

DFT and Front End Methodology in D2D Analog Mixed Signal IPs

Thesis submitted in partial fulfillment of the requirement for the Award of the Degree of

MASTER OF TECHNOLOGY

in VLSI Design

Submitted By

SHREYASH PURWAR
(602362036)

Under Supervision of

Dr. MOHIT AGARWAL
(ASSISTANT PROFESSOR)

&

Dr. PARVEEN MALIK
(ASSISTANT PROFESSOR)



THAPAR INSTITUTE
OF ENGINEERING & TECHNOLOGY
(Deemed to be University)

ELECTRONICS AND COMMUNICATION ENGINEERING DEPARTMENT

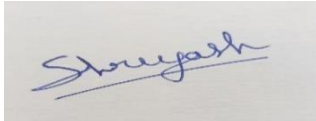
THAPAR INSTITUTE OF ENGINEERING AND TECHNOLOGY

(A DEEMED TO BE UNIVERSITY), PATIALA, PUNJAB

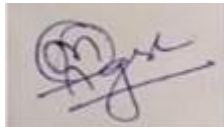
JULY, 2025

DECLARATION

I, **Shreyash Purwar** hereby declare that the work presented in this thesis entitled “ **DFT and Front End Methodology in D2D Analog Mixed Signal IPs** ” in partial fulfilment of the requirement for the award of degree of Master of Technology (VLSI Design) submitted at Electronics & Communication department, Thapar Institute of Engineering & Technology (Deemed to be University), Patiala is an authentic record of work carried out under supervision of **Dr. Mohit Agarwal (Assistant Professor, ECED, Thapar Institute of Engineering & Technology)** , **Dr. Parveen Malik (Assistant Professor, ECED, Thapar Institute of Engineering & Technology)** & **Industry mentor Archana Sainis (DFT Design Engineer Manager, Intel)** from **June 2024 to May 2025**. The matter presented in this has not been submitted in part or full to any other university or institute for the award of any other degree.



Shreyash Purwar
(602362036)



(signature of faculty supervisor)
Name: Dr. Mohit Agarwal
Designation: Assistant Professor



(signature of Industry Mentor)
Name: Archana Sainis
Designation: DFT Design Engineer
Manager



(Signature of faculty supervisor)
Name: Dr. Parveen Malik
Designation: Assistant Professor

CERTIFICATE

DocuSign Envelope ID: FC5518F5-817B-481F-BD45-07A9F60CE9F6

Regd. Office:
Intel Technology India Private Limited
23-56P, Outer Ring Road,
Devarabeesanahalli, Varthur Hobli website: www.intel.in
Bellandur Post
Bangalore 560 103, India
CIN-U85110KA1997PTC021606

Tel: +91-80-2605 3000
Fax: +91-80-2605 6190



To Whomsoever It May Concern

WWID: 12251973

Employee Name: Shreyash Purwar

Internship Dates: 24/06/2024 to 23/05/2025

The letter is to confirm the mentioned above has undergone internship at Intel Technology India Private Limited.

We wish you all the best for your future assignments.

Yours Sincerely

A handwritten signature in black ink, appearing to read "Sumana Dey Srivastava".

Sumana Dey Srivastava

Date: May 29, 2025

Place: **Bangalore**

ACKNOWLEDGEMENT

I would like to place on record my deep sense of gratitude to **Dr. Mohit Agarwal (Assistant Professor) & Dr. Parveen Malik (Assistant Professor)** of Electronics and Communication Engineering, Thapar Institute of Engineering & Technology, Patiala, India for his generous guidance, help and useful suggestions during my project work.

I also wish to extend my thanks to **Dr. Kulbir Singh** (Head of Department) for motivating me throughout my work.

I also wish to extend my thanks to **Archana Sainis (DFT Design Engineer Manager, Intel)** who gave me chance to work in project.

SHREYASH PURWAR

Roll No: 602362036

MTech (VLSI Design)

4th Semester

Department of Electronics and communication Engineering Thapar

Institute of Engineering & Technology

(Deemed to be University), Patiala

ABSTRACT

The main objective of this thesis is to improve the fault coverage of a digital design by targeting specific fault types such as pin constraints, black box faults, and tied cells. The study primarily focuses on two widely used fault models: Stuck-At Fault and At-Speed Fault. A key part of the approach involves addressing black box faults by ensuring that the inputs and outputs of such blocks become observable during the test phase. This improves visibility into areas of the circuit that are otherwise difficult to access, allowing for more thorough testing.

The methodology begins with a gate-level netlist, which undergoes scan insertion to enable better control and observation of internal nodes during testing. The result is a scan-enabled version of the netlist. Throughout this process, various Design Rule Checks (DRCs) are performed to identify rule violations or structural issues that could affect testability. Faults detected at this stage are analyzed and corrected to prevent them from impacting the final coverage results.

Once the design has been cleaned of rule violations and structurally improved, pattern generation is executed on the updated netlist. These test patterns are designed to activate and propagate faults to observable points, increasing the likelihood of detecting them during test application. The overall process leads to a significant enhancement in fault coverage, ensuring that more potential defects are identified during manufacturing tests. Ultimately, this improves the design's testability and contributes to the production of more reliable and higher-quality semiconductor devices.

Keywords: DRC, Scan, Black Box, Stuck At, At-Speed

TABLE OF CONTENTS

DECLARATION	ii
ACKNOWLEDGEMENT	iii
ABSTRACT	iv
TABLE OF CONTENTS	v
LIST OF FIGURE	vii
CHAPTER 1: INTRODUCTION	1
1.1 OVERVIEW	1
1.2 BACKGROUND	2
1.2.1 Design for Testability	2
1.2.2 Assertion Based Verification	2
1.2.3 Clock Domain Crossing	3
1.3 MOTIVATION	3
1.4 NEED FOR PROJECT	4
1.5 OBJECTIVES	4
1.6 THESIS ORGANIZATION	5
CHAPTER 2: LITERATURE REVIEW	6
2.1 ASSERTION BASED VERIFICATION (SVA)	6
2.2 CLOCK DOMAIN CROSSING ANALYSIS (CDC)	7
2.3 SCAN INSERTION, ATPG AND GATE LEVEL SIMULATION	8
2.4 FAULT COVERAGE AND IMPROVEMENT	9
CHAPTER 3: RESEARCH METHODOLOGY	10
3.1 DESIGN FOR TESTABILITY	10
3.1.1 Scan Operation	11
3.2 AUTOMATIC TEST PATTERN GENERATION	21
3.2.1 Tessent ATPG Tool	22
3.2.2 ATPG Tool Flow	23
3.2.3 Test Coverage Analysis and Fault Classes	24
3.3 CLOCK DOMAIN CROSSING	27
3.4 ASSERTION BASED VERIFICATION	32
CHAPTER 4: RESULT ANALYSIS	34
4.1 DFT DESIGN RULES CHECKING	34

4.1.1 C3 Violations	34
4.1.2 C4 Violations	36
4.1.3 T3 Violation	37
4.1.4 T4 Violation	38
4.2 ATPG COVERAGE ANALYSIS	39
4.3 SOCIETAL AND ENVIRONMENTAL IMPACT	42
CHAPTER 5 : CONCLUSION & FUTURE SCOPE	43
REFERENCES	44

LIST OF FIGURES

Fig. 3.1: DFT Design Flow	10
Fig. 3.2: Scan Cell	11
Fig. 3.3: Scan Chain from Scan Cell	12
Fig. 3.4: Scan Opeartion	13
Fig. 3.5: Stuck At Fault	14
Fig. 3.6: AT Speed Fault	15
Fig. 3.7: Path Delay Fault	16
Fig. 3.8: Synchronous Launch on Shift	17
Fig. 3.9: Synchronous Launch on Capture	18
Fig. 3.10: Clocking Architecture	19
Fig. 3.11: Scan Clocking Architecture	20
Fig. 3.12: Inputs and Outputs for ATPG Tool	22
Fig. 3.13: Tessent Tool Flow	24
Fig 3.14 ATPG Unused Faults	25
Fig 3.15 ATPG Tied Faults	25
Fig 3.16 ATPG Blocked Faults	25
Fig. 3.17: ATPG Untestable Faults	26
Fig. 3.18: Fault Hierarchy	27
Fig. 3.19: Example of Clock Domain Crossing	28
Fig. 3.20: Setup/hold Violation	28
Fig. 3.21: CDC using Dual FF Synchronizer	29
Fig. 3.22: Asynchronous FIFO	31
Fig. 3.23: Assertion States	32
Fig. 3.24: Assertion Waveform	32
Fig 3.25 Functional vs Assertion Based Verification	33
Fig. 4.1: Clocking that results in C3 violation	34
Fig. 4.2: C3 Violation Snippet	34
Fig. 4.3: Timing Capture for C3 Violation	35
Fig. 4.4: Clocking that results in C4 violation	36

Fig. 4.5: C4 Violation Snippet	36
Fig. 4.6: Timing Capture for C4 Violation	37
Fig. 4.7: T3 Violation Snippet	37
Fig. 4.8: T3 Violation Backtracking	38
Fig. 4.9: T4 Violation Snippet	38
Fig. 4.10: ATPG Coverage Report	39
Fig. 4.11: Methodology to remove black box faults	40
Fig. 4.12: Fault Coverage before removal of BB fault	40
Fig. 4.13: Fault Coverage after removal of BB fault	41

CHAPTER 1

INTRODUCTION

1.1 OVERVIEW

In the development of mixed signal IPs, ensuring the highest possible performance and reliability is crucial for their integration into larger systems. These IPs combine both analog and digital components, which must operate harmoniously to meet strict performance specifications. The complexity of such designs necessitates a thorough validation process to avoid issues that could lead to functional failures or suboptimal performance in the final product. Design for Testability (DFT) methodologies and front-end RTL (Register Transfer Level) checks play a pivotal role in addressing these challenges by providing a structured approach to design optimization and error detection early in the development process.

Design for Testability (DFT) techniques, such as scan and Automatic Test Pattern Generation (ATPG), play a crucial role in ensuring the performance and reliability of Analog Mixed Signal (AMS) IPs. Scan techniques enhance the testability of a design by improving the controllability and observability of internal nodes. This is achieved by inserting scan chains into the design, which convert flip-flops into scan cells. During testing, these scan cells can be configured into a shift register, allowing test patterns to be shifted in and out. This process enables thorough testing of the internal logic, making it easier to detect and diagnose faults. ATPG is an automated process that generates test patterns to detect faults in the design. ATPG tools analyze the circuit to identify potential fault sites and create test vectors that can effectively detect these faults. These test patterns are then applied to the circuit during testing to ensure that it operates correctly under various conditions. By incorporating scan and ATPG techniques, we can ensure that AMS IPs are thoroughly tested for defects, leading to higher performance, reliability, and manufacturability of the SoCs in which they are integrated

Front-end RTL checks, on the other hand, are essential for verifying the digital logic and overall architecture of the mixed signal IP before it moves to the synthesis and physical design stages. These checks identify design flaws at the earliest stages, ensuring that digital and analog components interface correctly and that the overall functionality aligns with the

intended specifications. Through rigorous RTL verification and simulation, issues such as timing violations, logic errors, and improper signal routing can be detected and corrected. In combination with DfX techniques, front-end RTL checks provide a solid foundation for the performance, reliability, and manufacturability of mixed signal IPs, ensuring they meet both functional and quality standards in the final product.

1.1 BACKGROUND

The design and verification of Analog Mixed-Signal (AMS) Intellectual Properties (IPs) are essential for developing modern electronic systems that combine both analog and digital components. These mixed-signal systems are integral to various applications, ranging from consumer electronics to high performance computing and telecommunications. As the complexity of these systems increases, ensuring the reliability and quality of AMS IPs has become a significant challenge. To meet these demands, robust methodologies such as fault coverage, assertion-based verification, linting, and clock domain crossing (CDC) are crucial for maintaining high standards throughout the design and verification processes.

1.1.1 Design for Testability

Design for Testability (DFT) is a methodology used in electronic circuit design to improve the ease and effectiveness of testing integrated circuits (ICs). It involves embedding special design features that facilitate the detection of faults during the testing phase, ensuring that the final product operates reliably. One key technique in DFT is Automatic Test Pattern Generation (ATPG), which generates the test vectors needed to detect potential faults in the circuit. ATPG uses algorithms to create a set of input patterns that are applied to the design, helping to uncover faults such as stuck-at faults, transition faults, and path delays. Together, DFT and ATPG enhance the overall test coverage and fault detection capabilities, leading to higher- quality, more reliable ICs.

1.1.2 Assertion Based Verification

Assertion-based verification is a method that enhances the precision and effectiveness of the verification process by defining expected design behaviors through assertions. These assertions assist in detecting design flaws early, allowing for faster feedback and resolution. In AMS systems, assertions are employed to verify both the analog and digital components of the design, ensuring they align with performance requirements and operate correctly.

The difficulty lies in creating assertions capable of managing the intricacies of mixed-signal systems, where the interaction between analog and digital parts adds complexity.

1.1.3 Clock Domain Crossing

Clock domain crossing (CDC) issues arise when signals transition between different clock domains, which can lead to timing violations and metastability. In AMS systems, this problem is more complex due to the interaction between analog signals and digital clocks. CDC verification is crucial to ensure that data transferred between different clock domains does not lead to unexpected errors or functional failures. Ensuring reliable communication across clock domains is essential for maintaining system stability and performance.

1.2 MOTIVATION

- **Demand for High-Quality AMS IPs:** As the demand for dependable, high- performance electronic systems increase, it is essential to improve the quality of AMS IPs to comply with industry standards. By advancing verification and fault detection techniques, the design process can yield more reliable and durable AMS components, which will benefit various industries.
- **Increasing Complexity of AMS Designs:** As systems increasingly incorporate both analog and digital components, the verification process becomes more complex. Traditional verification methods are frequently insufficient for mixed- signal environments, highlighting the need to improve fault coverage and verification techniques.
- **Significance of Fault Coverage:** Identifying faults early in the AMS design phase is essential to avoid expensive mistakes during later stages of development. Comprehensive fault coverage must encompass both analog and digital components to ensure thorough error detection, guaranteeing the reliability of the final product.

Value of Assertion-Based Verification: By defining expected behaviors in AMS designs, assertion-based verification helps detect functional errors early. This proactive approach ensures the design adheres to performance specifications and prevents potential issues stemming from the interaction between analog and digital components.

- **Challenges of Clock Domain Crossing (CDC):** CDC problems, including timing mismatches or metastability, are particularly challenging in AMS designs because of the complexity involved in analog-digital interactions. Ensuring reliable data transfer across clock boundaries is vital for maintaining system stability and avoiding unpredictable behavior.

1.3 NEED FOR PROJECT

As mixed signal IP designs become increasingly complex, ensuring their reliability and performance is more challenging. These designs, combining both analog and digital components, require early-stage validation to prevent errors and performance degradation in the final product. Traditional methods often fail to adequately address the interactions between analog and digital parts, leading to potential failures. Thus, there is a clear need for a project that integrates DfX methodologies and front-end RTL checks to catch issues early and optimize mixed signal IP designs.

This project will focus on applying Design for Testability (DFT) along with enhanced front-end RTL checks tailored for mixed signal designs. By doing so, it aims to identify and address issues related to reliability early in the design process. This approach is essential to reduce risks and improve both the performance and manufacturability of the final product.

1.4 OBJECTIVES

1. To perform assertion-based verification (SVA) to ensure the design meets its specifications in all scenarios.
2. To ensure proper dependable data transfers between different clock domains through CDC analysis.
3. To perform scan insertion, automatic test pattern generation (ATPG) and gate level simulations on the design.
4. To analyze the fault coverage, report and to improve the coverage.

1.5 THESIS ORGANIZATION:

- The work overview, the project's motivation, and its goals are covered in Chapter 1.
- A summary of the literature review supporting the project work is given in Chapter 2.
- The research approach used for the paper is covered in Chapter 3.
- The Result analysis from the project work are covered in Chapter 4.
- Conclusion and Future scope is carried out in Chapter 5.

CHAPTER 2

LITERATURE REVIEW

The increasing complexity of mixed-signal IP designs, particularly those integrating both analog and digital components, requires comprehensive verification techniques to ensure performance, reliability, and functionality. This literature review discusses recent advances in assertion-based verification (SVA), clock domain crossing (CDC) analysis, automatic test pattern generation (ATPG), and fault coverage analysis, all essential to enhancing the verification process of mixed-signal systems.

2.1 ASSERTION BASED VERIFICATION (SVA)

Assertion-Based Verification (SVA) is widely recognized as a powerful technique for verifying the correctness of designs by specifying and checking expected behavior during simulation. SVA enables automated detection of design violations related to timing, logic, and functional correctness, especially in complex systems such as mixed-signal IPs. Recent studies have underscored the effectiveness of SVA in identifying subtle design errors that are difficult to detect using conventional simulation methods. According to Lee et al. (2021), SVA plays a crucial role in detecting corner-case scenarios and ensuring compliance with temporal properties, thus providing confidence in system behavior across various operating conditions (Lee et al., 2021).

Additionally, SVA allows for the integration of formal verification techniques, offering a more comprehensive validation process for both analog and digital components. In mixed-signal designs, where the interaction between analog and digital blocks is critical, SVA enables early detection of synchronization issues, timing violations, and other complex faults (Sung et al., 2022).

Fang et al. (2024) introduced an automatic assertion generation framework called AssertLLM, which processes complete specification files using customized Large Language Models (LLMs). The framework breaks down the task into three phases: extracting structural specifications, mapping signal definitions, and generating assertions (Fang et al., 2024).

Mann and Kumar (2023) discussed the design and verification of the AMBA AHB protocol using System Verilog in the Synopsys VCS® tool. Their study highlights the implementation of assertions for different components of the design and the functional verification of the overall design (Mann and Kumar, 2023).

Smith et al. (2023) focused on the verification of mixed-signal System-on-Chip (SoC) designs, which involve complex interactions between analog and digital components. Their paper emphasizes the use of mixed-signal simulators to verify these interactions and the role of assertions in enhancing the verification process (Smith et al., 2023).

2.2 CLOCK DOMAIN CROSSING ANALYSIS (CDC)

Clock Domain Crossing (CDC) analysis is vital for ensuring reliable data transfer between different clock domains, which is a significant challenge in complex mixed-signal systems. As designs with multiple clock sources are common, CDC errors such as metastability, glitches, and data corruption can lead to serious system failures. Recent studies have emphasized the importance of automated CDC analysis tools to identify

and resolve issues early in the design cycle. According to Wang et al. (2022), effective CDC analysis is crucial for mixed-signal IPs, as it helps detect issues in the synchronization of digital and analog signals across asynchronous clocks (Wang et al., 2022). These tools allow designers to identify critical timing paths and ensure proper synchronization, which is essential to maintain signal integrity and prevent data corruption. Furthermore, the ability to perform comprehensive CDC analysis early in the design process significantly reduces the risk of costly design iterations and improves the overall reliability of the system (Gupta et al., 2023).

Cummings (2008) detailed some of the latest strategies and best-known methods to address passing of one and multiple signals across a CDC boundary. The paper includes techniques related to CDC verification and an interesting 2-deep FIFO design for passing multiple control signals between clock domains (Cummings, 2008).

Gnusin and Zaychenko (2018) outlined CDC issues and their solutions for FPGA designs. The paper presents various design techniques together with real-life examples for Xilinx and Intel FPGA devices,

summarizing the most important CDC guidelines for highly reliable FPGA designs (Gnusin and Zaychenko, 2018).

Smith et al. (2023) presented an efficient way to verify clock domain crossing issues at the FPGA implementation stage. The paper discusses various methods and tools used to guarantee flawless multi-clock design operation, emphasizing the importance of CDC verification in complex FPGA designs (Smith et al., 2023)

2.3 SCAN INSERTION, ATPG AND GATE LEVEL SIMULATION

Scan insertion and Automatic Test Pattern Generation (ATPG) are fundamental techniques in the verification process for detecting faults in digital circuits. Scan insertion introduces additional control logic, enabling easier observation of internal states, while ATPG generates a set of test patterns to validate the functionality of the design. The combination of these methods has proven to be effective in achieving high fault coverage, which is essential in detecting both stuck-at faults and more complex issues such as delay faults. Recent research by Singh et al. (2021) discusses the use of ATPG for mixed-signal designs, where test patterns are generated not only for the digital parts of the system but also for the analog components, ensuring comprehensive fault detection across both domains (Singh et al., 2021). Furthermore, gate-level simulations are crucial to validate the design at the final stage before physical implementation. These simulations verify that the synthesized netlist behaves as expected, including all optimizations, ensuring that the design performs correctly at the transistor level (Chen et al., 2022).

The integration of ATPG with advanced techniques like fault simulation and coverage analysis enhances the detection of faults across a wide range of failure modes, including those specific to mixed-signal systems. Recent advancements have focused on improving ATPG efficiency and test pattern generation for mixed-signal circuits. Zhao et al. (2023) highlight improvements in ATPG tools for mixed-signal designs that account for both analog and digital fault models, providing a more accurate and comprehensive approach to fault detection (Zhao et al., 2023).

2.4 FAULT COVERAGE AND IMPROVEMENT

Achieving high fault coverage is a critical goal in the verification of digital and mixed- signal designs, as it ensures that most potential defects are detected before production. Fault coverage reports generated by ATPG tools indicate the effectiveness of the test patterns, but they also provide insight into areas where coverage may be insufficient. Recent studies have focused on improving fault coverage in mixed-signal systems by analyzing fault coverage reports and identifying gaps in test coverage. According to Kim et al. (2022), improving fault coverage is an iterative process that involves refining test patterns, adding new vectors, or modifying the design to enhance observability and controllability of internal states (Kim et al., 2022). Their work emphasizes that fault coverage analysis must be performed continuously to ensure that test patterns evolve as the design matures.

The importance of fault coverage improvement has been further discussed by Gupta et al. (2022), who note that mixed-signal designs present additional challenges due to the interaction between analog and digital domains, requiring specialized techniques to achieve high fault coverage across both domains. Their study suggests combining ATPG with fault diagnosis methods to not only detect faults but also isolate and address them effectively (Gupta et al., 2022). Moreover, recent innovations in ATPG algorithms have focused on increasing fault coverage in less-explored areas of the design, ensuring a more thorough test process and reducing the likelihood of undetected faults (Liu et al., 2023)

CHAPTER 3

RESEARCH METHODOLOGY

3.1 DESIGN FOR TESTABILITY

Design for Testability (DFT) is a crucial approach in Very Large-Scale Integration (VLSI) design that focuses on improving the testability of integrated circuits (ICs) during both production and operation. As electronic systems grow more complex, traditional testing methods are often insufficient to guarantee the reliability and performance of modern chips. DFT tackles this issue by incorporating testability features directly into the circuit design, making testing more straightforward and efficient.

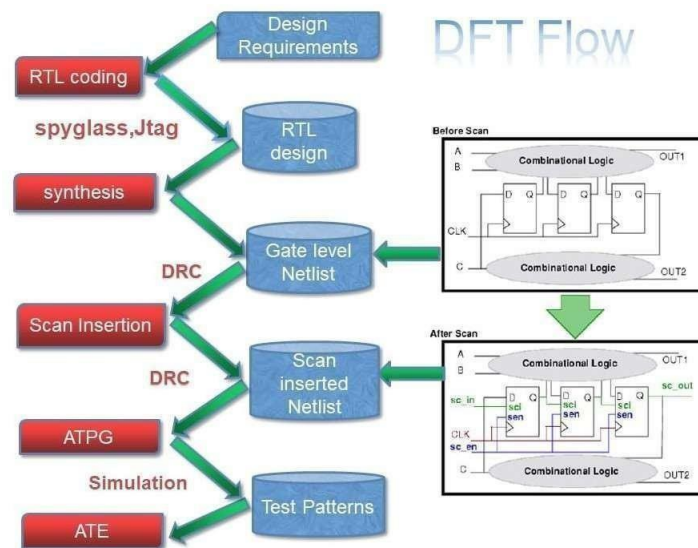


Fig 3.1 DFT Design Flow

The primary goal of DFT is to increase fault coverage, which refers to the percentage of potential defects that can be detected during testing. Achieving high fault coverage is critical for maintaining product quality and reliability, especially in safety-critical applications like automotive, medical devices and aerospace.

DFT techniques are designed to identify manufacturing defects, design errors, and operational faults that may occur over the lifecycle of a product. By enhancing the testing capabilities, DFT helps ensure that only fully functional and reliable products reach the market.

Following table compares the functional verification to DFT.

Functional Verification	DFT
confirms the accuracy of the design	confirms that the created hardware is correct.
Performed by simulation, hardware emulation, or formal methods	Two-part process: Test generation: a software procedure that is carried out once throughout the design phase. performed just once. Test application: Electrical tests conducted on hardware are one example of test application. performed several times.
carried out once before production.	Applications for testing are carried out on each manufactured device.
Responsible for the quality of design	Responsible for the quality of devices.

Table 3.1 Functional Verification vs DFT

3.1.1 Scan Operation

A scan flip-flop is the fundamental component of a scan chain. An internal mux is present at the input of a scan flip-flop. When the active clock edge arrives at CLK, SE (enable signal for mux) decides whether D (functional input) or SI (test input) will reach the flip-flop's output.

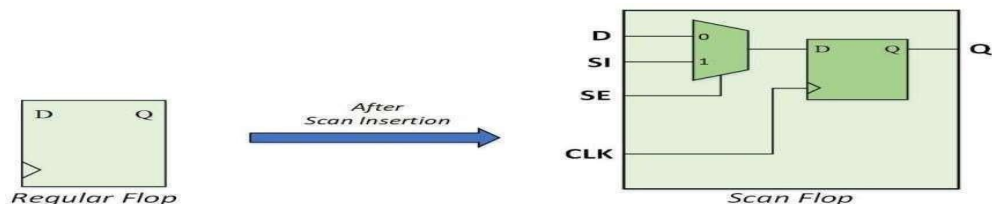


Fig 3.2 Scan Cell

The modified flip-flops, known as scan cells, enable the design to be broken down into smaller, more manageable segments of combinational logic that are easier to test. For a design with millions of flip-flops, incorporating scan cells adds a million observation and control points. This segmentation makes it possible to automatically generate test patterns that can activate the logic between the flip-flops. The testing software does not need to understand the exact function of the logic; it simply targets the segments connected to the scan cells. Since scan testing modifies existing flip-flops to also serve as scan cells, the impact on the overall gate count is minimal, usually adding only 1- 5%.

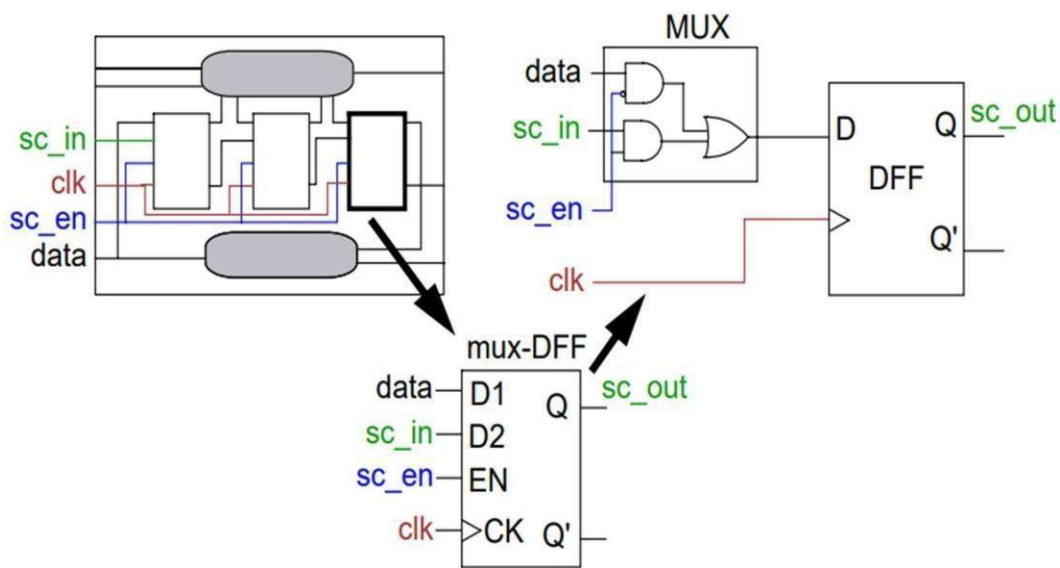


Fig 3.3 Scan chain from scan cell

Scan cells are connected into "scan chains" that act like large shift registers when the circuit enters test mode. These chains are used by external automatic test equipment (ATE) to load test pattern data into the device. Once the pattern is loaded, the design switches back to functional mode, and the test response is captured in one or more clock cycles. The design then returns to test mode, where the test response is shifted out, and the next pattern is simultaneously shifted in. The ATE compares the captured response with the expected result stored in its memory, and any discrepancies are flagged as potential defects for further analysis.

There are three stages of scan operation:

- **Shift In:** Test patterns are loaded keeping the design in test timing mode.
- **Capture:** Design is kept in functional timing mode and test pattern response is captured.
- **Shift Out:** Design is brought back in test timing mode and test pattern response is unloaded.

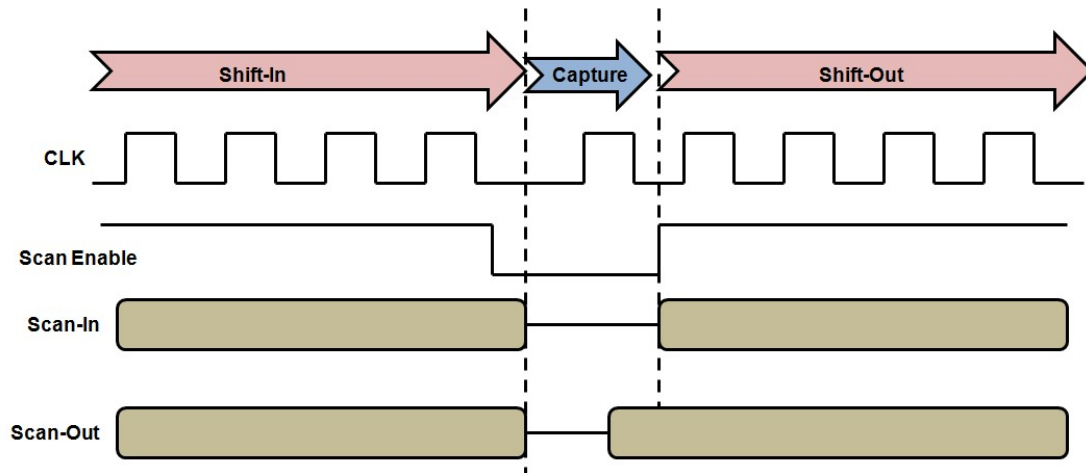
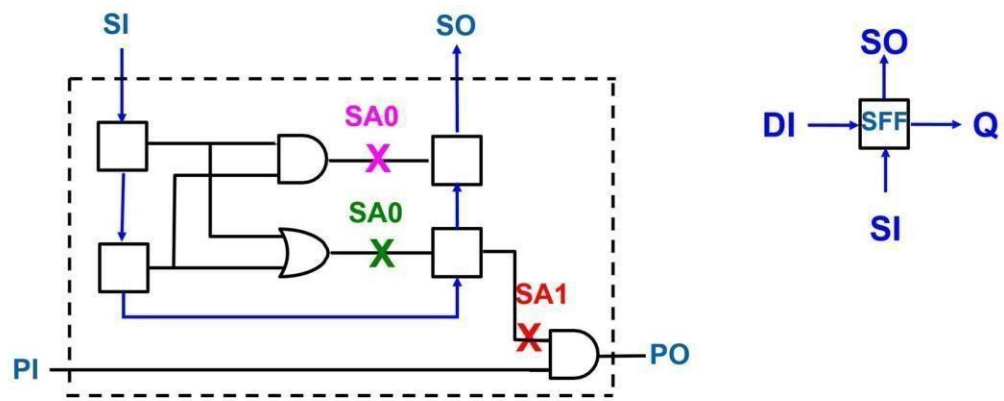


Fig 3.4 Scan Operation

The ATPG tool utilizes fault models to identify the necessary patterns for detecting faults at various points throughout the circuit, typically achieving coverage of 95% or higher. Several different fault models are commonly applied in this process.

Stuck at Test:

The "stuck-at" fault model is the most basic and widely used. It checks each node in the design to see if it is stuck at a logic value of either 1 or 0. For example, if a NAND gate has an input pin shorted to ground (logic 0) due to a defect, the stuck-at-0 test would identify the issue. This model can also detect other defects, such as bridges between two nodes or nets. The stuck-at model is considered a static model because it is a slower test that does not rely on gate timing, such as rise and fall times or propagation delays.



load		capture		unload
CK = PPPP		CK=P		CK = PPPP
SE = 1111	SE 1→0	SE = 0	SE 0→1	SE = 1111
SI = X011				SO = HHXX
PI = 1	PO = L			

VLSI Test 11.3

P = CK pulse
H=expected 1
L=expected 0
Left bit first in/out

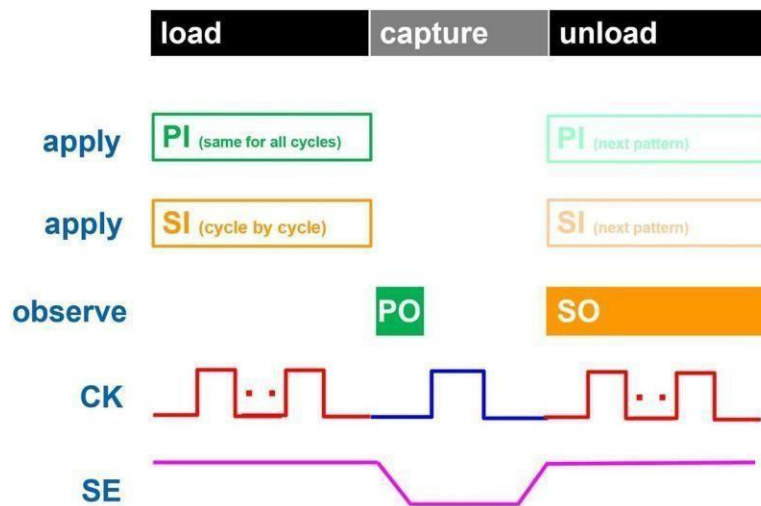


Fig 3.5 Stuck At Fault

At Speed Test:

The "transition" or "at-speed" fault model, a dynamic model aimed at identifying timing problems, is another popular fault model. It recognizes two faults—slow-to-rise and slow-to-fall faults—for every node in the architecture, just as the stuck-at model. The transition fault model creates a transition stimulus that shifts the logic value from 0 to 1 or from 1 to 0 using a test pattern. It is thought that there is a timing defect if the transition does not take place or does not occur within the allotted time.

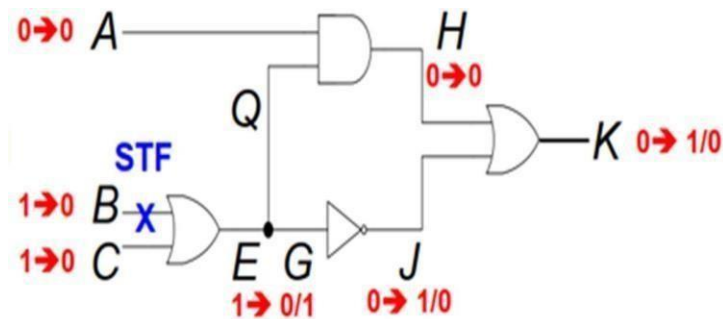
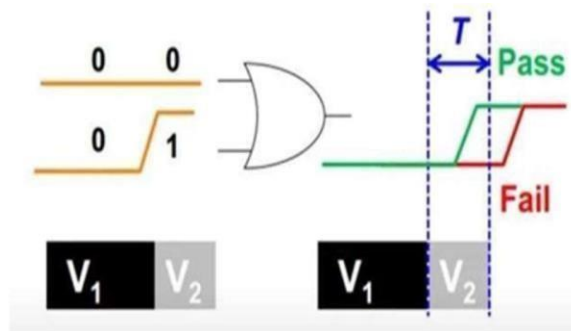


Fig 3.6 At Speed Fault

Path Delay Test:

Additionally dynamic, the "path delay" model tests particular timing-critical paths at speed. The path delay model only considers the paths that the engineer determined to be the most important through static time analysis, in contrast to the stuck-at and transition fault models, which normally test every node in the design. The ATPG program then uses these important paths to produce the path delay test patterns. The basic idea is that other paths with longer slack times shouldn't have timing problems if the most important timing paths pass the checks. In a way, path delay testing acts as a test for manufacturing flaws in specific devices as well as a process check, identifying timing issues if process factors vary too much.

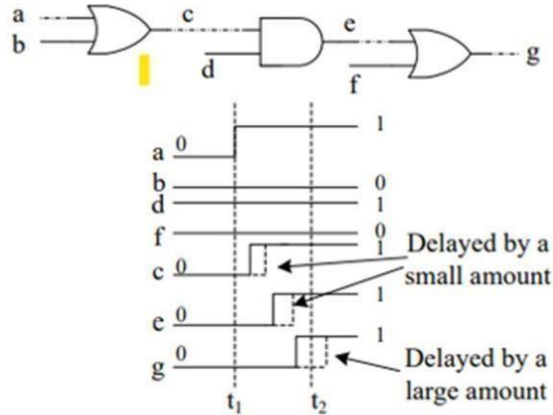


Fig 3.7 Path Delay Fault

Synchronous Launch on Shift:

Synchronous Launch-on-Shift (LOS) is a technique used in delay testing to ensure that the timing behavior of a digital circuit is correctly verified. This method is particularly useful for detecting delay faults that might not be apparent during functional testing. In LOS, test patterns are shifted into the scan chain while the circuit is in shift mode, loading the scan cells with the desired test vectors. After the test pattern is fully loaded, the circuit remains in shift mode for one additional clock cycle, known as the "launch" cycle, where the test pattern is applied to the combinational logic at-speed. Following the launch cycle, the circuit switches to capture mode, capturing the responses from the combinational logic in the scan cells, which is critical for detecting any delay faults that may have occurred during the launch cycle.

LOS provides high fault coverage for delay defects because it tests the circuit at its operational speed and is relatively straightforward to implement in existing scan-based designs. However, it requires fast scan enable signals to maintain timing integrity, which can add complexity to the design and verification process. LOS is widely used in delay testing to identify timing-related defects and is particularly useful in high-speed designs where timing margins are tight. Despite the challenges, the benefits of high fault coverage and effective timing verification make LOS a valuable tool in the design and verification process.

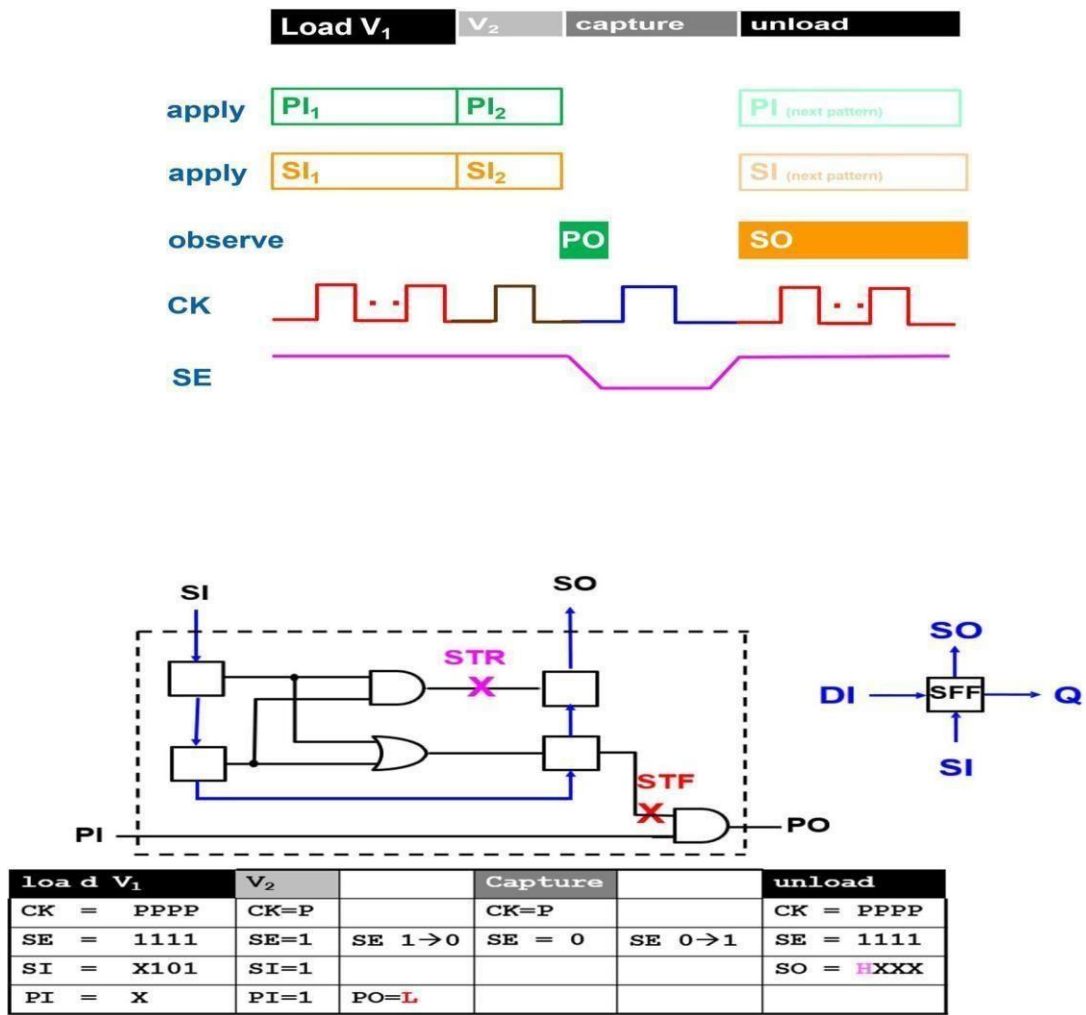


Fig 3.8 Synchronous Launch on Shift

Synchronous Launch on Capture:

Synchronous Launch-on-Capture (LOC) is a technique used in delay testing to ensure that the timing behavior of a digital circuit is accurately verified. This method is particularly effective for detecting delay faults that might not be evident during functional testing. In LOC, the test patterns are first shifted into the scan chain while the circuit is in shift mode, loading the scan cells with the desired test vectors. After the test pattern is fully loaded, the circuit switches to capture mode, where the test pattern is applied to the combinational logic at-speed. This transition from shift to capture mode is the "launch" phase, which is crucial for testing the circuit under real operational conditions. During the capture cycle, the responses from the combinational logic are captured in the scan cells, allowing for the detection of any delay

detection of any delay faults that may have occurred during the launch phase. LOC provides high fault coverage for delay defects because it tests the circuit at its operational speed and is relatively straightforward to implement in existing scan-based designs. However, it requires precise control of the timing signals to ensure accurate capture of the test responses, which can add complexity to the design and verification process. LOC is widely used in delay testing to identify timing-related defects and is particularly useful in high-speed designs where timing margins are tight. Despite the challenges, the benefits of high fault coverage and effective timing verification make LOC a valuable tool in the design and verification process.

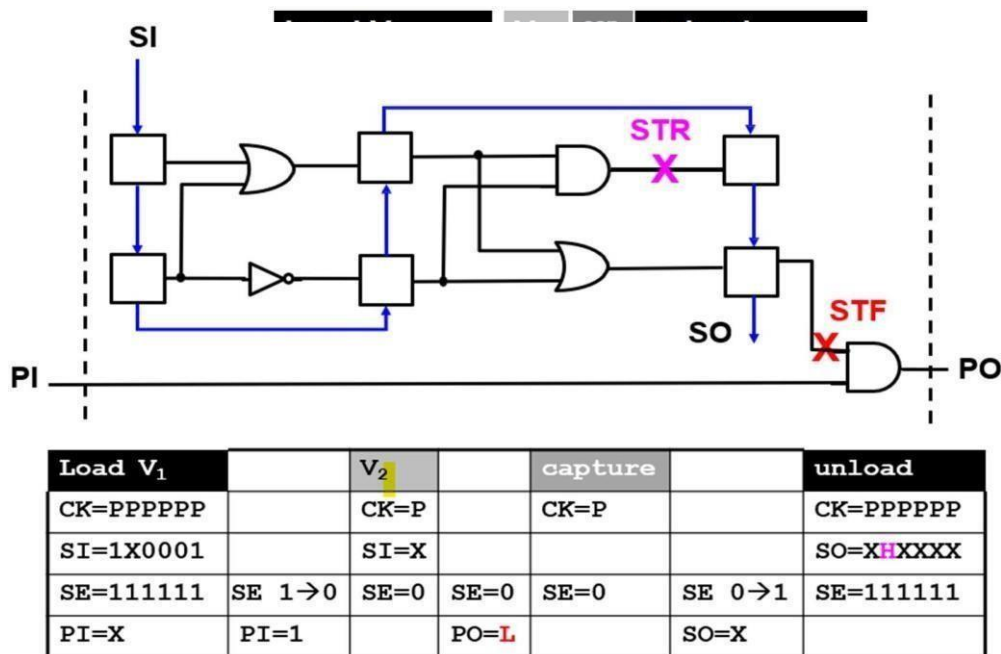


Fig 3.9 Synchronous Launch on Capture

Scan Clocking Architecture:

Scan clocking architecture is a fundamental aspect of Design for Testability (DFT) in digital systems. It involves modifying the clocking scheme of a design to facilitate effective testing, particularly for detecting faults in integrated circuits. This analysis delves into the key components and modifications involved in scan clocking architecture. In a typical functional clocking architecture, a Phase-Locked Loop (PLL) generates multiple clock frequencies to cater to different clock domains within the design. For instance, a PLL might produce clocks at 500 MHz, 400 MHz, and 100 MHz. These clocks are distributed to various parts of the circuit, each operating at different frequencies based on their functional requirements.

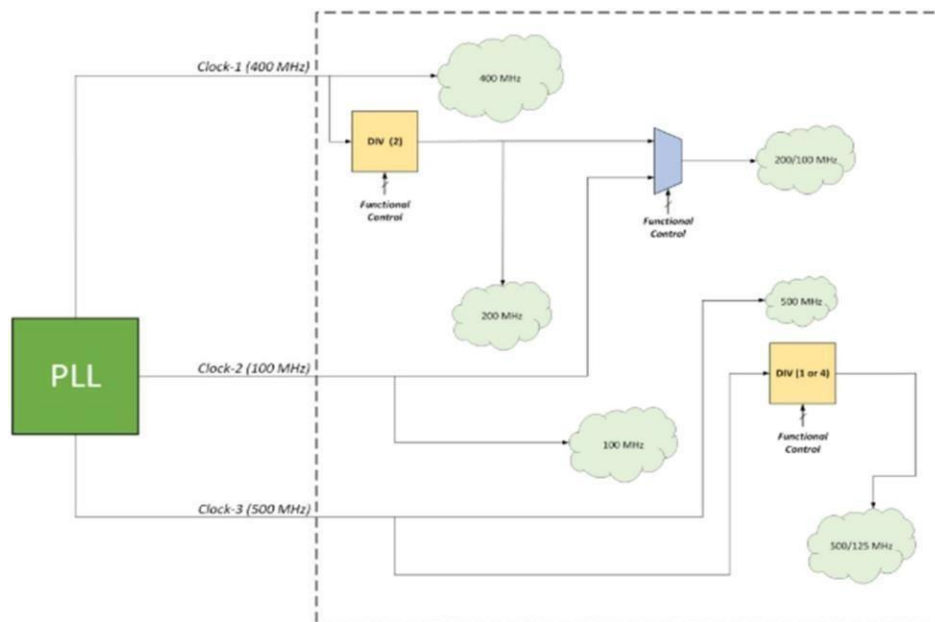


Fig 3.10 Clocking Architecture

Clock dividers and multiplexers (muxes) play a significant role in managing clock frequencies. Dividers reduce the frequency of the input clock; for example, a divider with a factor of 2 will output a clock at half the input frequency. Multiplexers select between different clock inputs based on control signals, allowing the system to switch between various clock sources as needed.

To support scan testing, several modifications are necessary. During scan testing, especially for at-speed tests, each clock domain must be tested at its highest operational

frequency to ensure the circuit can handle its maximum intended speed. Control signals for dividers and muxes must be overridden during test mode to ensure the correct clock frequencies are used, often involving additional muxes to bypass functional control signals. Dividers need to be configured to output the required test frequencies, ensuring stable operation during testing.

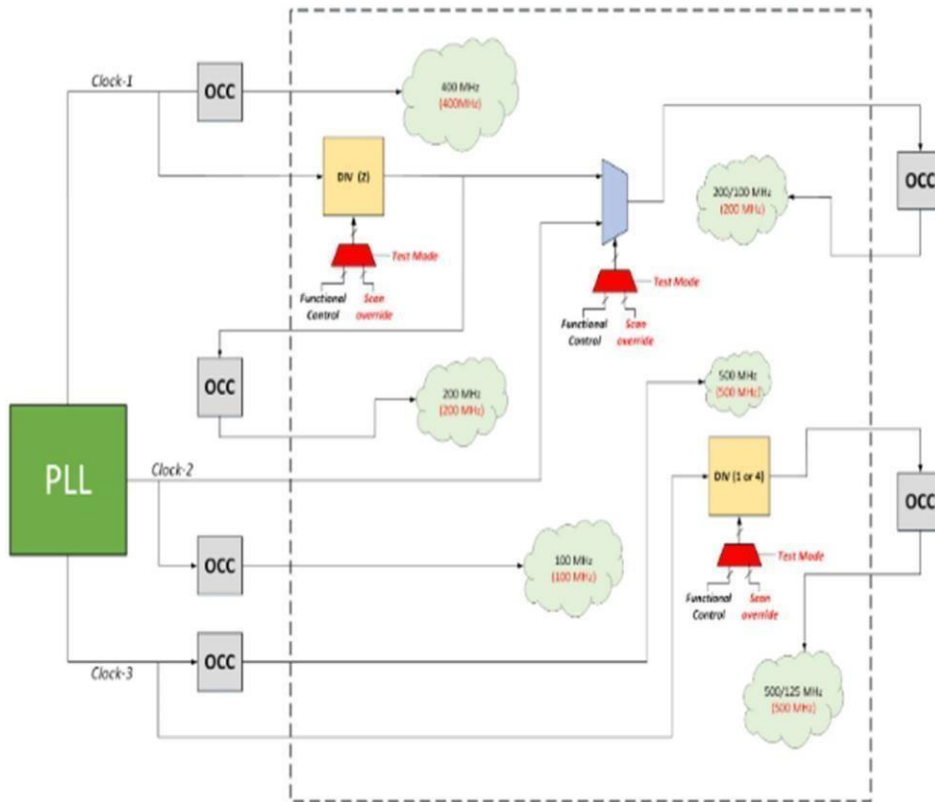


Fig 3.11 Scan Clocking Architecture

Each clock domain requires an On-Chip Clock Controller (OCC) to manage clock behavior during testing. The OCC ensures that clocks are free-running and not gated, providing a stable clock source for the scan chains. In scan design, regular flip-flops are replaced with scan flip-flops (SFFs), which can operate in both normal and test modes. These SFFs are connected to form shift registers, allowing test patterns to be shifted in and out of the circuit. This serialization of test data simplifies the testing process and reduces the number of I/O pins required.

During test mode, the circuit operates differently to facilitate testing. In shift mode, test patterns are shifted into the scan chains. In capture mode, the circuit operates normally for a few clock cycles to capture the test results. Finally, in shift-out mode, the captured results are shifted out for comparison with expected values.

In conclusion, scan clocking architecture is essential for ensuring the testability of digital circuits. By modifying the clocking scheme, managing control signals, and incorporating scan flip-flops and OCCs, designers can create a robust testing environment. This approach not only enhances fault detection but also ensures that the circuit can operate reliably at its intended speeds.

3.2 AUTOMATIC TEST PATTERN GENERATION (ATPG)

ATPG (Automatic Test Pattern Generation) is an essential method in verifying and validating digital circuits, particularly in intricate integrated circuits (ICs). Its primary purpose is to generate a series of test patterns designed to identify faults within a design, ensuring that the circuit functions as intended. These patterns simulate various inputs to detect defects or issues that could impact the circuit's performance. ATPG is a critical component of modern electronic design automation (EDA), enabling engineers to identify manufacturing flaws early in the development process and decreasing the likelihood of defects in the final product.

The ATPG process utilizes fault models to guide the creation of test patterns. These models simulate potential faults in the circuit and help ATPG tools determine the best input combinations for detecting errors. Some commonly used fault models include the stuck-at, transition, and path delay fault models. Each of these models targets specific faults: the stuck-at model detects cases where a node is fixed at either 1 or 0, the transition model looks for timing issues during signal changes, and the path delay model ensures that critical timing paths meet required specifications. ATPG tools generate patterns based on these models to provide thorough coverage, identifying a broad range of possible issues.

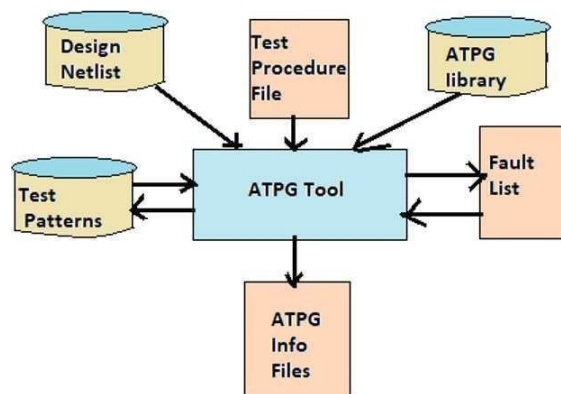
To ensure reliability, ATPG tools strive to achieve high fault coverage, typically over 95%. However, due to the complexity of modern circuits, testing every possible fault

can be time intensive. To make the testing process more efficient, ATPG tools prioritize the most critical nodes or paths in the design. This targeted approach allows for quicker, more efficient testing while still addressing the most significant faults. By concentrating on the most crucial areas, engineers can be confident that the essential parts of the circuit have been tested thoroughly without generating unnecessary test patterns.

Beyond fault detection, ATPG contributes to the overall quality and manufacturability of the design. Using dynamic models like transition and path delay, ATPG can uncover timing issues that might result from variations in the manufacturing process. This is especially important for high-performance circuits, where precise timing is vital. By incorporating ATPG into the design process, manufacturers can lower the risk of producing defective products, enhancing yield and reliability while minimizing testing and rework costs.

3.2.1 Tessent ATPG Tool

Tessent ATPG (Automatic Test Pattern Generation) is a tool designed for semiconductor testing that automatically creates test patterns to identify faults in digital circuits. As a component of the Tessent suite from Mentor Graphics (now Siemens EDA), it enhances test coverage, efficiency, and fault detection by generating patterns for different fault models, including stuck-at and bridging faults. By automating test generation, Tessent ATPG minimizes manual labor and ensures thorough validation of integrated circuits, ultimately improving product quality and yield.



3.12 Inputs and Outputs for ATPG Tool

Inputs to the tool:

- **Design Netlist:** This file contains the structural representation of the circuit design, including all components and their interconnections. It serves as the primary blueprint for the ATPG tool, allowing it to understand the circuit's architecture and identify potential fault locations.
- **Test Procedure File:** This file outlines the specific steps and conditions for applying test patterns to the design. It guides the ATPG tool in executing the tests in a structured manner, ensuring that each pattern is applied under the correct conditions and sequences for optimal fault detection.
- **ATPG Library:** This is a collection of predefined functions, algorithms, and techniques used by the ATPG tool to generate test patterns. It helps the tool apply various strategies for test pattern generation, such as fault simulation and pattern optimization, depending on the specific requirements of the design.
- **Predefined Fault List & Patterns:** This file includes a list of common fault models and test patterns that are known to be effective in detecting specific types of faults in digital circuits. Tessent ATPG uses this list to identify and test for issues such as stuck-at faults, bridging faults, and delay faults, ensuring comprehensive coverage of potential defects.

3.2.2 ATPG Tool Flow

Steps Involved are:

- Invoking the Tessent Shell and set the tool contexts to access ATPG functionality
- Load the ATPG Library and netlist. After this, setup mode is entered
- After setup, tool will create the flattened model of the design.
- In the next step, tool will perform design analysis that will help later in generating the patterns.
- DRC rules are checked in next stage. After the DRC rules analysis, design enters the analysis mode.
- In analysis mode, the tool will start generating the patterns and also, we can add additional set ups like adding the fault list.
- Finally, tool performs fault simulation and gives the coverage report and patterns.

Tool Flow is as shown in below figure.

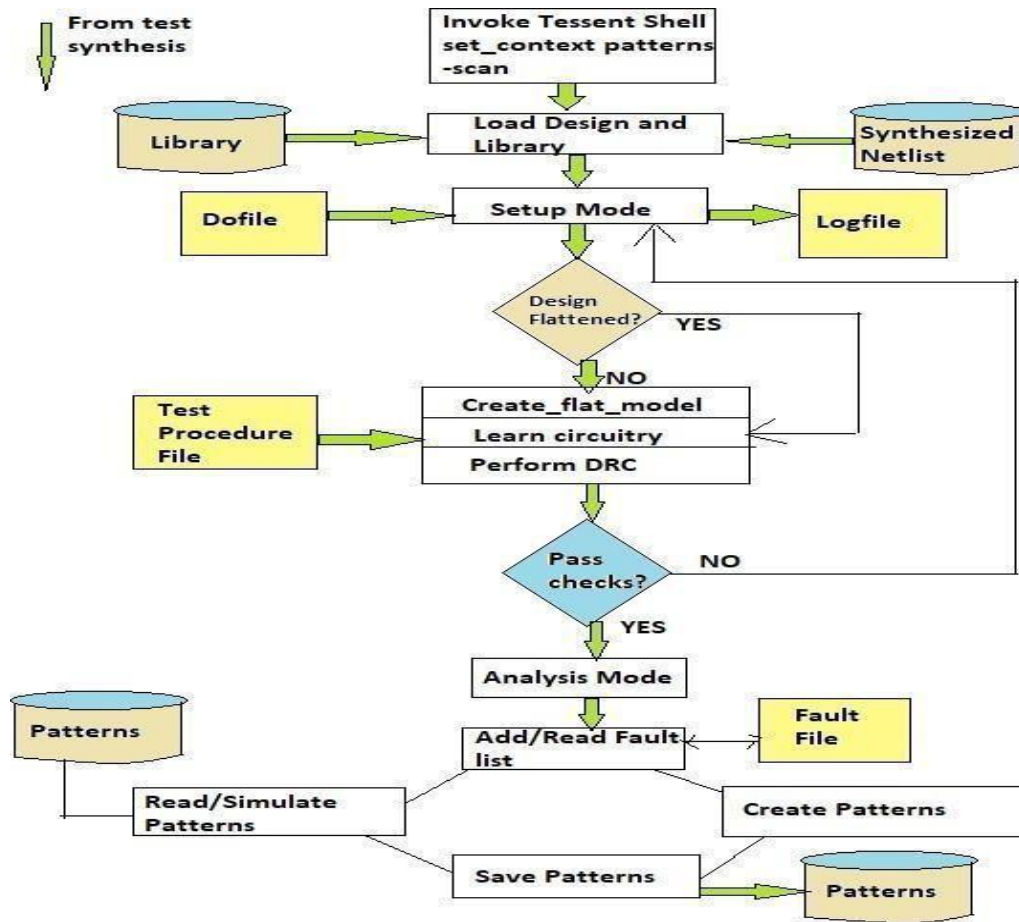


Fig 3.13 Tessent Tool Flow

3.2.3 Test Coverage Analysis and Fault Classes

The tool organizes faults into distinct classes based on their detectability or reasons for being undetectable. Each class is assigned a unique name and a two-character class code. In fault reports, the tool identifies the fault class by either its name or its code.

ATPG Untestable (AU): These are faults for which no test pattern can either detect or

potentially detect them. Since untestable faults do not lead to functional failures, they are excluded from the test coverage calculations. The following outlines the untestable fault categories:

Unused (UU): This category includes faults in circuitry that are not connected to any observation points, as well as faults in floating primary outputs.

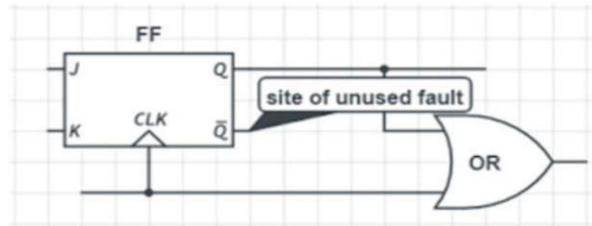


Fig 3.14 ATPG Unused Faults

Tied (TI): The tied fault class refers to faults in gates where the fault location is connected to a value that is the same as the fault's stuck value. Causes of tied circuitry include tied signals, AND or OR gates with complementary inputs etc.

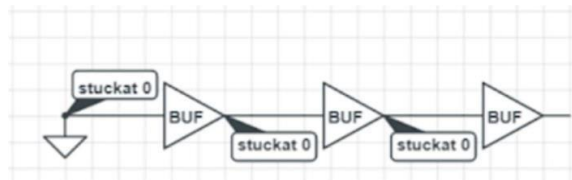


Fig 3.15 ATPG Tied Faults

Blocked (BL): This fault category refers to faults in the circuitry where tied logic obstructs all paths to an observable point. Possible reasons for this blocked circuitry include Tied signals, AND and OR gates with opposing inputs, Exclusive-OR gates with identical inputs

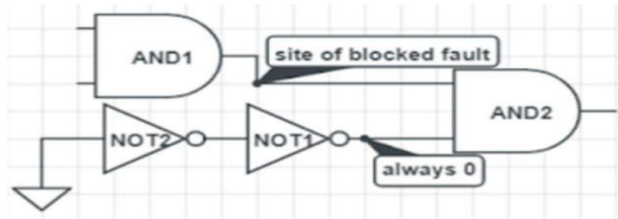


Fig 3.16 ATPG Blocked Faults

Redundant (RE): The redundant fault class consists of faults considered undetectable by the test generator. After all test patterns are applied, the generator performs a specific analysis to ensure that the fault remains undetectable under any condition.

Fault Class	Fault Sub-class	Code	Description
AU	AU.BB — BLACK_BOXES	BB	Fault untestable due to black box
	AU.EDT — EDT_BLOCKS	EDT	AU faults in EDT block
	AU.LBIST — HYBRID_LBIST	LBIST	AU faults in the hybrid IP controller
	AU.PC — PIN_CONSTRAINTS	PC	Tied or blocked by input constraint
	AU.TC — TIED_CELLS	TC	Tied or blocked by tied non-scan cell
	AU.OCC — ON_CHIP_CLOCK_CONTROL	OCC	AU faults in the OCC
	AU.IJTAG — IJTAG	IJTAG	AU faults in the IJTAG instrument
	AU.LPCT — LOW_PIN_COUNT_TEST	LPCT	AU faults in the low pin count test controller
	AU.CC — CELL_CONSTRAINTS	CC	Tied or blocked by cell constraint
	AU.FP — FALSE_PATHS	FP	Fault untestable due to false path
	AU.HPI — HOLD_PI	HPI	Faults in the fanout cone of the hold PIs
	AU.HTC — HOLD_TIME_CAPTURABILITY	HTC	Path hold time fault untestable because it is not capturable
	AU.MCP — MULTICYCLE_PATHS	MCP	Fault untestable due to multicycle path
	AU.MPO — MASK_PO	MPO	Faults in the fan-in cone of the masked POs
	AU.SEQ — SEQUENTIAL_DEPTH	SEQ	Untestable due to insufficient sequential depth
	AU.SSN — SSN	SSN	AU faults in the SSH instruments
	AU.UDN — UNDRIVEN	UDN	Undetectable faults caused by undriven input pins
	AU.WIRE — WIRE	WIRE	Faults that drive a WIRE gate with more than one input
User-defined	none	User-defined AU fault sub-class	

Fig 3.17 ATPG Untestable Faults

Testable (TE) faults are those that cannot be classified as untestable. The following are the testable fault classes:

- **Detected (DT)**: This class includes faults identified by the ATPG process as detected. It consists of two subgroups:
 - **det_simulation (DS)**: Faults detected during fault simulation performed by the tool.
 - **det_implication (DI)**: Faults detected through learning analysis conducted by the tool.

-
- 1. Full (FU)
 - 1.1 Testable (TE)
 - a. DETected (DT)
 - i. DET Simulation (DS)
 - ii. DET Implication (DI)
 - iii. DET Robust (DR)–Path Delay Testing Only
 - iv. DET Functional (DF)–Path Delay Testing Only
 - b. POSDET (PD)
 - i. POSDET Untestable (PU)
 - ii. POSDET Testable (PT)
 - c. Atpg_untestable (AU)
 - d. UNDetected (UD)
 - i. UNControlled (UC)
 - ii. UNObserved (UO)
 - 1.2 UNTestable (UT)
 - a. UNUsed (UU)
 - b. Tied (TI)
 - c. Blocked (BL)
 - d. Redundant (RE)
-

Fig 3.18 Fault Hierarchy

3.3 CLOCK DOMAIN CROSSING (CDC)

It is important to confirm proper synchronization and timing closure across CDC boundaries as AMS IPs includes increasingly intricate clocking with numerous asynchronous domains. By verifying handshake signals, identifying improper crossing patterns, and averting metastability problems, CDC analysis techniques are necessary. As seen in Fig., where clk_A and clk_B signals function at distinct frequencies, clock domain crossings happen when signals move along the circuits controlled by independent asynchronous clocks.

Asynchronous links between IP blocks, gated and divided derivative clocks, separate subsystem clocks with varying frequency, and other clocks with intricate interactions are all integrated into large SoCs. Careful synchronization is necessary when data crosses clock domains, necessitating thorough CDC validation. Metastability that causes data corruption, signal flipping and handshake failures, uncommon corner-case scenarios that are only revealed in the system, and intermittent errors that make defects difficult to duplicate are just a few of the subtle ways that CDC bugs appear and evade modeling. To prevent escaping flaws, early CDC analysis is essential.

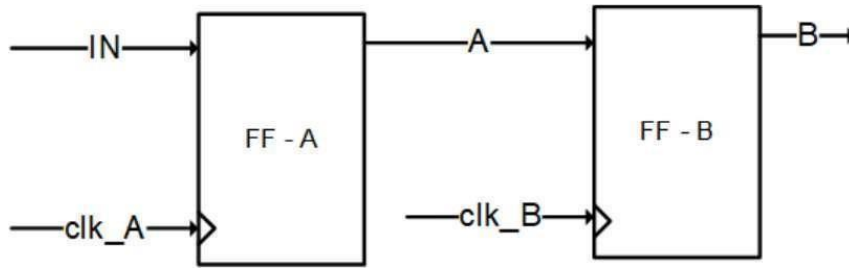


Fig. 3.19 Example of Clock Domain Crossing

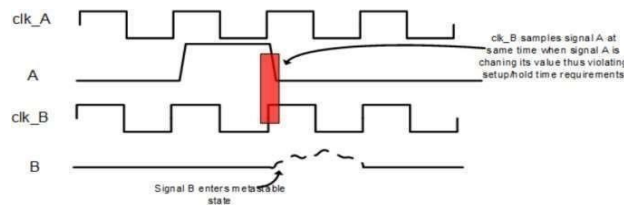


Fig. 3.20 Setup/hold Violation leading to metastability

The inputs and clocks must meet the FFs' setup and hold requirements in order for the flip flops to function properly. Since `clk_A` and `clk_B` are unrelated, signal A in Fig. 6 above is asynchronous to the target clock domain. The setup and hold timing requirements are unlikely to be fulfilled after the asynchronous signal enters the clock domain, allowing FF to transition into a metastable state. The output A of the FF-A deviates from the FF-B's configuration in the image below since it shifts quite near to the `clk_B`'s posge. The downstream FFs in the `clk_B` domain sample may be incorrect as a result of the FF-B output entering a metastable condition and being unable to settle to either 1 or 0.

Finding CDC signals in HDL code, examining synchronizer style and safety, searching for troublesome multi-flop patterns, verifying timing restrictions for crossing signals, and promoting synchronization logic optimization are all steps in the early RTL stage of CDC validation. The most effective way to address problems is through early analysis. Following synthesis, more CDC validation takes place, which includes examining the netlist to identify CDC paths, verifying constraints and false paths, simulating the behavior of synchronization logic, looking for problematic structures that were overlooked in RTL, and verifying synchronizer insertion using synthesis tools. This identifies CDC problems brought on by synthesis optimizations.

Asynchronous handshake paths, in which one side controls transfer timing, source synchronous paths, in which a strobe crosses the clock boundary with data, FIFOs and ripple paths, which necessitate special analysis, and multi-cycle paths, in which data crosses over several cycles, are some of the clocking scenarios involved in CDC paths. Every path type needs the right validation methods. Metastability, in which the synchronizer output enters a metastable state between logic levels, can happen if signal timing exceeds the synchronizer hold time. The settling time for this condition depends on the design parameters. This effect is modeled by mean time between failures (MTBF), with mission-critical SoCs needing a high MTBF (years). A well-designed synchronizer reduces the possibility of metastability.

In order to transfer data safely, CDC tools offer guidance on how to insert synchronizers, automatically place synchronizers, use multi-flop techniques, such as dual-flip flops, for dependability, place them next to the clock splitter for lowest skew, and customize them for different MTBF targets. Figure 7 depicts the dual Flip-flop synchronizer. A circuit with two flip-flops linked back-to-back in the destination clock domain is called a dual flip-flop synchronizer. In the event that setup/hold violations force the first flip flop to enter metastable state, the second flip flop provides sufficient time for the first flop to exit metastable condition. Only the output from second FF will be used by the receiving logic. Theoretically, it is possible that the first FF remains in a metastable state when the second FF samples the value. This could result in the incorrect value being sampled, which could lead to design malfunction (one every MTBF years). Dual FF synchronizers work well for the majority of digital systems, however for some very high-speed clock frequencies, the MTBF can be increased by employing specific cells found in technology libraries or by using synchronizers with more than two stages of FFs (three or four stages).

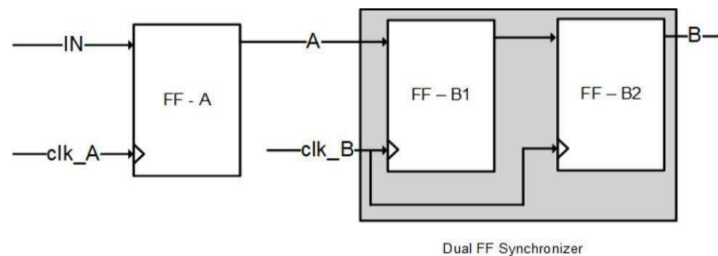


Fig. 3.21 CDC using Dual FF Synchronizer

For CDC closure, intelligent synchronization is essential. Timing constraints, such as source and destination clock periods, maximum delay limits on CDC paths, false paths to prevent pessimistic analysis, and ideal constraints to simulate intended CDC behavior, are crucial for accurate CDC analysis. In order to concentrate CDC tools for this job, constraint tuning was essential. In order to drive fixes, CDC reports necessitate thorough triage, which includes identifying the underlying causes of failing paths, ranking critical failures against false alerts, analyzing clock skews and delay distributions, fine-tuning constraints to get rid of false warnings, and directing synchronizer changes to fix problems. This process had a key role in the shutdown of the CDC. Advanced designs are used in modern CDC tools, such as integrated engines for both RTL and gate-level, scripting APIs for automation and reporting, highly parallel engines for quick analysis, specialized visualizations for effective debugging, and incremental analysis to speed up optimization. Rapid CDC validation was made possible by these features.

The final CDC signoff, which includes thorough analysis of worst-case scenarios, formal verification of complicated synchronizers, expansion of coverage with targeted stimulation, gate-level checks following placement, and the creation of signoff reports and tool certification, guarantees that there are no problems prior to tape-out. Confidence required this thorough pre-tapeout check. Completeness is guaranteed using a coverage-driven methodology that includes creating focused test cases to increase coverage, proving crossing attributes formally, tracking coverage metrics for every domain pair, and iterating until all crossings are covered 100% of the time.

For a multi-clock implementation to be flawless, closure was essential. Layout and floor planning are influenced by CDC criteria, which include limiting clock skew through balanced routing, isolating noise between asynchronous domains, installing synchronizers locally to decrease skew, and imposing time budgets for CDC paths on placement. The requirements for CDC validation served as the basis for layout optimizations. Smooth CDC verification requires careful preparation, which includes budgeting for possible RTL changes depending on findings, early tool evaluation according on design complexity, and planning signoff.

An Asynchronous FIFO (First-In-First-Out) is a type of memory buffer used to safely transfer data between two clock domains that are not synchronized. This design is essential in digital systems where data produced in one clock domain needs to be consumed in another, operating at a different frequency. The core challenge in asynchronous FIFO design is ensuring reliable data transfer without data loss or corruption, which is achieved through careful management of read and write pointers. These pointers are typically implemented using Gray code to minimize errors during clock domain crossing. The FIFO uses control signals to indicate full and empty conditions, ensuring that data is written and read correctly. Asynchronous FIFOs are widely used in applications such as data streaming, communication systems, and any scenario requiring robust data transfer between different clock domains. Proper design and verification of asynchronous FIFOs are crucial to prevent issues like metastability, which can lead to unpredictable behavior and system failures

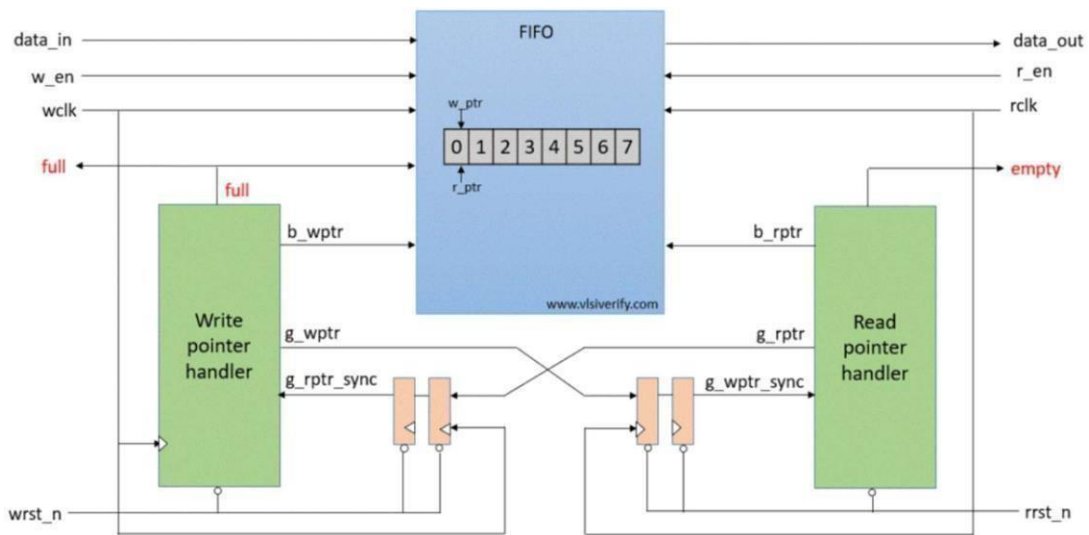


Fig 3.22 Asynchronous FIFO

3.4 ASSERTION BASED VERIFICATION

Assertion-Based Verification (ABV) is a technique used to verify the correctness of digital designs by embedding assertions within the design. Assertions are statements that describe expected behavior or conditions that must always be true. They are typically written in hardware description languages like SystemVerilog.

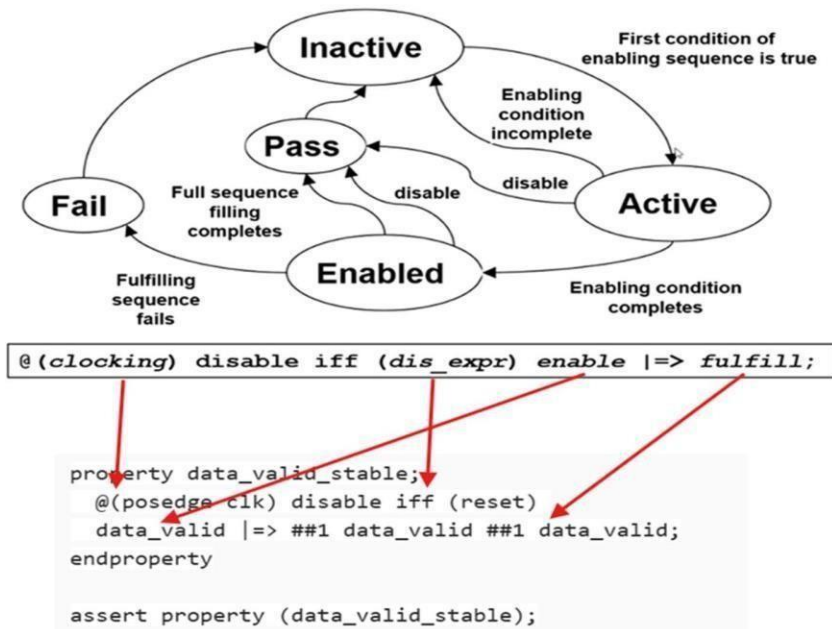


Fig 3.23 Assertion States

Example:

Here is an assertion to check the following property: At the positive edge of clock, if phase enable is active, scale factor should not be equal to zero.

```
REG_NON_ZERO_CHECK: assert property ( @(posedge i_clk) (phase_en == 1'b1) |-> (scale_factor != 9'b0))
else $error("ASSERTION ERROR: RCOMP Scaling factor values are zero for CREG");
```

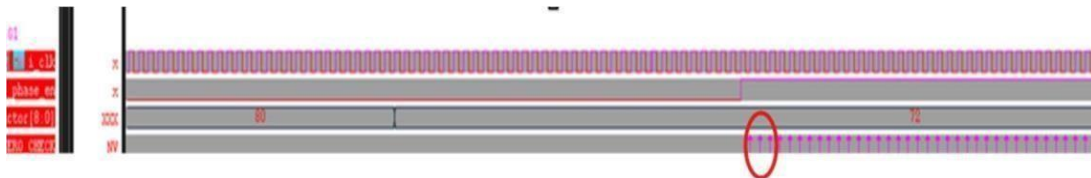


Fig 3.24 Assertion Waveform

Following Table shows the difference between functional verification and ABV.

Characteristic	Functional Verification	Assertion-Based Verification
Approach	Simulations with test benches	Assertions and formal methods
Flexibility	High, can test a wide range of scenarios	Limited to specified assertions
Coverage	Functional coverage metrics	Assertion coverage
Complexity	Requires complex test benches	Requires writing and debugging assertions
Automation	Manual effort required	Automated checking of assertions
Debugging	Manual debugging of test failures	Automatic detection of assertion violations
Scope	Broad, covers many scenarios	Narrow, focused on specific conditions
Early Detection	May detect issues late in the process	Detects issues early by specifying expected behavior

Fig 3.25 Functional vs Assertion Based Verification

CHAPTER 4

RESULT ANALYSIS

4.1 DFT DESIGN RULE CHECKING

4.1.1 C3 Violations

When a sequential element (or RAM) source and sink are driven by the same clock, there is a possibility that the data captured by the sink may come from both the source and sink within the same clock cycle. In other words, the sink could capture the source's updated data (from the current cycle) rather than the previous cycle's data.

Source	Sink
LS	LS
LS	TE
LE	LS
LE	TE

Fig 4.1 clocking that results in C3 violation

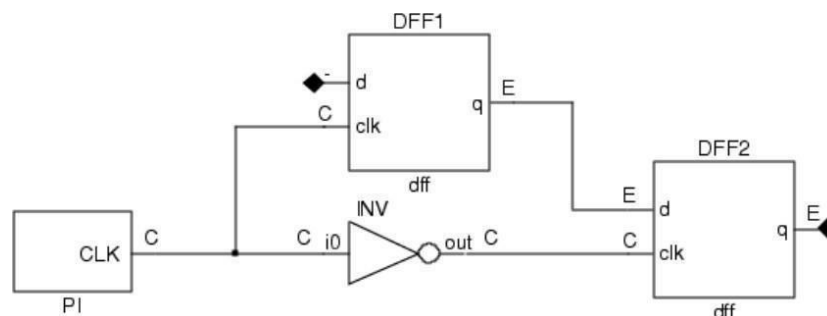


Fig 4.2 C3 Violation Snippet

The figure illustrates an example of a Tessent Visualizer Flat Schematic showing a circuit segment that experiences a C3 violation when the clock's off state at the PI is set to logic 0. In Tessent FastScan and Tessent TestKompress, the off state for each clock is specified using the `add_clocks` command or by utilizing `analyze_control_signals - Auto_fix` during the necessary tool setup. DFF1 updates on the clock's leading edge (LE), while DFF2 updates on the trailing edge (TE) due to the inverter. Under specific timing conditions, such as when data passes through DFF1 to its Q output in less than half a clock cycle, data may flow through both DFF1 and DFF2 within the same clock cycle.

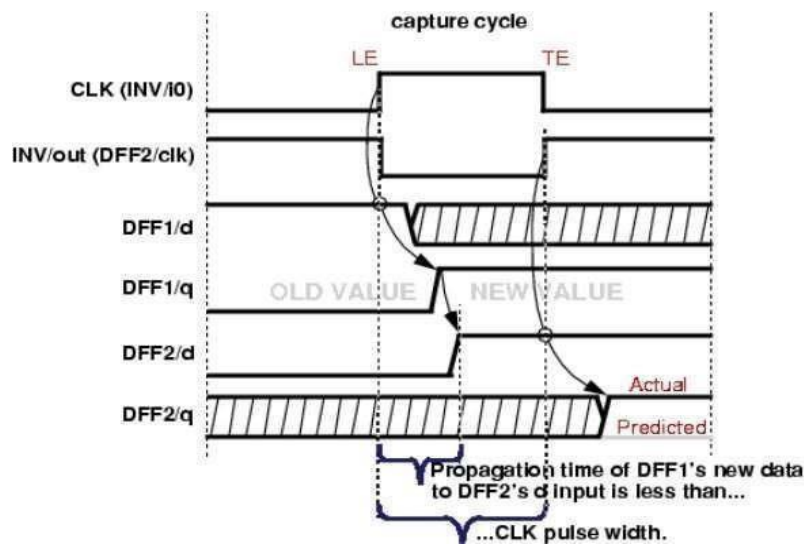


Fig 4.3 Timing Capture for C3 Violation

Fixing C3 Violations:

- Define Clocks Properly: Use tools like Tessent's `add_clocks` command or `analyze_control_signals - Auto_fix` to accurately define clock states for each sequential element. Ensure that clocks are assigned correctly to avoid situations where the sink (flip-flop) updates data on an unintended clock edge.
- Ensure Consistent Clocking: Avoid conflicting clock domains or improper edge selection. Ensure that flip-flops in a scan chain or clocked logic receive the correct clock edges (leading or trailing) as intended.

4.1.2 C4 Violations

When a sequential element (or RAM) source and sink are driven by the same clock, and the source's output is linked via combinational logic to the clock input of the sink, there is a possibility that the data captured by the source could impact the clocking of the sink within the same clock cycle. Specifically, the newly captured value from the source (in the current cycle) may pass through the connected logic and influence the sink's clock signal. By default, the tool anticipates that the downstream combinational logic will use the source's previous value (captured in the prior cycle).

Source	Sink
LS	LS
LS	TE
LE	LS
LE	TE

Fig 4.4 clocking that results in C4 violation

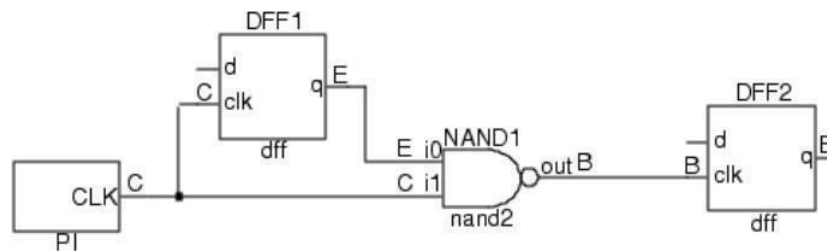


Fig 4.5 C4 Violation Snippet

The figure 4.5 illustrates a potential timing scenario for the circuit segment discussed, where DFF1 updates on the leading edge (LE) of the clock, while DFF2 updates on the trailing edge (TE) due to the CLK signal being gated by the output of DFF1. It is important to note that the old value of DFF1/q disables the clock input to DFF2. As a result, the tool's default simulation assumes that no clock pulse reaches DFF2, causing it to hold its state. However, the leading edge of the CLK signal quickly activates the clock gate, allowing a pulse to reach DFF2's clock input. Consequently, instead of holding its state

as the tool expects, DFF2 captures its D input value on the trailing edge of the now enabled clock gate. C4 violations can be fixed in same way as C3.

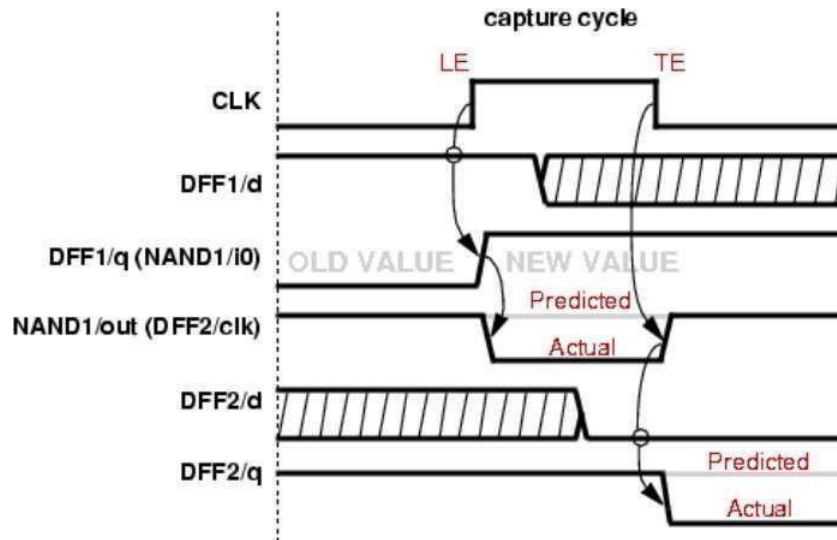


Fig 4.6 Timing Capture for C4 Violation

4.1.3 T3 Violations

The shift process needs to establish a sensitizable route from the scan chain's output to its input. If a gate in this path is not properly sensitized, it results in an error, which the tool flags as a T3 DRC violation.

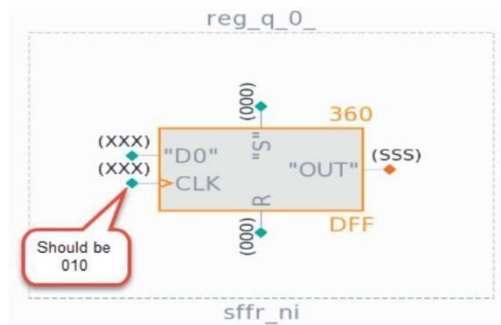


Fig 4.7 T3 Violation Snippet

In the above fig, clock signal should be toggling (010). However, while tracing the design, it is showing XXX configuration resulting in T3 violation.

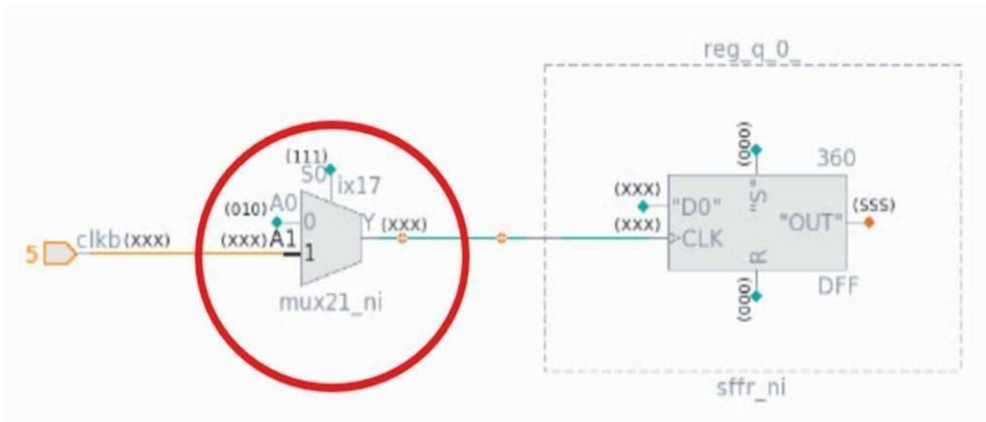


Fig 4.8 T3 Violation Backtracking

With backtracking, it is seen that CLK is driven by another clock clkb which is not toggling. The solution is to modify the test procedure to pulse “clkb” during shift.

4.1.4 T4 Violations

A memory element within the scan path needs to have an active clock at some point during the shift process.

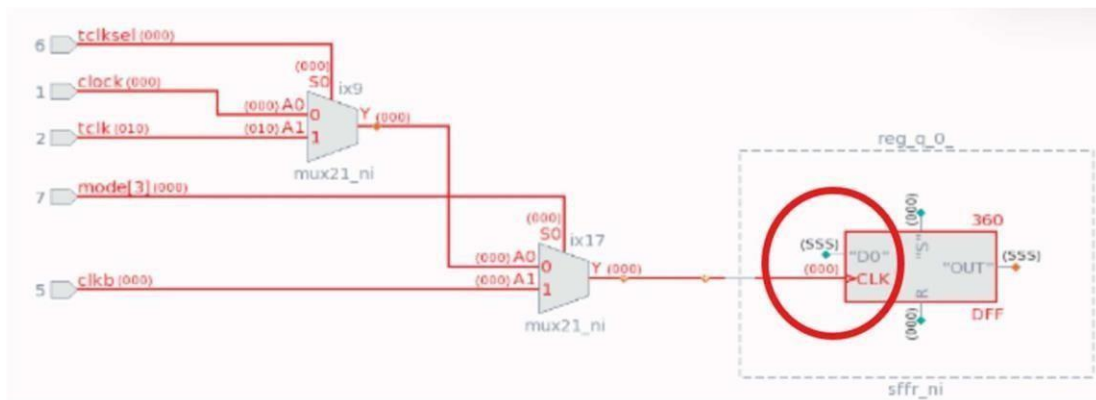


Fig 4.9 T4 Violation Snippet

The traceback indicates that the control signal for MUX /ix17 is set to select input A0. Input A0 is driven by MUX /ix9, whose control signal, tcksel, is also selecting A0. However, the

pulse required to fix the CLK problem is on input A1. Therefore, tclkssel needs to be set high to select A1. To correct T4 violations, ensure that the clock signal is active for a memory element in the scan path at some point during the shift procedure. This active clock period is crucial for the proper functioning of the memory element, allowing it to capture and shift data correctly.

4.2 ATPG COVERAGE ANALYSIS

Statistics Report Stuck-at Faults		
Fault Classes	#faults (total)	#faults (total relevant)
FU (full)	1610404	1552722
UD (unobserved)	16 (0.00%)	same (0.00%)
DS (det_simulation)	1235328 (76.71%)	same (79.56%)
DI (det_implication)	209041 (12.98%)	same (13.46%)
PU (posdet_untestable)	854 (0.05%)	same (0.06%)
UU (unused)	33408 (2.07%)	same (2.15%)
TI (tied)	8699 (0.54%)	same (0.56%)
BL (blocked)	785 (0.05%)	same (0.05%)
RE (redundant)	2129 (0.13%)	same (0.14%)
AU (atpg_untestable)	120144 (7.46%)	62462 (4.02%)
Fault Sub-classes		
AU (atpg_untestable)		
UDN (undriven)	19 (0.00%)	same (0.00%)
BB (black_boxes)	15337 (0.95%)	same (0.99%)
WIRE (dominated_by_wire)	48 (0.00%)	same (0.00%)
PC* (pin_constraints)	9721 (0.60%)	same (0.63%)
TC* (tied_cells)	8663 (0.54%)	same (0.56%)
MPO (mask_po)	2476 (0.15%)	same (0.16%)
SEQ (sequential_depth)	5193 (0.32%)	same (0.33%)
FG_TRISTATELOGIC	255 (0.02%)	same (0.02%)
FG_NELB	6675 (0.41%)	same (0.43%)
FG_SYNCPTH	4206 (0.26%)	same (0.27%)
VISA	4100 (0.25%)	deleted
DFX	20666 (1.28%)	deleted
DIG_VIEW_OBS	26045 (1.62%)	deleted
BURNIN	6871 (0.43%)	deleted
FG_PERLANE	8196 (0.51%)	same (0.53%)
DFX_PIPE_SE	1373 (0.09%)	same (0.09%)
FSCAN_BYP	95 (0.01%)	same (0.01%)
SAFETY_SEAL	179 (0.01%)	same (0.01%)
Unclassified	26 (0.00%)	same (0.00%)
UC+UD		
AAB (atpg_abort)	16 (0.00%)	same (0.00%)

*Use "report_statistics -detailed_analysis" for details.

Fig 4.10 ATPG Coverage Report

The main goal of ATPG was to enhance fault coverage in Design for Testability (DFT) by targeting and eliminating black box faults, which often impede a complete evaluation of a circuit's integrity during testing. The outcomes of our strategy for addressing black box faults indicate a notable increase in fault coverage, resulting in a more comprehensive and efficient testing process. Our approach focused on identifying and analyzing the sections of the circuit that were previously treated as black boxes due to

limited visibility or accessibility during testing. By employing advanced techniques, such as scan chains, we were able to extend the coverage to these previously inaccessible areas. This enabled more thorough fault detection across the entire design, beyond just the sections exposed to standard testing methods. The application of these DFT techniques ensured that faults in both digital and analog components, which might have otherwise remained undetected, were identified and rectified early in the design process.

Flow to improve the coverage by making the black boxes observable.

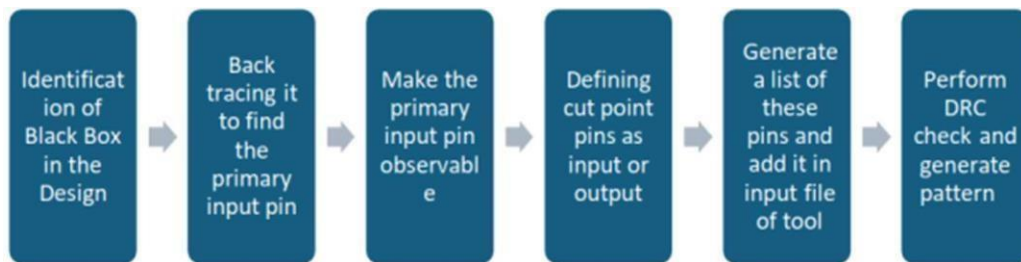


Fig 4.11 Methodology to remove black box faults

The results demonstrate a marked increase in fault coverage when black box faults were mitigated. This increase in fault coverage confirms the effectiveness of our approach in addressing the limitations of black box testing.

Coverage	
test_coverage	91.44%
fault_coverage	88.57%
atpg_effectiveness	99.83%

Fig 4.12 Fault Coverage before removal of BB fault

Coverage		
test_coverage	92.27%	95.80%
fault_coverage	89.69%	93.02%
atpg_effectiveness	100.00%	100.00%

Fig 4.13 Fault Coverage after removal of BB fault

The outcomes of this project demonstrate notable progress in Design for Testability (DFT), with considerable improvements in both fault detection and test coverage. DFT coverage was enhanced from 91.44% to 95.8%, marking a significant 4.4% increase in the efficiency of the testing approach. This enhancement was achieved by refining the scan chain architecture, optimizing test patterns, and introducing more detailed fault models. The increased coverage ensures a larger portion of the circuit's functional areas undergo testing, ultimately improving the detection of potential manufacturing flaws and boosting the overall reliability of the integrated circuits.

In addition to the improved coverage, the number of black box faults decreased from 18,000 to 8,000. These faults, which are typically challenging to diagnose, were greatly reduced through the application of advanced fault isolation and diagnostic methods. By enhancing fault localization and utilizing sophisticated testing techniques, the project was able to minimize the occurrence of these hidden faults that may have otherwise gone unnoticed during production. The reduction in black box faults not only raises the chances of identifying and fixing issues before deployment but also ensures the final product adheres to higher quality standards.

The project successfully met its key objectives of boosting DFT coverage and minimizing black box faults, both of which are essential for strengthening the robustness and dependability of integrated circuits. These results lead to more efficient testing procedures, reduced defect rates, and better yield, ultimately contributing to the production of higher-quality electronic products. The advancements in test coverage and fault detection make the DFT process more effective, ensuring that the designed circuits perform reliably in real-world scenarios.

4.3 SOCIETAL AND ENVIRONMENTAL IMPACT

Electronic device energy consumption can be greatly decreased by optimizing mixed signal IP designs. Increased IP reliability guarantees that vital systems in industries like healthcare, such as diagnostic instruments and medical devices, run more effectively and with fewer malfunctions, improving patient outcomes and the standard of healthcare as a whole. More reliable mixed signal IPs in the transportation sector can help create safer and more efficient automobile systems, reducing the chance of collisions and improving the efficiency of electric cars, hence promoting environmental sustainability. Optimized IP designs help the communications industry by facilitating quicker, more reliable networks, which improve connectivity and information flow— all of which are essential for the advancement of society. In terms of the economy, the initiative stimulates innovation in the semiconductor industry, which leads to the creation of high-tech jobs and boosts economic expansion. By offering cutting-edge products, it also helps businesses become more competitive on a worldwide scale. All things considered, the project promotes a more connected, safe, and sustainable society in addition to improving technological skills.

CHAPTER 5

CONCLUSION & FUTURE SCOPE

In this research work, we looked into how to improve fault coverage in Design for Testability (DFT), with a special focus on removing black box faults during testing. Fault coverage is an important part of making sure digital circuits work correctly and reliably. Improving it helps in creating better-quality products. Black box faults are problems that occur when some parts of a circuit cannot be seen or tested properly. These hidden areas can prevent us from fully checking if everything is working, making it harder to ensure the entire system is reliable.

One of the best ways to deal with black box faults is by improving DFT methods. This includes adding features like scan chains, built-in self-test (BIST), and boundary scan, which help test parts of the circuit that were previously hard to access. Also, using better fault models and simulation tools during the design phase helps detect black box areas early, so we can improve their testability before the chip is even made.

To sum up, removing black box faults is a key step in making DFT more effective. By increasing access to hidden parts of a circuit and using smarter testing methods, we can catch more faults and make our designs more reliable. This not only improves testing but also leads to better product quality, lower costs, and stronger overall performance. As electronic designs become more complex, tackling black box faults will continue to be an important part of making sure our testing keeps up and our products stay dependable.

In the future, the focus can shift towards using smarter and more automated methods to find and fix black box faults earlier in the design process. Techniques like machine learning can be added to testing tools to quickly spot areas in the circuit that are hard to observe, based on data from previous designs. Existing scan and BIST methods can also be improved to handle more complex parts of the circuit. As electronic designs become larger and more complicated, these advanced testing approaches will help in creating designs that are easier to test. This will not only save time and cost during the DFT stage but also lead to more reliable and high-quality products.

REFERENCES

- [1] Lee, J., Kim, M., & Park, S. (2021). "Assertion-Based Verification for Mixed- Signal Circuits." IEEE Transactions on Very Large-Scale Integration (VLSI) Systems, 29(6), 1182-1194.
- [2] Sung, C., Oh, D., & Hwang, Y. (2022). "Leveraging Assertion-Based Verification for Analog- Digital System Co-Verification." Design Automation for Embedded Systems, 26(4), 33-45.
- [3] Wang, L., Zhang, W., & Chen, Z. (2022). "Automated Clock Domain Crossing Verification for Mixed-Signal Systems." Journal of Electronic Testing: Theory and Applications, 38(2), 205- 214.
- [4] Fang, W., Li, M., Yan, Z., Liu, S., Xie, Z., & Zhang, H. (2024). "AssertLLM: Generating and Evaluating Hardware Verification Assertions from Design Specifications via Multi- LLMs." arXiv. 10.48550/arXiv.2402.00386.
- [5] Fang, W., Li, M., Yan, Z., Liu, S., Xie, Z., & Zhang, H. (2024). "AssertLLM: Generating and Evaluating Hardware Verification Assertions from Design Specifications via Multi- LLMs." arXiv. 10.48550/arXiv.2402.00386.
- [6] Gupt Fang, W., Li, M., Yan, Z., Liu, S., Xie, Z., & Zhang, H. (2024). "AssertLLM: Generating and Evaluating Hardware Verification Assertions from Design Specifications via Multi-LLMs." arXiv. 10.48550/arXiv.2402.00386.
- [7] Cummings, C. E. (2008). "Clock Domain Crossing (CDC) Design & Verification Techniques Using SystemVerilog." Sunburst Design, Inc.
http://www.sunburst-design.com/papers/CummingsSNUG2008Boston_CDC.pdf.
- [8] Gnusin, A., & Zaychenko, S. (2018). "Clock Domain Crossings in the FPGA World." Aldec,Inc.
<https://research.ssl.berkeley.edu/~teq/research/Clock%20Domain%20Crossings%20in%20the%20FPGA%20World.pdf>.

- [9] Smith, J., Patel, R., & Lee, H. (2023). "FPGA-based Clock Domain Crossing Validation for Safety-Critical Designs." DVCon Proceedings.
<https://dvcon-proceedings.org/wp-content/uploads/fpga-based-clock-domain-crossing-validation-for-safety-critical-designs.pdf>.
- [10] a, A., Chandra, R., & Kothari, A. (2023). "CDC Verification for High- Performance Mixed- Signal Circuits." IEEE Transactions on Circuits and Systems II: Express Briefs, 70(3), 101-109.
- [11] Singh, R., Sharma, A., & Patel, S. (2021). "Enhancing ATPG for Mixed- Signal Circuits: New Approaches." Journal of Electronic Testing: Theory and Applications, 37(5), 779-790.
- [12] Chen, T., Li, J., & Wang, H. (2022). "Gate-Level Simulation and Its Role in Mixed- Signal IP Verification." IEEE Access, 10, 45678-45687.
- [13] Zhao, Q., Liu, J., & Zhou, X. (2023). "Optimizing ATPG for Mixed-Signal Designs." IEEE Transactions on CAD of Integrated Circuits and Systems, 42(4), 678-687.
- [14] Kim, J., Kim, S., & Lee, H. (2022). "Improving Fault Coverage in Mixed- Signal Designs through Iterative Test Generation." Journal of Hardware and Systems Security, 6(1), 55-65.
- [15] Gupta, R., & Das, S. (2022). "Advances in ATPG and Fault Coverage for Mixed- Signal IPs."
- [16] IEEE Transactions on Instrumentation and Measurement, 71, 1-9.
- [17] Liu, X., Zhang, J., & Zhou, Q. (2023). "Advanced Techniques for Fault Coverage in Mixed- Signal Circuits." IEEE Transactions on Circuits and Systems I: Regular Papers, 70(2), 234 -237

M. Tech Thesis

by Sandeep Mandia

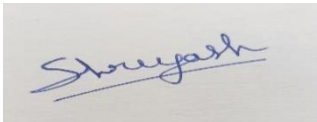
Submission date: 25-Jun-2025 07:07PM (UTC+0530)

Submission ID: 2705848440

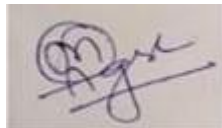
File name: MTech-PVL492-602362036.pdf (1.97M)

Word count: 10512

Character count: 57972



Shreyash Purwar
(602362036)



Dr. Mohit Agarwal
(Assistant Professor)



Dr. Parveen Malik
(Assistant Professor)

M. Tech Thesis

ORIGINALITY REPORT

18%	12%	7%	7%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	documents.mx Internet Source	3%
2	semiengineering.com Internet Source	2%
3	Submitted to Manipal Academy of Higher Education (MAHE) Student Paper	2%
4	Submitted to Thapar University, Patiala Student Paper	1%
5	"Optimizing integrated circuit testing: a comprehensive approach to testability and efficiency", International Journal of Advanced Technology and Engineering Exploration, 2025 Publication	1%
6	tudr.thapar.edu:8080 Internet Source	1%
7	H S Poornima, C Nagaraju. "Functional Verification of Clock Domain Crossing in Register Transfer Level", 2023 International Conference on Recent Trends in Electronics and Communication (ICRTEC), 2023 Publication	1%
8	"A Designer's Guide to Built-In Self-Test", Springer Nature, 2002 Publication	1%
9	www.design-reuse.com Internet Source	1%

Submitted to CSU Northridge

10	Student Paper	1 %
11	www.coursehero.com Internet Source	1 %
12	Submitted to Indian Institute of Technology, Bombay Student Paper	< 1 %
13	arxiv.org Internet Source	< 1 %
14	Xiaokun Yang. "Integrated Circuit Design – IC Design Flow and Project–Based Learning", CRC Press, 2024 Publication	< 1 %
15	Submitted to Amrita Vishwa Vidyapeetham Student Paper	< 1 %
16	Submitted to Indian Institute of Information Technology, Design and Manufacturing – Kancheepuram Student Paper	< 1 %
17	dvcon-proceedings.org Internet Source	< 1 %
18	Lata, Kusum, and Akshay Mann. "Assertion based functional verification analysis of AMBA–AHB using System Verilog", International Journal of Circuits and Architecture Design, 2014. Publication	< 1 %
19	www.techdesignforums.com Internet Source	< 1 %
20	B.D. Cory, R. Kapur, B. Underwood. "Speed binning with path delay test in 150–nm technology", IEEE Design & Test of Computers, 2003 Publication	< 1 %
21	github.com	

Internet Source

< 1 %

22

dokumen.tech

Internet Source

< 1 %

23

dspace.dtu.ac.in:8080

Internet Source

< 1 %

24

Submitted to California State University,
Fresno

Student Paper

< 1 %

25

Cherry Bhargava, Gaurav Mani Khanal.
"Advanced VLSI Technology – Technical
Questions and Solutions", Routledge, 2022

Publication

< 1 %

26

technobyte.org

Internet Source

< 1 %

27

Ye, Sheng. "Novel full bridge topologies for
VRM applications", Proquest, 2011 1109

Publication

< 1 %

28

commsdesign.com

Internet Source

< 1 %

29

katalog.bibliothek.kit.edu

Internet Source

< 1 %

30

Submitted to Charotar University of Science
And Technology

Student Paper

< 1 %

31

P.R. Ramya, V. Beena, G. Radhika, M. Shynu,
K.K. Jayavardhanan. "Molecular
characterisation of coding regions of HIF-1a
gene in Vechur cattle by cDNA sequencing",
Heliyon, 2022

Publication

< 1 %

32

Zainalabedin Navabi. "Digital System Test and
Testable Design", Springer Science and
Business Media LLC, 2011

Publication

< 1 %

33	opencommons.uconn.edu Internet Source	< 1 %
34	www.electronicbus.com Internet Source	< 1 %
35	www.researchgate.net Internet Source	< 1 %
36	eprints.nottingham.ac.uk Internet Source	< 1 %
37	pdfcoffee.com Internet Source	< 1 %
38	www.mdpi.com Internet Source	< 1 %
39	A. A. Fennisi, D. S. Reeves. "Using the stuck-at fault model: how well does it work?", Proceedings of the 28th annual Southeast regional conference on – ACM-SE 28, 1990 Publication	< 1 %
40	f098959c0654768a40578da578b1d99868657eb1.googleusercontent.com Internet Source	< 1 %
41	funrtl.wordpress.com Internet Source	< 1 %
42	technodocbox.com Internet Source	< 1 %
43	www.duo.uio.no Internet Source	< 1 %
44	www.ijmrset.com Internet Source	< 1 %

Exclude quotes Off

Exclude matches < 8 words

Exclude bibliography On