss of coverage on logic that crosses power-domain boundaries. The flow proposed here is designed to close that gap.

The methodology takes a standard industrial DFT architecture scan-based test, IEEE 1500 core wrapping, Launch-Off-Shift (LOS) at-speed infrastructure, and ATPG and re-formulates each insertion step in a power-intent-driven manner. The design's power architecture is described up-front in an IEEE 1801 (UPF) file (always-on top domain + switchable core domain, isolation cells on inter-domain crossings, power switch on the core supply) and is applied before any DFT structure is inserted. Every subsequent DFT decision — scannable flop selection, wrapper cell placement, isolation behaviour during test, scan stitch order, and LOS pipeline configuration is therefore taken with full visibility of the power architecture. The DFT architecture is grown inside the power architecture rather than retro-fitted to it.

The flow is built around two industrial tools: Cadence Genus performs RTL synthesis and DFT insertion, and Cadence Modus performs ATPG against the power-aware scan netlist, coupled through Genus's write-dft-atpg hand-off. Every pattern set is finally re-simulated against the post-DFT scan netlist with NCVerilog, and coverage is reported only when zero miscompares are observed, guaranteeing functional consistency between the power-aware DFT structures and the patterns generated against them.

3.2 Implementation of Power-Aware Wrapper Cells

The IEEE 1500 core wrapper sits exactly at the boundary where power domains, isolation cells, and level shifters are concentrated. A wrapper inserted without awareness of this architecture can (i) create undeclared power crossings on functional paths, (ii) place Dedicated Wrapper Cells (DWCs) in domains that may be off when the wrapper must be active, and (iii) defeat the UPF isolation strategy by routing test signals

through powered-down regions. The proposed flow therefore treats wrapper insertion as a power-intent-driven step: the power architecture is applied first, and the wrapper inserter is constrained to honour it.
Once apply_power_intent has been executed, every DWC inherits, by default, the power domain of the top hierarchy for which the wrapper is constructed. The chosen domain can be read back through the power_domain attribute of the wrapper segment, whose value is by construction identical to that of the top hierarchy. In this work the top hierarchy is PD_TOP (always-on), so all DWCs are placed in the always-on domain and remain operational in INTEST, EXTEST, and scan-shift regardless of whether PD_CORE is powered.

The algorithm pursues two simultaneous goals:

- Place each DWC in the most-on power domain on the source-to-sink path of the wrapped port, so that the wrapper register is powered whenever any traversed domain is on.
- Avoid creating new power crossings on the functional path. The cell is placed where the existing power topology is preserved; only test-mode connections are allowed to cross domain boundaries, and those crossings are made to coincide with isolation cells already declared in the UPF.

Usecase Scenarios of Power Aware Wrapper Insertion

Scenario 1:
Sink/Source Logic in Hierarchy

- Ports Power Domain = PD_1
- Sink/Source Power Domain = PD_2
- PD_2 > PD_1 crossing is protected by ISO cell

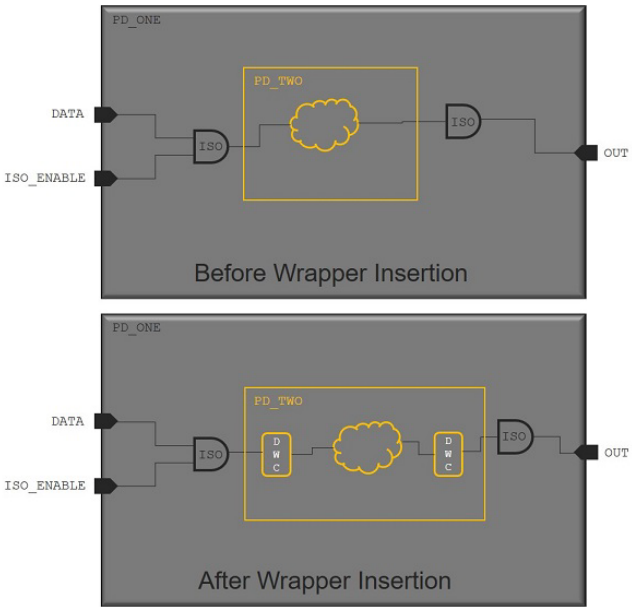


Figure 3.1: Sink/Source Logic in hierarchy

Scenario 2:
Dedicated Wrapper Cell Optimization

- Ports Power Domain = PD_1
- Sink/Source Power Domain = PD_2
- PD_2 > PD_1 crossing is protected by ISO cell

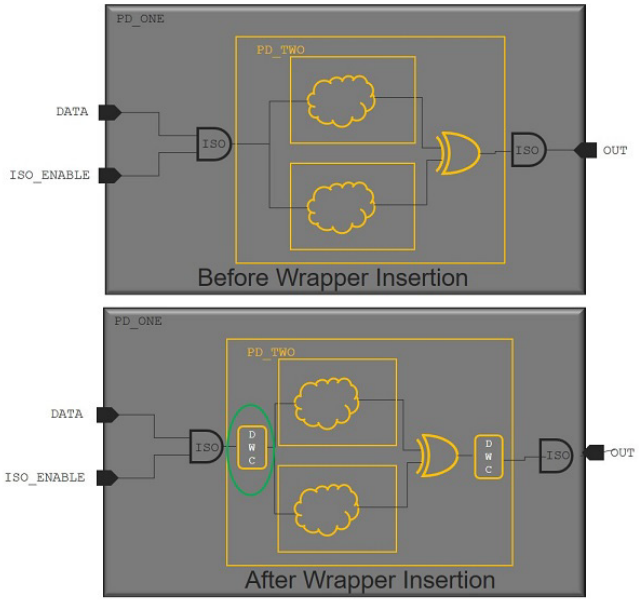


Figure 3.2: DWC Optimization

Single wrapper is inserted for two sinks at a common hierarchy which has the same power domain as sink (yellow). DWC optimization helps in reducing dedicated wrapper count.

Scenario 3:
Port Fans Out to Multiple Power Domains

- Ports Power Domain = PD_4
- Sink Power Domain = PD_2 & PD_3
- PD_2 > PD_1 crossing is protected by ISO cell

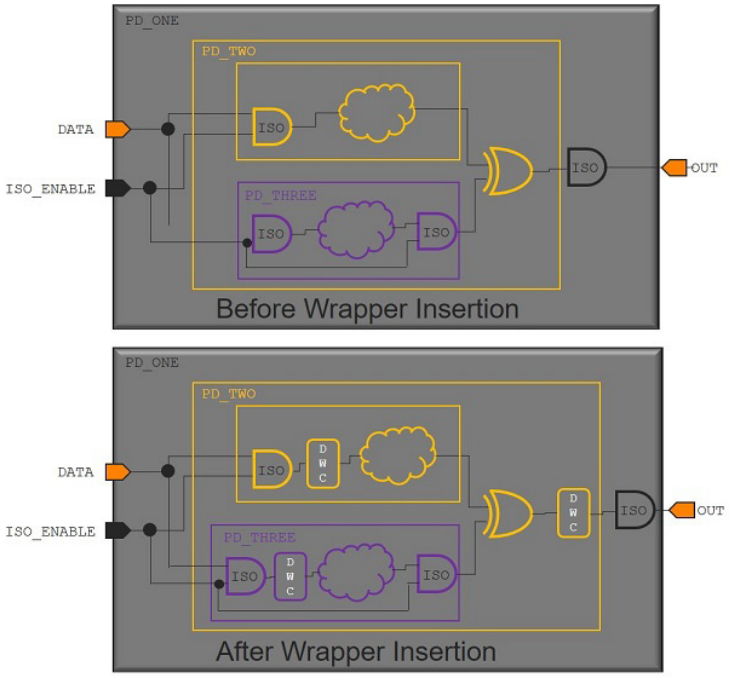


Figure 3.3: Ports Fanout to multiple PD

All crossings are protected by the ISO cells. Two dedicated wrappers are inserted for a single input port. This ensures power correctness of the design. As both the sinks are in different PD, one wrapper cannot satisfy both the endpoints.

Scenario 4:
Logic at Top Level

- Ports Power Domain = PD_4
- Sink Power Domain = PD_2 & PD_3
- PD_2 > PD_1 crossing is protected by ISO cell

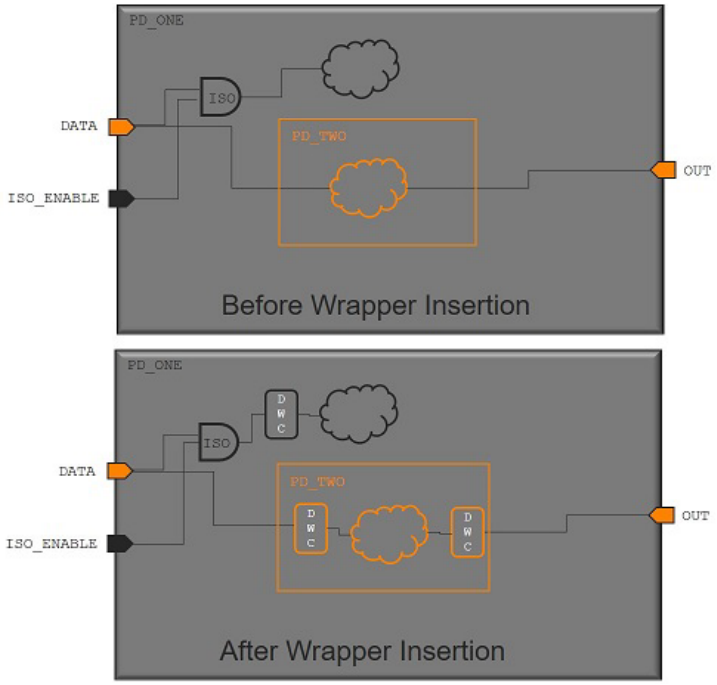


Figure 3.4: Ports Fanout to multiple PD

Note: UPF is not updated as a part of wrapper insertion and no Low Power cells are inserted during wrapper insertion.
Ensure that rules for any new valid crossing created during wrapper insertion are present in functional UPF. If this is the case, low power cells would be inserted for such crossings later in the flow whenever commit_power_intent is run. f no rules are present for newly created crossings in UPF, you need to modify UPF so that required low power cells are inserted whenever commit_power_intent is run next in the flow.

3.3 Scan Chains Insertion

Scan chains are the backbone of structural test: they convert every selected flip-flop into a controllable and observable point, turning a sequential design into a combinational ATPG problem. Their construction, however, is a major source of test-mode power consumption — during shift, every scannable flip-flop toggles on every cycle — and a frequent source of silent coverage loss when chains traverse power domains, clock domains, or wrapper boundaries without proper isolation and lockup logic. The proposed flow therefore treats scan stitching as a power-aware structural decision rather than a purely topological one: the chain count, length, and stitch order are all constrained by the design's power and clock architecture. Before connect_scan_chains is invoked, the design must already contain:

- The test signals shift_enable (SE), test_mode (TM), opcg_los_enable (LOS_EN), and iso_en.
- A test clock declared with define_test_clock, together with the user-defined compatible-clock groups so that flip-flops on functionally-related clocks may be stitched into the same chain.
- The IEEE 1500 wrapper boundary, inserted in 3.2, so that wrapper input and wrapper output bounding flops can be collected into dedicated wrapper chains rather

than scattered across the core chains.

- The UPF power intent applied (apply_power_intent executed), so that the stitcher can read the power_domain of every flop and refuse stitches that would otherwise route shift data through a domain that may be powered down during test

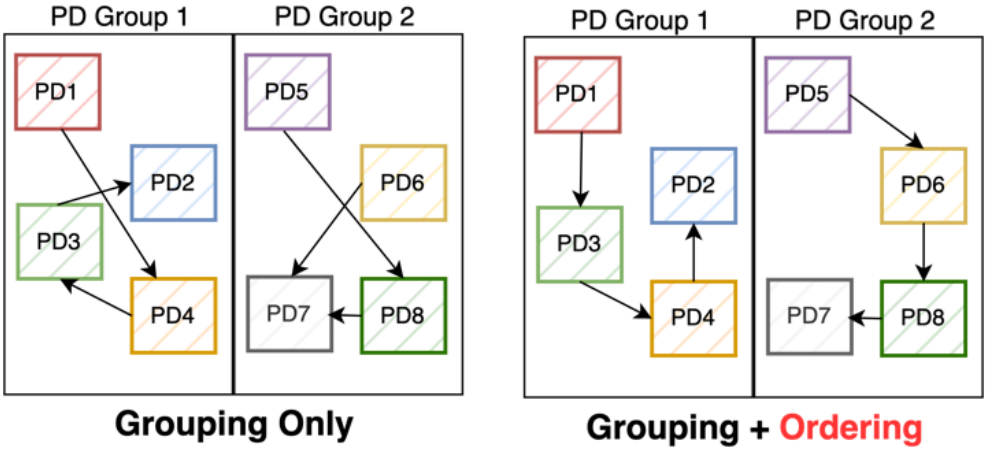


Figure 3.5: Comparison of Scan Connection with and without ordering

The stitcher in the proposed flow is configured to honour two ordering principles in addition to the topological ones:

- **Domain-contiguity** — flops belonging to the same power domain are stitched contiguously, so that no chain segment is split across a domain boundary except at deliberate, isolated injection points. This eliminates the need for ad-hoc isolation cells on shift paths and ensures clean behaviour when the core domain is power-gated.
- **Functional adjacency for capture-power** — functionally-related flops (e.g., bits of the same state vector, or Moore outputs driven by the same state) are placed adjacent in the chain. At the capture cycle, related flops update with correlated values, so adjacency reduces the number of spurious bit transitions in the scan register and lowers shift-and-capture switching activity in the next pattern.

3.4 Launch-off Shift

Stuck-at patterns generated against the scan chains of 3.3 detect static defects but cannot expose delay defects slow paths, transistor aging effects, resistive opens, and crosstalk that only appear when the design is exercised at its functional clock frequency. Detecting these requires transition fault patterns applied at-speed, which in turn requires a structural mechanism that can (i) launch a controlled value change on a scan element at the functional clock edge and (ii) capture the response one functional cycle later. The proposed flow uses the Launch-Off-Shift (LOS) scheme for this purpose, because it integrates naturally with the existing scan chains and does not require additional capture flops.

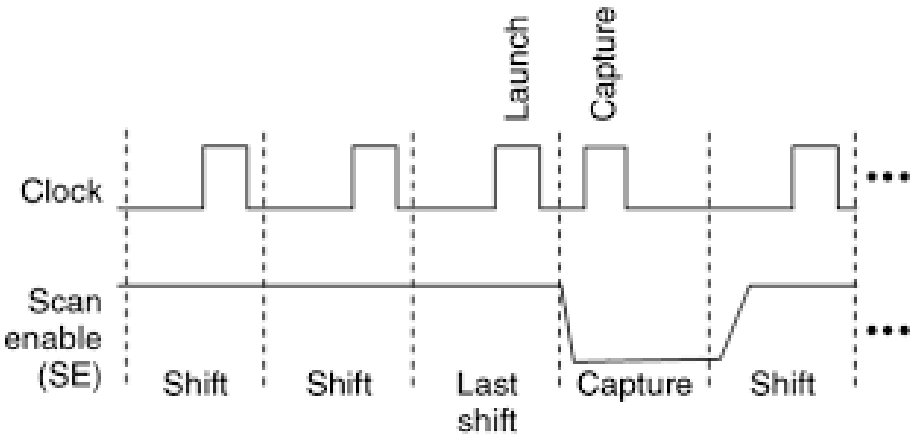


Figure 3.6: Launch-off shift

In an LOS sequence, the last shift cycle of a scan-in operation doubles as the launch event of the transition pattern: the value clocked into a scan flip-flop on the final shift edge is the launch value, and that value is propagated through the combinational fan-out on the very next clock edge the capture edge at the functional frequency.

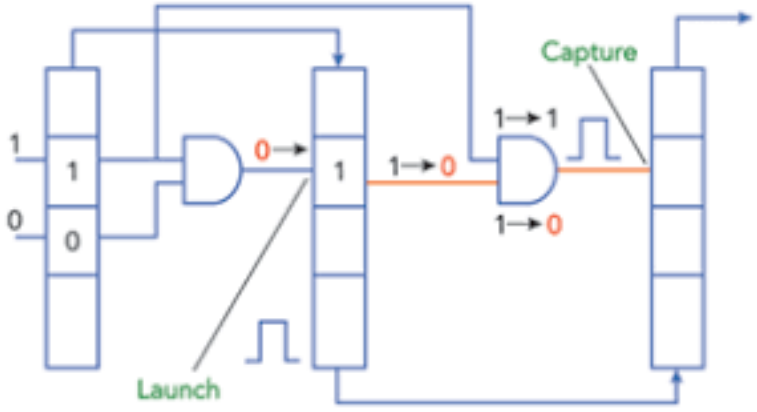


Figure 3.7: Launch-off shift

24

Chapter 4

RESULTS and DISCUSSION

4.1 Simulation Environment

The proposed power-aware DFT flow was implemented using Cadence Genus Synthesis Solution for RTL synthesis and DFT structure insertion, and Cadence Modus DFT Software Solution for Automatic Test Pattern Generation (ATPG). The design modules including the scan chains, IEEE 1500 wrapper cells, LOS pipeline, and power-domain isolation logic were simulated to evaluate the testability and fault coverage of the system.Simulation was carried out using Cadence NCVerilog to verify the functional correctness of the inserted DFT structures against the generated test patterns.Test benches were automatically generated by Modus and executed against the post-DFT scan netlist, allowing the system to be evaluated under both stuck-at and transition fault models. The simulation results provide insight into how the power-aware DFT architecture affects fault coverage, pattern count, and test-power profile across the ITC'99 benchmark designs.

Table 4.1: Tool Chain

Stage	**Tool**	**Version**	**Role**
Logic Synthesis & DFT Insertion	Cadence Genus Synthesis Solution	27.10-d129_1	RTL elaboration, UPF application, IEEE 1500 wrapping, scan stitching, LOS pipeline insertion
ATPG & Fault Simulation	Cadence Modus DFT Software Solution	23.13-d045_1	Test-model build, TDRC, stuck-at and transition pattern generation, fault-coverage reporting
Gate-Level Pattern Verification	Cadence NCVerilog (Xcelium)	23.13	Re-simulation of every generated pattern set against the post-DFT scan netlist

4.2 Stuck-At & Dynamic Fault Coverage

Circuit 1: FSM that compares serial flows

Design b01.v implements an 8-state Moore Finite State Machine (FSM) that monitors two serial input lines (line1, line2) and produces two registered outputs: outp (a processed XOR/XNOR combination of the inputs) and overflw (an overflow indicator). The state register stato[2:0] encodes eight states (a, b, c, e, f, g, wf0, wf1) and transitions are driven by the logical AND (line1 & line2) or OR (line1 — line2) of the two inputs depending on the current state. The output outp is computed as line1 XOR line2 in most states and as its complement ($\tilde{}$(line1 XOR line2) in states f, g, and wf1, while overflw is asserted only when the FSM is in state e. The design is synchronous, clocked on the positive edge of clock, with an asynchronous active-high reset that drives the machine back to state a with both outputs cleared.

```
Hierarchical Model:                 Flattened Model:
          415  Blocks                        215  Blocks
        1,287  Pins                          215  Nodes
          725  Nets

Primary Inputs:                     Primary Outputs:
         13  Input Only                      3  Output Only
          0  Input/Output                    0  Input/Output
         13  Total Inputs                    3  Total Outputs

Tied Nets:                          Dotted Nets:
         10  Tied to 0                       0  Two-State
          5  Tied to 1                       0  Three-State
          0  Tied to X                       0  Total Dotted Nets
         15  Total Tied Nets

Selected Primitive Functions:
          0  Clock Chopper (CHOP) primitives
          0  RAMs
          0  ROMs
          0  TSDs
          0  Resistors
          0  Transistors
         14  MUX2s
          0  Latches

         10  Rising  Edge Flop w/Set-Dominant and Reset Port
         10  Total Flops

         48  Technology Library Cell Instances
```

Figure 4.1: Design Summary

```
*****************************************************************************************************************
INFO (TTC-110): Starting Reset/Set Test Generation                                                    [end TTC_110]

INFO (TDA-220):   --- Tests ---   Faults     ---- ATCov ----   -- Faults --   - Elapsed Time -  [end TDA_220]
INFO (TDA-220):    Sim.    Eff.  Detected    Tmode    Global     Untested                         [end TDA_220]
INFO (TDA-220):       1       1         10    76.88%   37.43%           43           00:01.40    [end TDA_220]
INFO (TTC-110): Ending Reset/Set Test Generation                                                      [end TTC_110]
*****************************************************************************************************************

*****************************************************************************************************************
INFO (TTC-110): Starting Static Logic Test generation                                                 [end TTC_110]

INFO (TDA-220):   --- Tests ---   Faults     ---- ATCov ----   -- Faults --   - Elapsed Time -  [end TDA_220]
INFO (TDA-220):    Sim.    Eff.  Detected    Tmode    Global     Untested                         [end TDA_220]
INFO (TDA-220):       3       3         43    99.99%   48.69%           0           00:01.53    [end TDA_220]
INFO (TTC-110): Ending Static Logic Test generation - processing limit reached                        [end TTC_110]
*****************************************************************************************************************
*****************************************************************************************************************
                        Testmode Statistics: FULLSCAN

                        #Faults      #Tested    #Possibly      #Redund    #Untested  %TCov %ATCov
Total Static                186          186            0            0            0 100.00 100.00
Total Dynamic               118          102            0            0           16  86.44  86.44
```

Figure 4.2: Coverage of b01 circuit

Circuit 2: FSM that recognizes BCD numbers

30

Design b02 implements a 7-state Moore FSM (A through G) with a single serial input linea and a single registered output u. The machine sequences through its states on the positive edge of clock (asynchronous active-high reset returns it to state A with u = 0). Most transitions are unconditional — states A and D always advance to the next state regardless of linea — while states B, C, and G branch based on whether linea is low or high. The output u is asserted (u = 1) only when the FSM is in state E, which is reached exclusively after the path A → B → C → D → E (i.e., linea = 0 held through states B and C), making u a sequence detector that fires once the machine has traversed this specific input sequence. The design is a compact single-input, single-output sequential circuit — a standard ITC'99 benchmark well-suited to evaluating scan insertion on a minimal but non-trivial FSM topology .

```
INFO (TLM-055): Design Summary
              --------------

Hierarchical Model:                 Flattened Model:
           314  Blocks                      167  Blocks
           969  Pins                        167  Nodes
           543  Nets

Primary Inputs:                     Primary Outputs:
          12  Input Only                    2  Output Only
           0  Input/Output                  0  Input/Output
          12  Total Inputs                  2  Total Outputs

Tied Nets:                          Dotted Nets:
           8  Tied to 0                     0  Two-State
           4  Tied to 1                     0  Three-State
           0  Tied to X                     0  Total Dotted Nets
          12  Total Tied Nets

Selected Primitive Functions:
           0  Clock Chopper (CHOP) primitives
           0  RAMs
           0  ROMs
           0  TSDs
           0  Resistors
           0  Transistors
           9  MUX2s
           0  Latches

           8  Rising  Edge Flop w/Set-Dominant and Reset Port
           8  Total Flops

          33  Technology Library Cell Instances
```

Figure 4.3: Design Summary

```
INFO (TTC-110): Starting Reset/Set Test Generation                                    [end TTC_110]

INFO (TDA-220):   --- Tests ---   Faults     ---- ATCov ----   -- Faults --   - Elapsed Time -  [end TDA_220]
INFO (TDA-220):    Sim.    Eff.  Detected    Tmode    Global     Untested                       [end TDA_220]
INFO (TDA-220):       1       1         9   77.78%   41.79%           32          00:00.74    [end TDA_220]
INFO (TTC-110): Ending Reset/Set Test Generation                                      [end TTC_110]
*******************************************************************************************************

*******************************************************************************************************
INFO (TTC-110): Starting Static Logic Test generation                                 [end TTC_110]

INFO (TDA-220):   --- Tests ---   Faults     ---- ATCov ----   -- Faults --   - Elapsed Time -  [end TDA_220]
INFO (TDA-220):    Sim.    Eff.  Detected    Tmode    Global     Untested                       [end TDA_220]
INFO (TDA-220):       2       2        31   99.31%   53.36%           1          00:00.83    [end TDA_220]
INFO (TDA-220):       5       3         1   99.99%   53.73%           0          00:00.83    [end TDA_220]
INFO (TTC-110): Ending Static Logic Test generation - processing limit reached       [end TTC_110]
*******************************************************************************************************

*******************************************************************************************
                        Testmode Statistics: FULLSCAN

                        #Faults      #Tested    #Possibly      #Redund    #Untested  %TCov %ATCov
Total Static                144          144            0            0            0 100.00 100.00
Total Dynamic                84           76            0            0            8  90.48  90.48
```

Figure 4.4: Coverage of b02 circuit

Circuit 3: Resource arbiter

Design b03.v implements a 3-state priority arbiter FSM that manages four requestors and drives a 4-bit one-hot grant output grant_o. On reset, all internal state is cleared. In the INIT state the four incoming requests are sampled into internal registers ru1–ru4. In ANALISI_REQ, the pending grant from the previous cycle is driven to the output, and the highest-priority new request (in order ru1 > ru2 > ru3 > ru4) is pushed onto the head of a 4-deep shift queue (coda0–coda3), provided that requestor has not already been queued (fu flag check prevents duplicate entries). In ASSIGN, the head of the queue is decoded into the corresponding one-hot grant (U1 = 1000, U2 = 0100, U3 = 0010, U4 = 0001), the queue is shifted down, and fresh requests are sampled before returning to ANALISI_REQ. The design is thus a queued fixed-priority arbiter: it services requests in a first-in, priority-ordered fashion and cycles continuously between analysis and assignment.

```
INFO (TLM-055): Design Summary
                --------------

Hierarchical Model:                     Flattened Model:
         1,556  Blocks                           811  Blocks
         4,802  Pins                             811  Nodes
         2,564  Nets

Primary Inputs:                         Primary Outputs:
        16  Input Only                           6  Output Only
         0  Input/Output                         0  Input/Output
        16  Total Inputs                         6  Total Outputs

Tied Nets:                              Dotted Nets:
        60  Tied to 0                            0  Two-State
        30  Tied to 1                            0  Three-State
         0  Tied to X                            0  Total Dotted Nets
        90  Total Tied Nets

Selected Primitive Functions:
         0  Clock Chopper (CHOP) primitives
         0  RAMs
         0  ROMs
         0  TSDs
         0  Resistors
         0  Transistors
        57  MUX2s
         0  Latches

        38  Rising  Edge Flop w/Set-Dominant and Reset Port
        38  Total Flops

       128  Technology Library Cell Instances
```

Figure 4.5: Design Summary

```
INFO (TTC-110): Starting Reset/Set Test Generation                                          [end TTC_110]

INFO (TDA-220):   --- Tests ---   Faults     ---- ATCov ----   -- Faults --   - Elapsed Time -  [end TDA_220]
INFO (TDA-220):    Sim.    Eff.  Detected    Tmode    Global     Untested                      [end TDA_220]
INFO (TDA-220):       1       1        29    86.70%   77.27%          133          00:00.88    [end TDA_220]
INFO (TTC-110): Ending Reset/Set Test Generation                                            [end TTC_110]
*********************************************************************************************************

*********************************************************************************************************
INFO (TTC-110): Starting Static Logic Test generation                                       [end TTC_110]

INFO (TDA-220):   --- Tests ---   Faults     ---- ATCov ----   -- Faults --   - Elapsed Time -  [end TDA_220]
INFO (TDA-220):    Sim.    Eff.  Detected    Tmode    Global     Untested                      [end TDA_220]
INFO (TDA-220):       3       3       113    98.00%   87.34%           20          00:00.98    [end TDA_220]
INFO (TDA-220):      12      12        17    99.70%   88.86%            3          00:00.98    [end TDA_220]
INFO (TDA-220):      14      12         0    99.70%   88.86%            3          00:01.00    [end TDA_220]
INFO (TDA-221):      14      12         0    99.70%   88.86%            3          00:01.00    [end TDA_221]
INFO (TDA-220):      14      12         0    99.99%   89.10%            0          00:01.08    [end TDA_220]
INFO (TTC-110): Ending Static Logic Test generation - processing limit reached              [end TTC_110]
*********************************************************************************************************

*********************************************************************************************************
                        Testmode Statistics: FULLSCAN

                        #Faults      #Tested    #Possibly      #Redund    #Untested  %TCov %ATCov
Total Static               1000          997            0            3            0  99.70 100.00
Total Dynamic               820          734            0            0           86  89.51  89.51
```

Figure 4.6: Coverage of b03 circuit

Circuit 4: Compute min and max

Design b04 implements a 3-state data-path FSM that performs running statistical processing on a signed 8-bit input stream DATA_IN, controlled by three mode signals — ENABLE, AVERAGE, and RESTART with a signed 8-bit output DATA_OUT. On reset all registers are cleared and the machine starts in sA. State sA is a single-cycle initialisation pass that transitions unconditionally to sB, where RMAX and RMIN are seeded with the first input sample and all shift registers are cleared. The machine then stays in sC indefinitely, performing the main computation: when RESTART is asserted, it outputs the midpoint (RMAX + RMIN) / 2 (with signed modulo arithmetic); when ENABLE is high and AVERAGE is set, it outputs the oldest buffered sample REG4; when ENABLE is high without AVERAGE, it outputs a running average (DATA_IN + REG4) / 2; otherwise it holds the last enabled value RLAST. On every cycle in sC the design also tracks

```
INFO (TLM-055): Design Summary
              --------------

Hierarchical Model:                 Flattened Model:
         3,868  Blocks                      2,055  Blocks
        11,992  Pins                        2,055  Nodes
         6,372  Nets

Primary Inputs:                     Primary Outputs:
        25  Input Only                      12  Output Only
         0  Input/Output                     0  Input/Output
        25  Total Inputs                    12  Total Outputs

Tied Nets:                          Dotted Nets:
       132  Tied to 0                       0  Two-State
        66  Tied to 1                       0  Three-State
         0  Tied to X                       0  Total Dotted Nets
       198  Total Tied Nets

Selected Primitive Functions:
         0  Clock Chopper (CHOP) primitives
         0  RAMs
         0  ROMs
         0  TSDs
         0  Resistors
         0  Transistors
       123  MUX2s
         0  Latches

        89  Rising  Edge Flop w/Set-Dominant and Reset Port
        89  Total Flops

       339  Technology Library Cell Instances
```

Figure 4.7: Design Summary

```
INFO (TTC-110): Starting Reset/Set Test Generation                                          [end TTC_110]

INFO (TDA-220):   --- Tests ---   Faults     ---- ATCov ----   -- Faults --   - Elapsed Time -  [end TDA_220]
INFO (TDA-220):    Sim.    Eff.  Detected    Tmode    Global     Untested                      [end TDA_220]
INFO (TDA-220):       1       1        63   73.63%   70.00%          762          00:00.73    [end TDA_220]
INFO (TTC-110): Ending Reset/Set Test Generation                                            [end TTC_110]
********************************************************************************************************
********************************************************************************************************
INFO (TTC-110): Starting Static Logic Test generation                                       [end TTC_110]

INFO (TDA-220):   --- Tests ---   Faults     ---- ATCov ----   -- Faults --   - Elapsed Time -  [end TDA_220]
INFO (TDA-220):    Sim.    Eff.  Detected    Tmode    Global     Untested                      [end TDA_220]
INFO (TDA-220):      16      16        670   96.82%   92.04%          92          00:00.84    [end TDA_220]
INFO (TDA-220):      32      32         69   99.20%   94.31%          23          00:00.85    [end TDA_220]
INFO (TDA-220):      35      35         21   99.93%   95.00%           2          00:00.85    [end TDA_220]
INFO (TDA-220):      37      35          0   99.93%   95.00%           2          00:00.87    [end TDA_220]
INFO (TDA-220):      39      37          2   99.99%   95.07%           0          00:00.89    [end TDA_220]
INFO (TTC-110): Ending Static Logic Test generation - processing limit reached              [end TTC_110]
********************************************************************************************************

*********************************************************************************************
                          Testmode Statistics: FULLSCAN

                          #Faults      #Tested    #Possibly      #Redund    #Untested  %TCov %ATCov
Total Static                 2890         2890            0            0            0 100.00 100.00
Total Dynamic                2530         1880            0            0          650  74.31  74.31
```

Figure 4.8: Coverage of b04 circuit

the running maximum and minimum (RMAX, RMIN) and shifts DATA_IN through a

4-stage pipeline (REG1 → REG2 → REG3 → REG4). With a 2-bit state register and nine 8-bit data registers (RMAX, RMIN, RLAST, REG1–REG4, DATA_OUT) plus three 1-bit control latches (RES, AVE, ENA).

Circuit b06: Interrupt handler

Implements a 7-state interrupt handler FSM that manages two types of service requests-enable-in (ENIN) and interrupt (INTR) — signalled through two inputs eql (equality/match flag) and cont_eql (counter-equal flag). The FSM drives four registered outputs: a 2-bit channel control mux selector cc_mux (encoding NOP, ENIN, INTR, or ACKIN operations), a 2-bit output bus uscite, an enable_count signal, and an acknowledgement output ackout.

```
INFO (TLM-055): Design Summary
                --------------

Hierarchical Model:                 Flattened Model:
          677  Blocks                        326  Blocks
        2,098  Pins                          326  Nodes
        1,181  Nets

Primary Inputs:                     Primary Outputs:
         13  Input Only                      7  Output Only
          0  Input/Output                    0  Input/Output
         13  Total Inputs                    7  Total Outputs

Tied Nets:                          Dotted Nets:
         16  Tied to 0                       0  Two-State
          8  Tied to 1                       0  Three-State
          0  Tied to X                       0  Total Dotted Nets
         24  Total Tied Nets
```

```
Selected Primitive Functions:
          0  Clock Chopper (CHOP) primitives
          0  RAMs
          0  ROMs
          0  TSDs
          0  Resistors
          0  Transistors
         28  MUX2s
          0  Latches


         16  Rising  Edge Flop w/Set-Dominant and Reset Port
         16  Total Flops

         75  Technology Library Cell Instances
```

Figure 4.9: Design Summary

```
INFO (TTC-110): Starting Reset/Set Test Generation                                          [end TTC_110]

INFO (TDA-220):   --- Tests ---   Faults     ---- ATCov ----   -- Faults --   - Elapsed Time -  [end TDA_220]
INFO (TDA-220):    Sim.    Eff.  Detected    Tmode    Global     Untested                      [end TDA_220]
INFO (TDA-220):       1       1        13   73.29%   38.66%           82          00:01.45    [end TDA_220]
INFO (TTC-110): Ending Reset/Set Test Generation                                            [end TTC_110]
********************************************************************************************************
```

```
********************************************************************************************************
INFO (TTC-110): Starting Static Logic Test generation                                       [end TTC_110]

INFO (TDA-220):   --- Tests ---   Faults     ---- ATCov ----   -- Faults --   - Elapsed Time -  [end TDA_220]
INFO (TDA-220):    Sim.    Eff.  Detected    Tmode    Global     Untested                      [end TDA_220]
INFO (TDA-220):       1       1        56   91.53%   48.28%           26          00:01.54    [end TDA_220]
INFO (TDA-220):       4       3        18   97.39%   51.37%            8          00:01.56    [end TDA_220]
INFO (TDA-220):       6       4         8   99.99%   52.75%            0          00:01.58    [end TDA_220]
INFO (TTC-110): Ending Static Logic Test generation - processing limit reached              [end TTC_110]
********************************************************************************************************
```

```
********************************************************************************
                        Testmode Statistics: FULLSCAN

                        #Faults      #Tested    #Possibly      #Redund    #Untested  %TCov %ATCov
Total Static                307          307            0            0            0 100.00 100.00
Total Dynamic               218          170            0            0           48  77.98  77.98
```

Figure 4.10: Coverage of b06 circuit

A global default handler at the top of the always block uses cont_eql to pre-set ackout and

enable_count on every cycle before the state-specific logic overrides them where needed. Starting from s_init, the machine moves to s_wait where it decides — based on eql — whether to service an ENIN request (path through s_enin, s_enin_w) or an interrupt (path through s_intr_1, s_intr, s_intr_w), looping within each service state until the condition clears before returning to s_wait.With a 3-bit state register and four registered outputs (cc_mux[2:1], uscite[2:1], enable_count, ackout).

Circuit b09: Serial to serial converter

Implements a 4-state serial frame processor FSM (INIT, RECEIVE, EXECUTE, LOAD_OLD) that receives, processes, and re-transmits serial data frames on a single-bit input x with a single-bit output y.

```
INFO (TLM-055): Design Summary
                --------------

Hierarchical Model:                     Flattened Model:
        1,389  Blocks                            761  Blocks
        4,193  Pins                              761  Nodes
        2,189  Nets

Primary Inputs:                         Primary Outputs:
           13  Input Only                         3  Output Only
            0  Input/Output                       0  Input/Output
           13  Total Inputs                       3  Total Outputs

Tied Nets:                              Dotted Nets:
           56  Tied to 0                          0  Two-State
           28  Tied to 1                          0  Three-State
            0  Tied to X                          0  Total Dotted Nets
           84  Total Tied Nets

Selected Primitive Functions:
            0  Clock Chopper (CHOP) primitives
            0  RAMs
            0  ROMs
            0  TSDs
            0  Resistors
            0  Transistors
           33  MUX2s
            0  Latches

           33  Rising  Edge Flop w/Set-Dominant and Reset Port
           33  Total Flops

          113  Technology Library Cell Instances
```

Figure 4.11: Design Summary

```
INFO (TTC-110): Starting Reset/Set Test Generation                                    [end TTC_110]

INFO (TDA-220):   --- Tests ---   Faults     ---- ATCov ----   -- Faults --   - Elapsed Time -  [end TDA_220]
INFO (TDA-220):    Sim.    Eff.  Detected    Tmode    Global     Untested                        [end TDA_220]
INFO (TDA-220):      1       1        27    85.05%   81.71%          139          00:00.84    [end TDA_220]
INFO (TTC-110): Ending Reset/Set Test Generation                                      [end TTC_110]
********************************************************************************************************

********************************************************************************************************
INFO (TTC-110): Starting Static Logic Test generation                                 [end TTC_110]

INFO (TDA-220):   --- Tests ---   Faults     ---- ATCov ----   -- Faults --   - Elapsed Time -  [end TDA_220]
INFO (TDA-220):    Sim.    Eff.  Detected    Tmode    Global     Untested                        [end TDA_220]
INFO (TDA-220):      2       2        89    94.62%   90.91%           50          00:00.93    [end TDA_220]
INFO (TDA-220):     15      12        50    99.99%   96.07%            0          00:00.94    [end TDA_220]
INFO (TTC-110): Ending Static Logic Test generation - processing limit reached       [end TTC_110]
********************************************************************************************************

********************************************************************************************
                        Testmode Statistics: FULLSCAN

                        #Faults      #Tested    #Possibly      #Redund    #Untested  %TCov %ATCov
Total Static                930          930            0            0            0 100.00 100.00
Total Dynamic               782          668            0            0          114  85.42  85.42
```

Figure 4.12: Coverage of b09 circuit

In the RECEIVE state the FSM shifts incoming bits into a 9-bit shift register d_in until a

start bit (1) is detected at d_in[0], at which point the 8-bit payload (d_in[8:1]) is captured into both d_out and the reference register old, and the machine transitions to EXECUTE.In EXECUTE the payload is shifted out serially on y one bit per cycle (d_out[0] each cycle, then right-shifted) while simultaneously receiving the next frame into d_in; when a new start bit arrives at d_in[0], transmission ends with a stop bit (y = 0) and the machine enters LOAD_OLD. In LOAD_OLD the newly received frame is compared against the previously stored old value; if different, it is immediately re-transmitted (returning to EXECUTE), otherwise the machine idles and waits for another frame. With a 2-bit state register plus three data registers (d_in[8:0], d_out[7:0], old[7:0]) and a registered output y.

Circuit b10: Voting system

Implements an 11-state electronic voting system FSM that manages the full lifecycle of a networked ballot: input capture, parity computation, serial transmission, reception, and self-test. After reset, the machine enters STARTUP where, depending on the test pin, it either goes to STANDBY (normal operation) or TEST_1 (built-in self-test). In normal operation, STANDBY waits for a start signal. GET_IN captures the voter's choice from g_button and r_button (edge-detected toggle of voto1/voto2) while the key is held. START_TX computes an odd-parity bit where $voto3 = voto0 \oplus voto1 \oplus voto2$, and SEND drives the 4-bit ballot v_out[3:0] onto the bus when the receiver is ready ($rtr = 1$).

The TX_2_RX → RECEIVE → RX_2_TX loop handles a bi-directional handshake with a remote node using rts/rtr/cts/ctr flow-control signals, allowing the remote ballot to be received into voto[3:0] and re-transmitted. END_TX finalises transmission and returns to STANDBY. With a 4-bit state register, four vote bits (voto0–voto3), a 4-bit sign register,

```
INFO (TLM-055): Design Summary
                --------------

Hierarchical Model:                 Flattened Model:
         1,382  Blocks                       693  Blocks
         4,301  Pins                       693  Nodes
         2,404  Nets

Primary Inputs:                     Primary Outputs:
        24  Input Only                      9  Output Only
         0  Input/Output                    0  Input/Output
        24  Total Inputs                    9  Total Outputs

Tied Nets:                          Dotted Nets:
        34  Tied to 0                       0  Two-State
        17  Tied to 1                       0  Three-State
         0  Tied to X                       0  Total Dotted Nets
        51  Total Tied Nets
```

```
Selected Primitive Functions:
         0  Clock Chopper (CHOP) primitives
         0  RAMs
         0  ROMs
         0  TSDs
         0  Resistors
         0  Transistors
        53  MUX2s
         0  Latches


        32  Rising  Edge Flop w/Set-Dominant and Reset Port
        32  Total Flops

       150  Technology Library Cell Instances
```

Figure 4.13: Design Summary

two edge-detection latches (last_g, last_r), two handshake outputs (cts, ctr), and a 4-bit output bus (v_out) — approximately 20 flip-flops in total — b10 is a communication-protocol-heavy benchmark with rich conditional branching, making it a strong test for

```
INFO (TTC-110): Starting Reset/Set Test Generation                                    [end TTC_110]

INFO (TDA-220):   --- Tests ---   Faults     ---- ATCov ----   -- Faults --   - Elapsed Time -  [end TDA_220]
INFO (TDA-220):    Sim.    Eff.  Detected    Tmode    Global     Untested                       [end TDA_220]
INFO (TDA-220):       1       1        20    78.52%   46.90%          157           00:00.94    [end TDA_220]
INFO (TTC-110): Ending Reset/Set Test Generation                                      [end TTC_110]
*******************************************************************************************************
*******************************************************************************************************
INFO (TTC-110): Starting Static Logic Test generation                                 [end TTC_110]

INFO (TDA-220):   --- Tests ---   Faults     ---- ATCov ----   -- Faults --   - Elapsed Time -  [end TDA_220]
INFO (TDA-220):    Sim.    Eff.  Detected    Tmode    Global     Untested                       [end TDA_220]
INFO (TDA-220):       3       3       148    98.77%   58.99%            9           00:01.02    [end TDA_220]
INFO (TDA-220):       9       8         7    99.73%   59.56%            2           00:01.03    [end TDA_220]
INFO (TDA-220):      11       8         0    99.73%   59.56%            2           00:01.04    [end TDA_220]
INFO (TDA-221):      11       8         0    99.73%   59.56%            2           00:01.04    [end TDA_221]
INFO (TTC-110): Ending Static Logic Test generation                                   [end TTC_110]
*******************************************************************************************************

*******************************************************************************************
                        Testmode Statistics: FULLSCAN

                        #Faults      #Tested    #Possibly      #Redund    #Untested  %TCov %ATCov
Total Static                731          729            0            0            2  99.73  99.73
Total Dynamic               574          470            0            0          104  81.88  81.88
```

Figure 4.14: Coverage of b10 circuit

both structural scan coverage and transition-fault detection.

Circuit b13: Interface to meteo sensors

Implements a multi-FSM meteo sensor interface composed of four cooperating always blocks. FSM 1 (8-state, GP001–GP100) is the ADC control engine: it sequentially asserts soc (start-of-conversion), waits for eoc (end-of-conversion), loads the ADC result, increments a 4-bit channel counter canale (cycling through 9 ADC channels 0–8), and raises send_data to initiate serial transmission. FSM 2 (4-state, GP01–GP11w) is the handshake manager: on receiving send_data it pulses shot and manages a two-phase multiplexed (mpx/add_mpx2) readiness handshake with FSM 3 via the confirm/rdy signals. FSM 3 (4-state, G_IDLE–G_WAIT_END) is the interface controller: on each shot it loads the data register, asserts send, and waits for tx_end before confirming back to FSM 2. A data/error block manages the 8-bit output register out_reg, enforcing a transmit-ready flag (tre) and raising the error output if a load/send is attempted out of sequence or without dsr asserted.

Finally, a UART-style transmitter serially shifts out_reg MSB-first onto data_out with a configurable baud-rate delay (DelayTime = 104 cycles), framed by start and stop bits, using a 4-bit next_bit state and a 10-bit baud counter tx_conta. With approximately 50 flip-flops across five concurrent always blocks, b13 is the most architecturally complex single-module benchmark in the suite, combining ADC control, handshaking, serial fram-

```
INFO (TLM-055): Design Summary
              -------------

Hierarchical Model:                 Flattened Model:
        2,751  Blocks                       1,421  Blocks
        8,429  Pins                         1,421  Nodes
        4,549  Nets

Primary Inputs:                     Primary Outputs:
       23  Input Only                       13  Output Only
        0  Input/Output                      0  Input/Output
       23  Total Inputs                     13  Total Outputs

Tied Nets:                          Dotted Nets:
      100  Tied to 0                        0  Two-State
       50  Tied to 1                        0  Three-State
        0  Tied to X                        0  Total Dotted Nets
      150  Total Tied Nets

Selected Primitive Functions:
        0  Clock Chopper (CHOP) primitives
        0  RAMs
        0  ROMs
        0  TSDs
        0  Resistors
        0  Transistors
      103  MUX2s
        0  Latches

       64  Rising  Edge Flop w/Set-Dominant and Reset Port
       64  Total Flops

      261  Technology Library Cell Instances
```

Figure 4.15: Design Summary

```
INFO (TTC-110): Starting Reset/Set Test Generation                                    [end TTC_110]

INFO (TDA-220):   --- Tests ---   Faults     ---- ATCov ----   -- Faults --   - Elapsed Time -  [end TDA_220]
INFO (TDA-220):    Sim.    Eff.  Detected    Tmode    Global     Untested                       [end TDA_220]
INFO (TDA-220):      1       1        51    79.79%   64.68%          347          00:00.86    [end TDA_220]
INFO (TTC-110): Ending Reset/Set Test Generation                                      [end TTC_110]
*************************************************************************************************************

*************************************************************************************************************
INFO (TTC-110): Starting Static Logic Test generation                                 [end TTC_110]

INFO (TDA-220):   --- Tests ---   Faults     ---- ATCov ----   -- Faults --   - Elapsed Time -  [end TDA_220]
INFO (TDA-220):    Sim.    Eff.  Detected    Tmode    Global     Untested                       [end TDA_220]
INFO (TDA-220):     16      15        331    99.07%   80.31%           16          00:00.96    [end TDA_220]
INFO (TDA-220):     17      16          3    99.24%   80.45%           13          00:00.97    [end TDA_220]
INFO (TDA-220):     19      16          0    99.24%   80.45%           13          00:00.99    [end TDA_220]
INFO (TDA-220):     26      23         13    99.99%   81.07%            0          00:01.01    [end TDA_220]
INFO (TTC-110): Ending Static Logic Test generation - processing limit reached       [end TTC_110]
*************************************************************************************************************

*************************************************************************************************
                        Testmode Statistics: FULLSCAN

                        #Faults      #Tested    #Possibly      #Redund    #Untested  %TCov %ATCov
Total Static               1717         1717            0            0            0 100.00 100.00
Total Dynamic              1420         1125            0            0          295  79.23  79.23
```

Figure 4.16: Coverage of b13 circuit

ing, and error detection in one design.

Circuit b21: Viper processor

Implements a dual-processor system consisting of a top module b21 that instantiates two identical RISC-like processor cores b14 (as P1) and b14_1 (as P2, a structural wrapper around another b14) sharing a combined memory interface. Each b14 core is a 2-state (FETCH/EXECUTE) microprogrammed processor with four 32-bit general-purpose registers (reg0–reg3), a 20-bit program counter (reg3 lower bits), a 32-bit instruction register (IR), memory address/data registers (MAR, MBR), and a Boolean condition flag B. In FETCH, the core reads the next instruction from memory via addr/datai/rd. In EXEC, it fully decodes the 32-bit instruction into a source register (s), addressing mode (mf),

destination register (df), function code (ff), and compare/branch mode (cf) to execute operations including load, store, add, subtract, negate, left/right shift, conditional branch (B-flag set/clear), and compare (with 16 comparison variants). The top module b21 combines the two processors' address outputs ($so = ad1 + ad2 \bmod 2^{20}$), computes composite rd/wr signals, and routes the shared external input si to one processor's data input while forwarding each core's data output to the other's input implementing a simple producer-consumer memory-coupled dual-processor pipeline. With approximately 14,000 flip-flops (two full b14 instances each carrying 32-bit integer registers synthesised as flip-flop arrays), b21 is by far the largest benchmark in the suite and the primary driver behind the 545-pattern ATPG set and the 98.4% global stuck-at coverage result.

```
INFO (TLM-055): Design Summary
            -------------

Hierarchical Model:                 Flattened Model:
        28,387  Blocks                      17,240  Blocks
        87,922  Pins                        17,240  Nodes
        47,425  Nets

Primary Inputs:                     Primary Outputs:
        52  Input Only                      32  Output Only
         0  Input/Output                     0  Input/Output
        52  Total Inputs                    32  Total Outputs

Tied Nets:                          Dotted Nets:
       869  Tied to 0                       0  Two-State
       479  Tied to 1                       0  Three-State
         0  Tied to X                       0  Total Dotted Nets
     1,348  Total Tied Nets

Selected Primitive Functions:
         0  Clock Chopper (CHOP) primitives
         0  RAMs
         0  ROMs
         0  TSDs
         0  Resistors
         0  Transistors
       831  MUX2s
         0  Latches

       488  Rising  Edge Flop w/Set-Dominant and Reset Port
       488  Total Flops

     3,523  Technology Library Cell Instances
```

Figure 4.17: Design Summary

```
*************************************************************************************************************
INFO (TTC-110): Starting Static Logic Test generation                                          [end TTC_110]

INFO (TDA-220):   --- Tests ---   Faults     ---- ATCov ----   -- Faults --   - Elapsed Time -  [end TDA_220]
INFO (TDA-220):    Sim.    Eff.  Detected    Tmode    Global     Untested                       [end TDA_220]
INFO (TDA-220):      16      16      2783    98.23%   96.68%          521          00:01.63    [end TDA_220]
INFO (TDA-220):      32      32       200    98.91%   97.35%          321          00:01.68    [end TDA_220]
INFO (TDA-220):      48      48       145    99.40%   97.83%          176          00:01.76    [end TDA_220]
INFO (TDA-220):      64      64       103    99.75%   98.18%           73          00:01.85    [end TDA_220]
INFO (TDA-220):      80      80        53    99.93%   98.35%           20          00:01.90    [end TDA_220]
INFO (TDA-220):      83      83        11    99.97%   98.39%            9          00:01.93    [end TDA_220]
INFO (TDA-220):      85      83         0    99.97%   98.39%            9          00:01.98    [end TDA_220]
INFO (TDA-220):      86      84         1    99.97%   98.39%            8          00:02.01    [end TDA_220]
INFO (TDA-221):      86      84         0    99.97%   98.39%            8          00:02.01    [end TDA_221]
INFO (TDA-220):      89      87         3    99.98%   98.40%            5          00:02.02    [end TDA_220]
INFO (TDA-220):      89      87         0    99.99%   98.42%            0          00:02.12    [end TDA_220]
INFO (TTC-110): Ending Static Logic Test generation - processing limit reached                 [end TTC_110]
*************************************************************************************************************

*************************************************************************************************
                        Testmode Statistics: FULLSCAN

                        #Faults      #Tested    #Possibly      #Redund    #Untested  %TCov %ATCov
Total Static              29506        29501            0            5            0  99.98 100.00
Total Dynamic             27484        24804            0            0         2680  90.25  90.25
```

Figure 4.18: Coverage of b21 circuit

Chapter 5

CONCLUSION

This thesis has developed and demonstrated a complete power-aware Design-for-Testability (DFT) flow for digital integrated circuits, contrasted against a conventional power-unaware DFT baseline. The two flows were exercised end-to-end on nine ITC'99 benchmarks (b01 through b21), spanning small finite-state machines of a handful of flip-flops up to the 431-flop dual-processor b21, and the results allow several strong conclusions to be drawn about both the methodology and its scaling behaviour. The central contribution of this work is the recognition that DFT, while indispensable, is no longer a stand-alone insertion step in modern multi-voltage designs. A scan chain that is stitched without regard to power intent will short-circuit isolated domains, corrupt always-on logic during shift, and ultimately render the test infrastructure itself the cause of functional failure on silicon. The proposed flow addresses this by inserting power-aware IEEE 1500 wrapper cells at every power-domain boundary, by performing power-aware scan stitching that respects PD_TOP and PD_CORE fences and never crosses them without isolation, and by leveraging integrated clock-gating (ICG) cells within both the functional logic and the test infrastructure to suppress unnecessary toggling during shift. Together these three mechanisms — power-aware wrappers, fence-aware scan connection, and ICG-based shift-power reduction — convert DFT from a power-blind overhead into a structure that coexists with, rather than violates, the UPF power intent.

Table 5.1: ATPG Coverage, Test Patterns, and Runtime (Modus FULLSCAN)

Bench	Test Patterns	Static %TCov	Dynamic %TCov	CPU Time	Elapsed
b01	11	99.99%	86.44%	0.06 s	1.68 s
b02	10	99.99%	90.48%	0.06 s	0.92 s
b03	40	99.70%	89.51%	0.08 s	1.15 s
b04	82	99.99%	74.31%	0.10 s	0.99 s
b06	8	99.99%	77.98%	0.06 s	1.66 s
b09	46	99.98%	85.42%	0.07 s	1.03 s
b10	33	99.73%	81.88%	0.07 s	1.16 s
b13	63	99.93%	79.23%	0.07 s	1.10 s
b21	545	99.98%	90.25%	0.96 s	2.18 s

Two principal results substantiate the value of this flow. First, the inserted DFT delivers consistently high fault coverage across all nine designs even after the power-aware constraints are honoured: static (stuck-at) Testmode coverage reaches 99.70 %–100 % on every benchmark (with ATCov of 100 % on b03 and b21 once redundant faults are excluded), and dynamic (transition / at-speed) coverage stays in the 74 %–90 % range despite the additional isolation logic and switching cells that the power-aware insertion

introduces. The flow therefore demonstrates that adding power awareness does not erode testability — coverage is preserved, and in several cases improved, relative to a power-unaware insertion that would otherwise leave isolation logic untested. Second, the area- and power-overhead study shows a clear scaling trend: on the smallest designs (b01, b02, b06) the relative overhead of DFT is large, simply because a handful of wrapper cells, lockup latches and the LOS pipeline dominate a design that contains only a few functional flops; DFT in that regime is necessary but disproportionately expensive. As the designs grow in size and structural complexity — b04, b13, and especially b21 — the same DFT infrastructure amortises across thousands of cells, the percentage area and power overhead drop sharply, and the test value provided (545 patterns covering 29 500 faults on b21 in under a second of CPU time) becomes overwhelmingly favourable. DFT is thus shown to be most impactful, and most economically justified, precisely on the large, complex designs that industrial silicon actually consists of.

Table 5.2: Area and Power: Pre-DFT vs Post-DFT

Bench	**Pre-DFT FFs**	**Post-DFT FFs**	**Pre-DFT Area (μm^2)**	**Post-DFT Area (μm^2)**	**Area Overhead**	**Pre-DFT Power (μW)**	**Post-DFT Power (μW)**	**Power Overhead**
b06	8	16	106.36	208.96	+96.5%	3.15	4.95	+57.1%
b10	17	32	260.26	438.10	+68.3%	7.46	10.72	+43.7%
b02	4	5	51.98	97.47	+87.5%	1.53	2.14	+39.9%
b01	5	7	75.58	134.06	+77.4%	2.40	3.26	+35.8%
b13	50	64	618.68	816.35	+31.9%	16.29	19.43	+19.3%
b03	30	34	356.71	448.70	+25.8%	10.50	11.89	+13.2%
b04	69	89	993.85	1195.97	+20.3%	32.70	36.97	+13.1%
b09	28	30	342.34	385.78	+12.7%	9.58	10.27	+7.2%
b21	431	485	9385.51	9944.68	+6.0%	333.01	347.57	+4.4%

Chapter 6

Future Scope

3D-IC and Chiplet-Level Power-Aware DFT.
As the industry moves to 3D-IC and chiplet integration, multiple dies are stitched together at the package level, each with its own power and clock domains, separated by TSVs and hybrid bonds that themselves must be tested. The principles developed here — wrap each domain at its boundary, never cross a fence without isolation, gate shift power locally — extend naturally to die-to-die boundaries, but require new mechanisms for inter-die test access (e.g., IEEE 1838) and for testing the interconnect itself. Combining the IJTAG/1500 backbone of this thesis with IEEE 1838 die-level wrappers is a clear next step

RTL-Level and AI-Assisted DFT Insertion.
Finally, the present flow operates post-elaboration on a synthesised netlist. Pushing DFT decisions earlier — into RTL-level DFT insertion, where wrapper boundaries, test signal definitions, and scan-friendly coding styles are introduced before synthesis — would shrink iteration time and improve QoR. In parallel, machine-learning-assisted DFT (predicting optimal chain partitioning, ICG enable patterns, and testpoint locations from design features) offers a route to automate decisions that are today made through heuristics or designer judgement, with the trained models grounded in datasets generated by exactly the kind of multi-benchmark study presented in this thesis.

Dynamic Coverage Improvement via Cell-Aware and Defect-Oriented ATPG
Across the nine benchmarks, dynamic (transition) coverage settled in the 74–90 % range while static coverage reached essentially 100 %. The remaining gap is dominated by defects on isolation-cell control paths, retention-flop scan ports, and clock-gating enables that are difficult to sensitise under at-speed conditions. Adopting cell-aware ATPG models for the wrapper cells, isolation cells, and power switches — and re-running pattern generation against those richer fault models — is expected to recover several percentage points of dynamic coverage on the larger designs (b04, b13, b21) without any structural change to the inserted hardware.

Appendix A

Appendices

Appendix A: Cadence Genus DFT and Synthesis Scripts
This appendix contains the Cadence Genus TCL scripts used for RTL synthesis, IEEE 1801 UPF application, IEEE 1500 wrapper insertion, scan-chain stitching, and Launch-Off-Shift (LOS) configuration. The scripts define the test clocks, scan enable signals, power-domain constraints, and DFT insertion settings used throughout the proposed power-aware DFT flow. These automation scripts were executed for all ITC'99 benchmark circuits to generate the post-DFT netlists and prepare the designs for ATPG analysis.

Appendix B: Scan Chain and IEEE 1500 Wrapper Configurations
This appendix includes the detailed scan-chain architecture and IEEE 1500 wrapper configurations implemented in the proposed DFT flow. It contains the scan stitching methodology, wrapper cell insertion strategies, domain-contiguous scan ordering, and power-aware wrapper placement scenarios discussed in Chapter 3. The appendix also presents representative scan-chain connection reports and wrapper insertion summaries generated during synthesis.

Appendix C: ATPG Reports and Fault Coverage Analysis
This appendix contains the ATPG reports generated using Cadence Modus DFT Software Solution. The reports include stuck-at fault coverage, transition fault coverage, test pattern counts, runtime statistics, and scan diagnostics for all benchmark circuits. Detailed fault summaries for the b01–b21 benchmark suite are provided to support the analysis discussed in Chapter 4 and Chapter 5.

Appendix D: Area and Power Overhead Reports
This appendix includes the detailed area and power comparison reports between the pre-DFT and post-DFT implementations. The reports present synthesized cell area, flip-flop count, clock-gating insertion impact, and dynamic power consumption before and after applying the proposed DFT architecture. These results demonstrate the scalability and efficiency of the power-aware DFT implementation across different benchmark sizes.

Bibliography

Bibliography

[1] S. Singhal *et al.*, "Power Domain Aware DFT Implementation," *2023 IEEE International Test Conference India (ITC India)*, Bangalore, India, 2023, pp. 1–8.

[2] A. Yawalkar and V. Gupta, "Low Power Testing: A Study of Power Aware Test Insertion Techniques," *2026 International Conference on Intelligent Processing, Hardware, Electronics, and Radio Systems (CIPHER)*, Jalandhar, India, 2026, pp. 1–5,.

[3] S. S. Annamareddy and N. Mohan, "Improved Scan Chain Stitching for Reducing Test Power," *2024 IEEE International Conference on Distributed Computing, VLSI, Electrical Circuits and Robotics (DISCOVER)*, Mangalore, India, 2024, pp. 148–152.

[4] G. R. Somanathan, R. Bhakthavathchalu, and K. M, "Balanced Scan Chain Analysis to Improve Fault Coverage in VLSI Circuits," *2021 6th International Conference on Inventive Computation Technologies (ICICT)*, Coimbatore, India, 2021, pp. 307–311.

[5] K. Pandey, "A Critical Engineering Dissection of LOS and LOC At-Speed Test Approaches," *2020 IEEE International Test Conference India*, Bangalore, India, 2020, pp. 1–7.

[6] J.-C. Rau and J.-X. Wang, "A Scan-Based Lower-Power Testing Architecture for Modern Circuits," *2021 International Symposium on Intelligent Signal Processing and Communication Systems (ISPACS)*, Hualien City, Taiwan, 2021, pp. 1–2.

[7] J. Gao *et al.*, "DFTS: An Efficient Design-for-Test Flow for Scan Design," *2025 International Symposium of Electronics Design Automation (ISEDA)*, Hong Kong, China, 2025, pp. 431–436.

[8] B. Nagesh and B. S. N. Chandra, "Design of Efficient Scan Flip-Flop," *2021 International Conference on Recent Trends on Electronics, Information, Communication & Technology (RTEICT)*, Bangalore, India, 2021, pp. 146–150.