2025 2nd International Conference on Recent Trends in Electrical, Electronics and Computing Technologies (ICRTEECT) | 979-8-3315-8172-5/25/$31.00 ©2025 IEEE | DOI: 10.1109/ICRTEECT67512.2025.11448754

Power-Aware Streaming Scan Network Implementation for Improved Test Coverage and IR Drop Mitigation

Praveenkumar Jayakumar
Department of ECE
Bharath Institute of Higher Education and Research Chennai, India
praveencando91@gmail.com

Sudhagar G
Department of ECE
Bharath Institute of Higher Education and Research Chennai, India
sudhagarambur@gmail.com

Abstract: As integrated circuits (ICs) continue to scale in complexity and density, ensuring efficient and reliable testing becomes increasingly critical. This dissertation presents a comprehensive approach to enhancing scan-based testing methodologies through the implementation of the Streaming Scan Network (SSN). The proposed work focuses on improving test coverage and efficiency while addressing key challenges such as IR drop during the capture phase. To mitigate these issues, pipeline stages are strategically introduced at the On-Chip Controller (OCC) level, thereby reducing power-related anomalies and improving signal integrity during high-speed testing.

The study also explores scalable SSN architectures that support design-for-testability (DFT) in large and hierarchical designs. By leveraging industry-standard tools such as Tessent Shell, the SSN framework is implemented, simulated, and optimized to ensure robust functionality. The methodology includes detailed verification and debugging processes to identify and resolve scan chain anomalies, ensuring accurate fault detection and diagnosis.

Through architectural enhancements and tool-driven optimization, this work demonstrates significant improvements in test reliability, power efficiency, and scalability. The proposed SSN-based testing framework offers a practical and effective solution for modern IC verification, contributing to faster design closure and higher manufacturing yield.

Keywords: SSN, Pipelines, OCC, EDT, IR Drop

I. INTRODUCTION

As ICs grow in complexity, efficient testing is essential for reliability. Scan-based testing is widely used but faces challenges in power, test time, and signal integrity. The Streaming Scan Network (SSN) addresses these issues by enabling parallel and hierarchical testing, reducing test time and improving coverage (see Fig. 2). However, IR drop during Transition Delay Fault (TDF) testing remains a concern. To mitigate this, pipeline delays are introduced at the On-Chip Clock Controller (OCC) level, staggering capture pulses across clock domains to reduce peak current (see Fig. 1). Implemented using Tessent Shell, this scalable, power-aware framework enhances test reliability and design-for-testability for advanced VLSI designs.

II. OBJECTIVE

The primary goal of this project is to develop a robust scan-based testing approach using pipeline delays in OCC to reduce IR drop during the capture phase of testing. The goal is to improve power integrity, test accuracy, and scalability of the Streaming Scan Network (SSN) architecture for complex SoC designs. (see Fig. 1 for OCC architecture)

• Enhance test performance using SSN to support parallel and concurrent scan operations. (see Fig. 2 for SSN architecture).

• Reduce IR drop by introducing programmable delays at the OCC level, thereby spreading switching activity during TDF capture.

• Develop a scalable architecture that supports hierarchical scan testing and adapts to different IC sizes and clock domains.

• Minimize test data volume by leveraging compression techniques provided by Embedded Deterministic Test (EDT) logic. (see Fig. 3 for EDT architecture).

• Improve fault diagnosis and scan debugging through automation scripts and real-time pattern verification using tools like Tessent Shell

III. LITERATURE SURVEY

The following literature provides a comprehensive overview of recent advancements in Streaming Scan Network (SSN) architectures and their applications in Design-for-Test (DFT) for complex System-on-Chip (SoC) designs:

J. Gaudet, J. Burchard, M. Kampmann, J.-F. Côté, T. Callahan, and H. H. Chai, "High-bandwidth IJTAG over SSN," in Proc. IEEE Int. Test Conf. (ITC), Nov. 3–8, 2024. [1]

This paper addresses the growing test challenges in modern SoCs, including limited I/O availability, scan path routing constraints, and increased test data volume. It introduces an enhanced DFT solution that integrates IEEE 1687 (IJTAG) over SSN, enabling high-speed serial communication for embedded instruments. The authors detail the integration methodology, timing and design implications, and verification results from Intel's multi-chiplet CPU project, along with insights into die area impact and test cost reduction.

H. Iwata, M. AbdAlwahab, R. Press, and O. Sugiura, "Bus-based packetized scan architecture trade-offs for heterogeneous multi-core SoCs," in *Proc. IEEE Int. Test Conf. (ITC)*, Nov. 3–8, 2024. [2]

This study evaluates the trade-offs of SSN in heterogeneous SoC environments, comparing it with hierarchical DFT approaches. The authors analyze real-world implementation data, demonstrating that SSN achieves higher scan throughput and better compliance with physical and pin constraints, making it a viable solution for modern SoC testing.

F. Côté, M. Kassab, W. Janiszewski, R. Rodrigues, R. Meier, and B. Kaczmarek, "Streaming Scan Network (SSN): An efficient packetized data network for testing complex SoCs," in *Proc. IEEE Int. Test Conf. (ITC)*, Nov. 1–6, 2020. [3]

This foundational paper introduces the SSN architecture as a scalable, high-speed packet-based scan network. It supports simultaneous testing across multiple cores, accommodates varying scan lengths and pattern counts, and simplifies integration in tiled layouts. The authors compare SSN's efficiency with Intel's Structural Test Fabric, highlighting its adaptability and reduced timing closure effort.

S. Shukla, B. R. Kumar, and V. Singh, "SSSN: Secured Streaming Scan Network," in *Proc. IEEE Latin American Test Symp. (LATS)*, Mar. 21–24, 2023. [4]

While SSN improves test performance, it introduces security vulnerabilities due to its shared bus architecture. This paper proposes a secured SSN (SSSN) framework with enhancements to the Streaming Scan Host (SSH) node, including access authorization and data encryption. The proposed model is evaluated for area overhead and test resource impact, showing minimal degradation while maintaining flexibility.

V. A. Huddar, "On-die SSN methodology for high-speed I/O," in *Proc. IEEE Elect. Design Adv. Packag. Syst. (EDAPS)*, Dec. 14–16, 2020.[5]

This work investigates the impact of simultaneous switching noise (SSN) on high-speed I/O systems, particularly in DDR5 technologies. It presents a simulation-based methodology to analyze and mitigate on-die noise effects caused by internal capacitance and package inductance, helping maintain signal integrity in high-speed environments.

IV. COMPONENTS IN THE DESIGN

The Streaming Scan Network (SSN) design consists of several key components that work together to improve the efficiency, reliability, and scalability of integrated circuit (IC) testing. Below are the major components involved in the design: (see Fig. 2 for SSN architecture).

On-chip controller (occ)

As shown in Fig. 2, The On-Chip Controller (OCC) is a critical component in the design-for-test (DFT) architecture of integrated circuits, responsible for generating programmable clock pulses under the control of Automatic Test Pattern Generation (ATPG). Key OCC components include Clock Selection, which allows the controller to choose between different clock sources such as shift and capture clocks; Clock Chopping Control, which enables the generation of clock pulses with varying widths; and Clock Gating, which controls the enabling or disabling of clock signals to specific parts of the design. OCCs can be classified into Standard OCCs, offering basic clock selection functionality, and Parent-Child OCCs, which provide more advanced clock control by working in tandem with other controllers. Typically, the OCC consists of registers, a clock generator, and control logic, all working together to generate the required clock pulses. The specific architecture and functionality of the OCC may vary based on the design requirements and technology used.

OCC Components

Clock selection: Selects between different clock sources, such as shift clock and capture clock. Used in both

standard and parent occs to route the appropriate clock signal.

Clock Chopping Control: Generates specific clock pulses by suppressing or modifying clock cycles. This is essential for precise timing control, especially during at-speed testing. Typically managed by child or standard occs.

Clock Gating: Enables or disables clock signals to specific parts of the design. This helps in reducing power

consumption during scan or functional testing. Available in standard and optionally in child occs.

OCC Modes: Defines how the OCC operates:

Standard Mode: Performs clock selection, chopping, and gating.

Parent Mode: Only handles clock selection and passes it to a child OCC.

Child Mode: Focuses on chopping and optional gating using the clock from the parent OCC.

OCC Types: Variants of OCC tailored to design needs:

Standard OCC: Supports both fast and slow clocks for capture and shift.

Parent OCC: Manages clock selection for child occs.

Child OCC: Performs clock chopping and optional gating.

Synchronous OCC: Provides clocks synchronized with the system clock.

Mini-OCC: A compact version for area-sensitive designs.

Operating modes

• **Functional Mode:** Activated when test_mode is set to 0. In this state, the fast clock gater is turned on to provide

a high-speed clock to the circuit, while the slow and internal clock gaters remain off to conserve power.

• **Shift Mode:** Triggered when scan_en is enabled (set to 1). The design uses the slow clock to shift data into and

out of the scan chains, including loading the condition bits into the shift register.

• **Shift-Only Mode:** Occurs when test_mode is 0 (disabling the OCC) and scan_en is 1. Here, only the slow clock

is active for shifting, and the shift bypass is enabled to allow scan operations without OCC involvement.

• **Capture Modes:**

• **Slow Capture Mode:** Both shift and capture operations are performed using the slow clock. The number of

capture pulses is defined by the capture_cycle_width parameter.

• **Fast Capture Mode:** Shifting is done with the slow clock, while capture pulses are generated from the fast

clock, supporting at-speed testing.

• **Kill Clock Mode:** When kill_clock_en is activated, all clocks to the design are stopped. This mode helps manage

power or freeze the design in a static condition.

• **Active Upstream Mode:** In this configuration, a component or controller located upstream coordinates or

influences the behavior of downstream modules. This mode is typically used during testing or system

operation to ensure proper data flow and control.

Applications

• Clock Selection and Control: OCCs enable the selection and regulation of clock signals, generating

necessary clock pulses for circuit testing.

• Clock Gating and Chopping: OCCs manage clock gating and chopping, reducing power consumption during

testing and improving test efficiency.

• At-Speed Testing: OCCs facilitate at-speed testing, allowing the circuit to be tested at its operational

frequency to ensure proper functionality under real-world conditions.

• Scan-Based Testing: OCCs assist in scan-based testing by controlling scan clocks and generating the

required clock pulses for testing the circuit.

• Memory Testing: OCCs are used in memory testing, especially for embedded memories, to generate clock

pulses and control signals.

• Scan Testing: Controls clocking during scan-based testing.

• Fault Injection: Introduces faults for reliability testing.

Design for Testability (DFT): Enhances circuit testability.

Fault models

In digital circuit testing, different fault models are used to detect possible defects and ensure correct functionality:

• Stuck-at Fault Model: A signal is stuck at a constant logic 0 (SA0) or logic 1 (SA1), preventing it from changing as expected.

• Pseudo Stuck-at Fault Model: Mainly used in IDDQ (Quiescent Current) testing, this model detects faults by activating them and observing their effect at the cell output.

• Toggle Fault Model: Simulates a fault where a signal keeps toggling between 0 and 1 incorrectly.

• Transition Fault Model: Represents slow signal transitions (slow-to-rise or slow-to-fall) and is detected using at-speed, two-cycle tests.

• Path Delay Fault Model: Simulates delayed signal propagation along a specific path, which can affect circuit timing.

• Bridge Fault Model: Simulates two or more signals being shorted together, causing incorrect behavior.

The On-Chip Controller (OCC) manages the scan test process by controlling the flow of scan data in the Streaming Scan Network (SSN). It ensures synchronization between scan chains and applies test patterns correctly. During Transition Delay Fault (TDF) testing, pipelines are used at the OCC level to delay capture pulses and reduce IR drop, helping improve test accuracy.

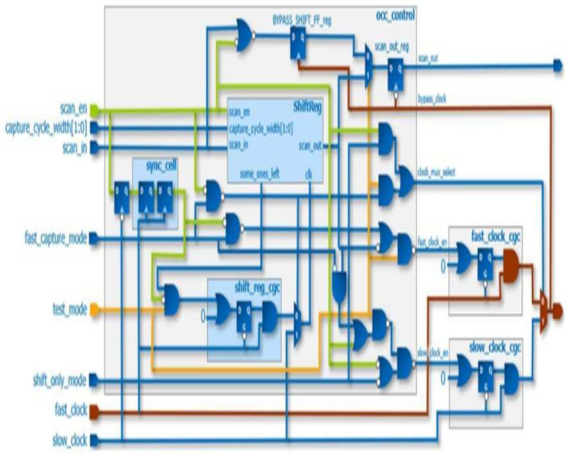

Fig 1 OCC Architecture

Streaming scan network (ssn)

As shown in Fig. 2, The Streaming Scan Network (SSN) is an advanced methodology used in the testing of complex integrated circuits. It enables the streaming of scan test data over a synchronous network, known as the SSN bus, allowing for efficient and flexible test data delivery. The SSN bus clock serves as the synchronous timing source for this data movement.SSN technology is designed to work seamlessly with other testing solutions such as IJTAG (Internal JTAG) and EDT (Embedded Deterministic Test), forming a comprehensive and scalable Design-for-Test (DFT) architecture. Tools like Tessent Shell support SSN integration, offering commands and features for managing SSN datapaths, configuring SSN bus clocks, and generating scan patterns tailored to SSN operation.The adoption of SSN provides numerous benefits, including reduced test time, improved efficiency, and greater flexibility in handling a wide range of design complexities. It also supports advanced test features, such as on-chip comparison and streaming scan data through IJTAG, leading to enhanced test quality and reduced overall test cost.

In summary, the Streaming Scan Network (SSN) is a powerful and efficient solution for modern semiconductor testing, designed to optimize scan data delivery and streamline the testing of increasingly complex System-on-Chip (SoC) designs.

Components of ssn

• Scan Host Module – Manages scan data distribution across the SoC.

• IJTAG Registers – Control scan shift cycles and timing.

• Clock Generation Module – Generates synchronization clocks for scan operations.

• Datapath Shifters – Shift scan data efficiently between cores.

• Scan Control FSM – Handles scan enable and test mode transitions.

• Clock Signals Block – Ensures proper timing closure in scan testing.

• SSN Bus – The main network for transferring scan data.

• BFD/BFM – Manages timing variations between clock tree regions.

• Wrapper Chains – Support hierarchical scan testing and bypass modes.

• On chip compare – used for identical block

5. Uses of ssn

• Reduces test time and cost by optimizing scan data distribution.

• Improves timing closure and eliminates scan chain mismatches.

• Enhances debugging through better test data visibility.

• Supports hierarchical and concurrent core testing in large SoCs.

• Efficient Test Data Delivery: SSN streams test data over the SSN bus for more efficient semiconductor testing.

• Advanced Debugging: Features like on-chip comparison enable quick identification of discrepancies between

expected and actual results, aiding in debugging.

• Comprehensive Test Coverage: Supports multiple test modes (e.g., scan testing, diagnosis, retargeting) to ensure thorough device testing.

• Reduced Testing Time: By utilizing data streaming and advanced techniques, SSN cuts down testing time, boosting productivity and lowering costs.

• Improved Yield Management: Provides yield statistics and failure analysis tools to address yield issues and enhance product quality.

• Support for Hierarchical Designs: Works with hierarchical designs, allowing for reuse of existing test infrastructure and simplifying testing of complex devices.

• Tool Integration: SSN can be integrated with other design and testing tools (e.g., Tessent Shell) for a comprehensive solution.

• Specific Use Cases: Implementing SSN in designs, retargeting test patterns, debugging, and verifying SSN Datapath integrity.

• As shown in Fig. 2, the SSN architecture enables optimized movement of test data through scan chains, facilitating parallel and concurrent testing of multiple chains to improve efficiency. These datapaths also support scalable architectures, making them adaptable for different chip designs.

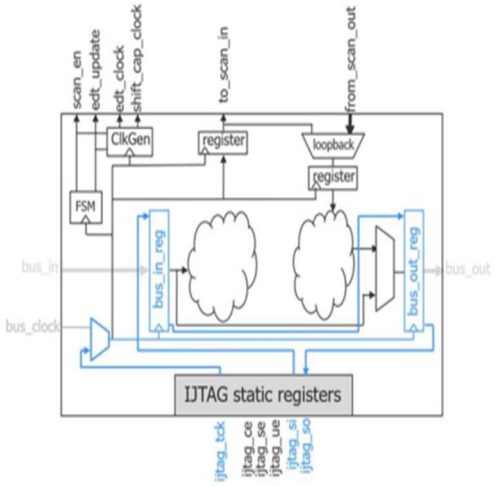

Fig 2. SSN architecture

Embedded deterministic test (edt) logic

Embedded Deterministic Test (EDT) is a scan compression technique used in Automatic Test Pattern Generation (ATPG) to optimize integrated circuit (IC) testing by reducing test data volume, decreasing test time, and improving power efficiency. It works by compressing test patterns before applying them to the chip and decompressing them internally, significantly enhancing the efficiency of the testing process. EDT is a key feature of the Tessent TestKompress tool, which is designed for test compression and embedded deterministic testing, allowing for efficient pattern generation, compression, and application in design for testability (DFT). The tool provides various commands to configure and manage EDT blocks, such as delete_edt_blocks, set_current_edt_block, and report_edt_blocks, while also incorporating EDT rules like K12 and K18 to verify the correctness of the implementation. These features ensure optimal test performance, making EDT a critical technology for improving testing efficiency, reducing data volume, and optimizing power consumption in modern semiconductor testing.

Components of edt

• EDT Controller – Manages test pattern flow and synchronization with the tester.

• Decompressor – Expands compressed test patterns using LFSR, phase shifters, and lockup cells.

• Compactor – Reduces test response data using XOR-based compression and spatial compactors.

• Bypass Logic – Allows switching between compressed and uncompressed scan testing.

• Power Controller – Regulates power consumption during scan testing by controlling switching activity.

Edt controller

• Role: Acts as the timing and control engine that sequences the entire test process.

• Responsibilities: Generates scan enable, shift, and capture signals; manages test mode transitions; controls

decompressor seeds and scan chain activities.

• Advanced Features: Supports various test configurations, adaptive scan control, and integration with poweraware and hierarchical DFT structures.

Decompressor

• Role: Expands low-bandwidth compressed input patterns into full-width scan vectors internally.

• Mechanisms: Uses LFSRs, XOR networks, and mapping logic to distribute compressed data to many scan chains.

• Practical Benefit: Achieves compression ratios up to 100x, significantly reducing tester memory and input pin requirements.

Compactor

• Role: Compresses the output responses from multiple scan chains into compact signatures.

• Mechanisms: Implements MISRs or XOR trees to generate unique compressed responses.

• Challenge Addressed: Maintains fault visibility while reducing test output data and avoiding aliasing of fault responses.

Power controller

• Role: Controls power to specific design regions or scan chains during test mode.

Functions: Activates or deactivates power domains, manages clock gating, and prevents overheating or IR

drop.

• **Design Challenges Addressed**: Enables safe and effective testing of low-power, multi-voltage designs.

Bypass logic

• **Role**: Allows selective deactivation or bypassing of certain scan chains or logic blocks.

• **Use Cases**: Speeds up testing by isolating stable or pre-tested modules; facilitates debug and fault localization

by narrowing test scope.

• **Implementation**: May involve multiplexers or dedicated control paths in the scan structure.

Uses of edt

• Reduces test data volume by compressing patterns.

• Decreases test time by minimizing scan shifts.

• Lowers tester memory requirements, reducing test costs.

• Optimizes power consumption by controlling switching activity.

• Improves fault coverage while maintaining high test efficiency.

• Test Data Compression: Compresses deterministic test patterns, reducing test data volume, test time, and

ATE bandwidth requirements.

• Improved Test Coverage: Increases fault coverage by reaching hard-to-access logic, enabling better

detection of stuck-at and transition faults.

• Low-Power Testing Support: Supports power-aware scan testing through features like clock gating and

shift power control, reducing dynamic power during test.

• Debugging and Diagnosis: Enhances post-silicon debug with deterministic pattern replay and tools like

EDT Finder to isolate and analyze test issues.

• Asymmetrical Channel Configuration: Allows different numbers of input and output scan channels,

optimizing compression ratios and interface design.

• Support for Multiple Scan Groups: Enables partitioning of scan chains into separate test groups, allowing selective and parallel testing for better efficiency and fault isolation.

• Integration with DFT Tools: Seamlessly works with industry-standard DFT tools like Synopsys DFTMAX,

Cadence Modus, and Siemens Tessent for smooth pattern generation and test insertion.

• As shown in Fig. 3, the EDT architecture supports flexible controller options such as general-purpose, TAP-based, and pin-limited types to suit diverse design and test environments. It also provides adaptability to post-synthesis changes, allowing scan chain length modifications while maintaining constraints like longest chain length and lockup latch placement.

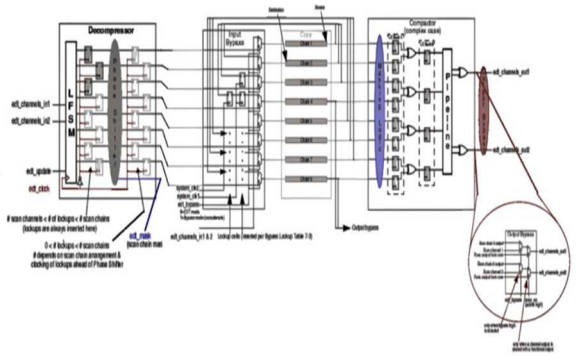

Fig 3 EDT architecture

Pipeline stages for capture pulse delay

Pipeline stages at the OCC level delay capture pulses to mitigate IR drop by spreading switching activity across cycles. TDR stores static values during testing. Pipelining improves performance by dividing tasks into stages executed in parallel, increasing throughput without raising clock frequency. Pipeline registers ensure synchronization and prevent data corruption.

Advantages:

• Improves overall system throughput

• Efficient hardware utilization

• Enables high-speed performance

• Scalable to complex designs

Latency vs. Throughput: Latency (time to complete a single instruction) remains similar, but throughput (instructions per unit time) increases significantly. Hazards in Pipelines:

Data Hazards: Occur when an instruction depends on the result of a previous one. Mitigated using forwarding, stalling, or reordering.

Control Hazards: Caused by branches or jumps. Solved using branch prediction, speculative execution, or delayed branching.

Structural Hazards: Result from resource conflicts. Avoided by duplicating hardware resources or careful scheduling.

Pipeline Depth: Deeper pipelines (more stages) allow higher operating frequencies but increase design complexity. Each stage must be balanced in delay to prevent bottlenecks.

Applications:

• Widely used in CPU architectures (RISC, MIPS, ARM, x86)

• Digital Signal Processing (DSP) and multimedia hardware

• Custom ASIC and FPGA designs for timing closure

• Network-on-Chip (NoC), data streaming, and communication subsystems

Pipeline Balancing: Ensuring that each stage has roughly equal delay is critical to maximize pipeline efficiency and prevent underutilized resources.

Clocking: Each stage operates in sync with the system clock. The pipeline frequency is limited by the slowest stage's delay (critical path).

Superscalar and Multithreaded Pipelines: Advanced designs use multiple pipelines (superscalar) or interleaved instruction execution (multithreading) to enhance parallelism.

Power and Area Trade-offs: More registers and control logic increase silicon area and power consumption. Poweraware pipelining uses techniques like clock gating and dynamic voltage scaling to optimize power usage. Performance Metrics:

Throughput: Measured in instructions per cycle (IPC) or operations per second.

Latency: Measured in clock cycles or nanoseconds per instruction. *Efficiency*: Ratio of useful cycles to total cycles, considering stalls and hazards. Design Considerations:

Timing closure and critical path optimization

Pipeline flush and reset handling Resource sharing and arbitration

Clock domain crossing if used across different frequencies

V. SSN FLOW DIAGRAM

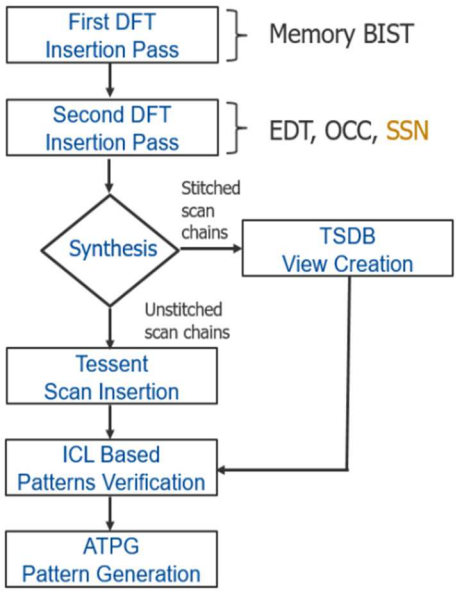

Fig 4. SSN Flow Diagram

As shown in Fig. 4, The flow diagram represents the DFT (Design-for-Test) insertion and ATPG (Automatic Test Pattern Generation) process in VLSI design. The process ensures testability and fault detection in an integrated circuit.

First DFT Insertion Pass (Memory BIST)

• Inserts Memory Built-In Self-Test (MBIST) at the core level.

• Enables testing of embedded memories for faults and potential repairs.

Second DFT Insertion Pass (EDT, OCC, SSN)

• Adds chip-level DFT components, including:

• EDT (Embedded Deterministic Test): Improves test compression.

• OCC (On-Chip Clocking): Manages test clocking efficiently.

• SSN (Signal Switching Noise): Ensures signal integrity during test.

• Synthesis

• Translates RTL design into a gate-level netlist.

• Generates scan chains, categorized as:

Stitched scan chains: Connected and ready for test.

• Unstitched scan chains: Require further scan insertion.

TSDB View Creation

• Applies if scan insertion is performed during synthesis.

• Updates the Tessent Scan Database (TSDB) with scan-related models.

Tessent Scan Insertion

• Inserts and stitches scan chains if they were not completed during synthesis.

• Ensures proper scan connectivity for ATPG.

• ICL-Based Patterns Verification

• Generates and verifies ICL (IJTAG Control Language) patterns.

• Runs continuity simulations to ensure scan connectivity.

ATPG Pattern Generation

• Generates test patterns for manufacturing defect detection, including:

• Stuck-at faults

• Transition faults

• SSN and loopback testing

• Simulates test patterns for verification before fabrication.

VI. IR DROP ISSUES DURING TDF CAPTURE

During Transition Delay Fault (TDF) testing, the On-Chip Controller (OCC) generates two functional clock pulses: one to launch the transition and another to capture it. In a complex design with multiple clock domains at both the block level and system-on-chip (SoC) level, if all OCCs issue capture pulses simultaneously, all clock domains toggle at the same time, leading to a sudden surge in current. This spike can cause IR drop, reducing voltage levels in certain areas of the chip and potentially leading to incorrect test results.

To prevent IR drop issues, pipelines are added at the OCC level to delay the capture pulse during TDF testing. This staggered approach spreads out switching activity, lowering peak power demand without affecting stuck-at testing.

Methods to Reduce IR Drop in TDF Testing

Several Design-for-Test (DFT) techniques can help control IR drop:

Pipelining at the OCC Level – Introducing small delays in the capture pulse helps distribute switching activity and reduce sudden power surges.

Clock Gating and Power Gating – Temporarily disabling parts of the circuit during testing reduces power consumption and prevents excessive voltage drops.

Optimized Test Patterns – Tools like Tessent Shell generate test patterns that minimize unnecessary switching, helping to balance power use.

At-Speed Testing – Running tests at the circuit's actual speed helps identify IR drop issues and ensures accurate fault detection.

By using these power-aware techniques, designers can mitigate IR drop, improve test reliability, and ensure accurate fault detection, even in complex and power-sensitive circuits.

VII. PIPELINE IMPLEMENTATION FOR CAPTURE PULSE DELAY

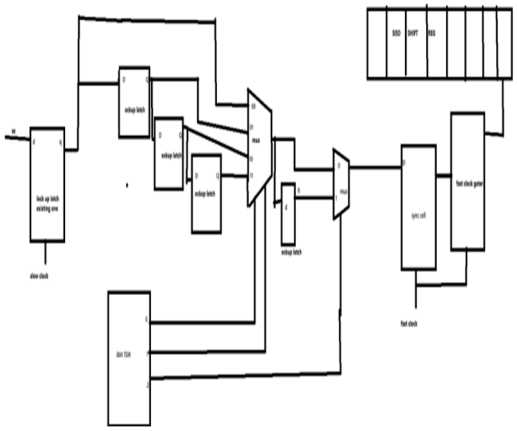

Fig 5: Pipeline Stages for Capture Pulse Delay in OCC.

As shown in Fig. 5, pipeline stages are implemented at the OCC level to delay capture pulses and mitigate IR drop during at-speed testing. The scan-en signal triggers flip-flops for fast-clock capture, while the scan enable register operates on the slow clock. To prevent race conditions between scan-en and the fast clock, a synchronization cell is inserted between the fast clock gater and the scan-en register, which runs on the slow clock

The scan-en signal is used to trigger the flip-flops to capture the fast clock during fast-capture mode, while the scan enable register operates with the slow clock.

To prevent race conditions between scan-en and the fast clock, a sync cell is inserted between the fast clock gater and the scan-en register, which runs on the slow clock.

In stuck-at mode, the OCC shift registers use the slow clock to generate a single pulse. However, in transition mode, they rely on the fast clock, which is supplied by the fast clock gater (as shown in "Figure 1"). The scan enable register and the clock gater introduce a delay in the clock pulse, causing the shift register to receive the clock with some delay. As a result, both pulses experience a delay.

A potential issue arises when all OCCs leak two pulses simultaneously (corresponding to the functional clock), which can lead to IR drop issues.

To mitigate this, we incorporate four pipeline stages that can be programmed via the TDR register. This TDR and the associated multiplexers (MUXes) enable the selection or bypassing of pipeline stages during capture to delay the capture pulses.

The TDR is programmed as follows: 000 (0) → Bypasses all pipeline stages 001(1) → Selects one pipeline stage 011(3) → Selects two pipeline stages 101 (5) → Selects three pipeline stages 111 (7) → Selects four pipeline stages

VIII. DESIGN SPECIFICATIONS

As summarized in Table 1, the design specifications outline the key parameters and constraints for implementing the proposed SSN architecture.

PARAMETER	DETAILS
Technology Node	3nm
Design Hierarchy	Block-Level Implementation
Number of Blocks	2 Blocks + 1 Top
Clock Domains	20
Working Level	Block-Level Experimentation
Synthesis Tool	Design Compiler
Simulation Tools	VCS, Verdi
DFT Tools	Tessent
Timing Analysis	PrimeTime
Power Analysis	Voltus
Signoff Checks	Calibre (DRC/LVS), PrimeTime (STA), Voltus (Power)

Table 1. Design Specifications

IX. RESULTS

Schematic Before Implementation

As shown in Fig 6, The original design includes the SE_SLOW_CLK_inv_reg, which captures input from test_mode_reg using the slow clock. This setup does not include any pipeline stages or delay control. The scan-en signal is used to trigger fast-clock capture, but there is no programmable delay to manage timing or reduce IR drop.

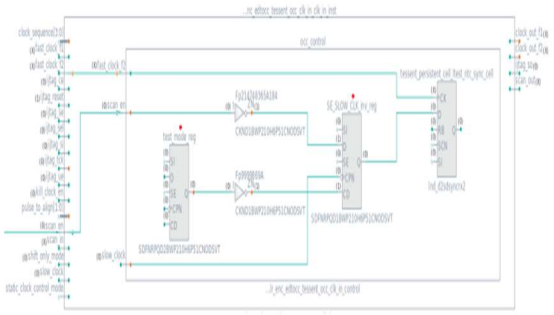

Fig 6 pipeline Schematic Before Implementation

Schematic After Implementation

As shown in Fig 7, the scan capture path with added pipeline stages to control timing and avoid IR drop. The SE_SLOW_CLK_inv_reg is the original register that captures data from test_mode_reg. To add delay when needed, four extra pipeline registers are added. SE_SLOW_CLK_inv_pipe1_reg, SE_SLOW_CLK_inv_pipe2_reg, SE_SLOW_CLK_inv_pipe3_reg, SE_SLOW_CLK_inv_pipe4_reg. A 4:1 MUX selects between the original register and the first three pipeline stages, while a 2:1 MUX helps select the fourth stage. As shown in fig 8, The number of active pipeline stages is controlled by the TDR register, which allows selecting 0 to 4 stages. The scan-en signal, used for fast-capture, is safely transferred from the fast clock to the slow clock using a sync cell. This setup helps avoid timing issues and reduces the chance of IR drop during scan operations.

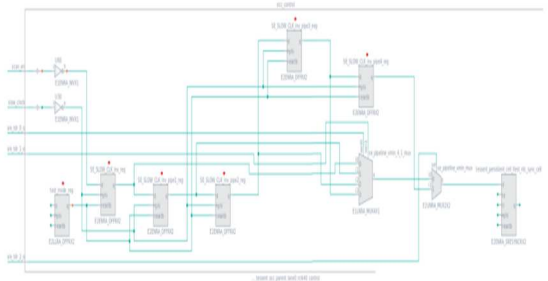

Fig 7 pipeline Schematic After Implementation

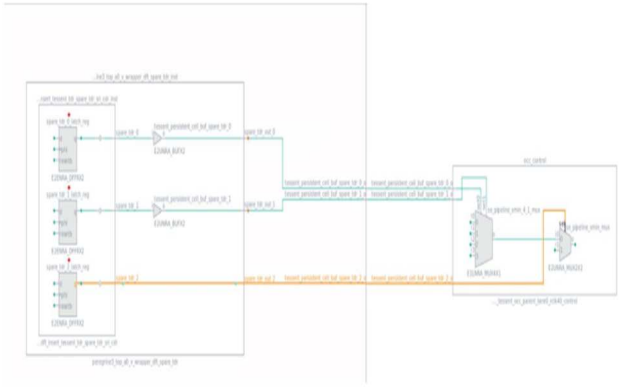

Fig 8 TDR registers

Waveform before implementation

As shown in Fig. 9, the waveform before pipeline implementation illustrates timing behaviour without delay stages.

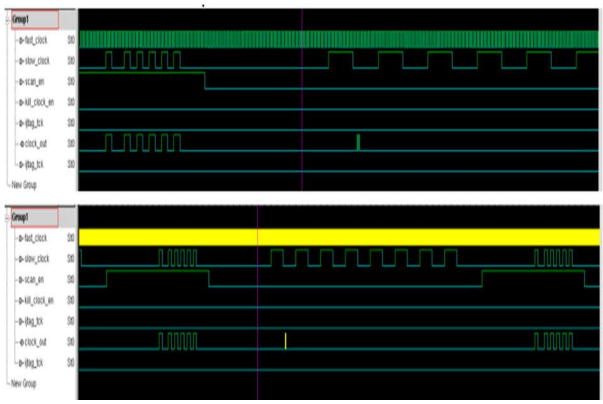

Fig:9 waveform before implementation of pipelines

Waveform after implementation (with SSN)

As shown in Fig. 10, the waveform after implementing pipelines with SSN demonstrates staggered capture pulses and pipeline selection commands, reducing IR drop and improving timing integrity.

"tdf_" "none" "-non_retargetable -append" {} {" processor_core _vmin_stagger_pipeline_select 3"}

"tdf_" "none" "-non_retargetable -append" {} {" processor_core _rclk40_vmin_stagger_pipeline_select enable"}

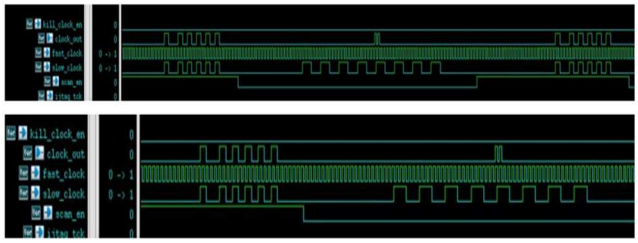

Fig 10 waveforms after implanting pipelines

X. SOC RESULTS

As shown in Fig. 11, the SOC-level results illustrate the impact of pipeline implementation on IR drop reduction and overall test performance improvement.

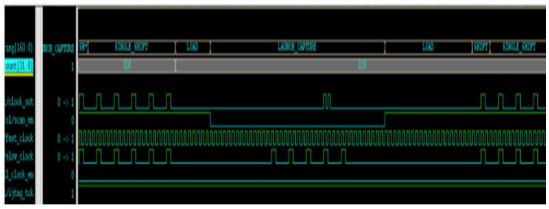

Fig 11 SOC level results

IR drop summary

As shown in Table 2, the IR drop summary compares voltage drop improvements before and after pipeline implementation.

IR Drop(%)	Before	After
5-6	20850	17320
6-7	4953	4626
7-8	2600	1257
8-9	726	328
9-10	115	84
10-11	5	19
11-12	0	1
Total	29249	23635

Table 2 IR drop summary

CONCLUSION

This project focuses on reducing IR drop during scan-based testing by inserting pipeline delays at the OCC (On-Chip Controller) level in the Streaming Scan Network (SSN) architecture.

During transition delay fault testing, high switching activity can cause a sudden drop in voltage (IR drop), which affects the accuracy of test results.

By introducing programmable delays between capture pulses in different clock domains, the switching activity is spread out, reducing peak power and minimizing IR drop.

This method helps in improving the reliability of scan test results without increasing test time or compromising pattern quality.

The approach was implemented using Tessent tools and automated scripting, making it suitable for industry-level testing.

It supports multi-clock domain designs and can be applied to advanced technology nodes.

Overall, the proposed method is an effective and practical solution for improving power-aware testing in modern VLSI designs.

ACKNOWLEDGMENT

I wish to convey my special thanks to my project guide, Dr.G.Sudhagar, from Bharath Institute of Higher Education and Research for his consistent support, timely help, and helpful recommendations over the course of my research.

I sincerely thank everyone for the teaching and non-teaching staff members of Bharath Institute of Higher Education and Research who helped me during the entire course.

REFERENCES

[1] Jonathan Gaudet; Jan Burchard; Matthias Kampmann; Jean-François Côté; Tim Callahan; Hung Ho Chai. “HighBandwidth IJTAG over SSN” in 2024 IEEE International Test Conference-03 November 2024 TO 08 November 2024.

[2] Hiroyuki Iwata; Mahmoud AbdAlwahab; Ron Press; Ohki Sugiura.” Bus-based Packetized Scan Architecture Trade-offs for Heterogeneous Multi-Core SoCs” in 2024 IEEE International Test Conference (ITC) 03 November 2024 - 08 November 2024.

[3] François Côté; Mark Kassab; Wojciech Janiszewski; Ricardo Rodrigues; Reinhard Meier; Bartosz Kaczmarek.” Streaming Scan Network (SSN): An Efficient Packetized Data Network for Testing of Complex SoCs”, in 2020 IEEE International Test Conference (ITC) -01 November 2020-06 November 2020.

[4] Sonali Shukla; Bhavika Ranjeet Kumar; Virendra Singh. " SSSN: Secured Streaming Scan Network," in 2023 IEEE 24th Latin American Test Symposium (LATS), 21 March 2023-24 March 2023.

[5] Vinod Arjun Huddar, " On Die SSN Methodology for High Speed IO”, in 2020 IEEE Electrical Design of Advanced Packaging and Systems (EDAPS), 14 December 2020 - 16 December 2020.