ATPG PATTERN GENERATION AND SIMULATION FOR ENHANCED FAULT COVERAGE WITH DFT FLOW

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

MASTER OF TECHNOLOGY

In VLSI Design

Submitted By

SUMIT MEHTA

6024620026

Under Supervision of

Dr. AMIT MUNJAL

Assistant Professor

Ms. POOJA JAIN

Staff Engineer, STMicroelectronics Pvt. Ltd.

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

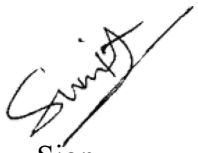
ELECTRONICS AND COMMUNICATION ENGINEERING DEPARTMENT

THAPAR INSTITUTE OF ENGINEERING & TECHNOLOGY
(A DEEMED TO BE UNIVERSITY), PATIALA, PUNJAB

JULY, 2026

DECLARATION

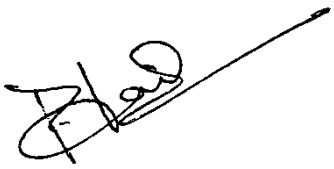
I, **Sumit Mehta** hereby declare that the work presented in this thesis entitled "**ATPG Pattern Generation And Simulation for Enhanced Fault Coverage with DFT Flow**" in partial fulfillment of the requirement for the award of degree of **Master of Technology (VLSI Design**) submitted at the Electronics and Communication Engineering Department, Thapar Institute of Engineering & Technology (Deemed to be University), Patiala is an authentic record of work carried out under supervision of **Dr. Amit Munjal** (**Assistant Professor, ECED, TIET**) from **July, 2025** to **June, 2026**. The matter presented in this has not been submitted either in part or full to any other university or institute for the award of any other degree.



Sign

Date: 30/06/2026

(Sumit Mehta)
(6024620026)



(Ms. Pooja Jain) Designation: Staff Engineer Date:30/06/2026	(Dr. Amit Munjal) Assistant Professor Department of Electronics and Communication Engineering, Thapar Institute of Engineering & Technology, Patiala Date: 30/06/2026

CERTIFICATE

STMicroelectronics Private Ltd.
Plot No.1, Knowledge Park-III,
Greater Noida – 201 308.
Uttar Pradesh, India
Tel : +91-120-4003000
Fax :+91-120-4003007
Email : infostm.india@st.com
CIN : U32109DL1990FTC039906
www.st.com

June 26, 2026

TO WHOMSOEVER IT MAY CONCERN

This is to certify that **Sumit MEHTA** has undergone internship in our **MDRF** Group from **July 25, 2025 to June 26, 2026** and has successfully completed the project on: **ATPG simulations and Test Mode Entry.**

During the internship period, Sumit was found to be sincere and professional in conduct.

We wish him all the best for the future endeavors.

for **STMicroelectronics Pvt. Ltd.**

Sanjay Kumar PIPLANI
India Talent Acquisition Head

Registered Office: S-327, Lower Ground Floor, Greater Kailash – II, New Delhi – 110048
For Employment Verification Related: Please write to sanjeev.kumar1@st.com

ACKNOWLEDGEMENT

I extend my heartfelt gratitude to the individuals who have significantly contributed to my internship journey at STMicroelectronics. Their guidance, support, and encouragement have been invaluable, and I deeply appreciate their contributions.

First and foremost, I sincerely thank my mentor, **Ms. Pooja Jain**, for her exceptional knowledge, expertise, and guidance, which have been pivotal in shaping my professional growth during this internship. Her patience, encouragement, and willingness to share her insights and experiences were truly inspiring. I am grateful for the time and effort she dedicated to mentoring me and for her unwavering support throughout the internship.

I would like to express my deepest gratitude to my thesis supervisor, **Dr. Amit Munjal**, for his invaluable guidance, unwavering support, and constant encouragement throughout this research. His expertise and insightful feedback were instrumental in shaping this work.

I extend my gratitude to **Dr. Kulbir Singh**, Head of the Department (ECE), and **Dr. Shireesh Kumar Rai**, MTech (VLSI) Coordinator, Thapar Institute of Engineering & Technology, Patiala, for providing a conducive learning environment and infrastructure in ECED.

Finally, I thank all my colleagues for their encouragement, insightful discussions, and valuable suggestions throughout this journey.

ABSTRACT

As the complexity of Very Large Scale Integration (VLSI) circuits continue to escalate, ensuring their reliability and correctness has become a paramount challenge. Manufacturing defects, even minor ones, can lead to complete system failure, making robust testing an indispensable part of the semiconductor production cycle. This thesis addresses the critical need for efficient testing methodologies through the exploration of Design for Testability (DFT).

The primary objective of this research is to develop and simulate Automatic Test Pattern Generation (ATPG) strategies to achieve enhanced fault coverage in complex digital circuits. The work begins with a comprehensive review of fundamental and advanced DFT techniques, including scan design, scan compression, and various fault models such as stuck-at and transition faults.

The proposed methodology follows a structured DFT flow, starting from Register-Transfer Level (RTL) design and proceeding through synthesis and DFT insertion. This involves integrating testability structures like scan chains and BIST logic directly into the design to improve the controllability and observability of internal nodes. The core of the work focuses on leveraging these DFT structures for effective ATPG and simulating the generated patterns to validate their effectiveness in detecting manufacturing faults. The research aims to contribute to reducing test time and improving the overall quality and reliability of VLSI products.

Table of Contents

List of Tables

List of Figures

CHAPTER 1
INTRODUCTION

1.1 Background and Motivation

In this new age of electronics, computing devices have evolved to unprecedented lev- els of complexity. This increase in design intricacy has made the process of testing an indispensable and critical phase in the semiconductor manufacturing lifecycle. **Testing is the process of verifying that a circuit functions correctly according to its specified requirements**. It serves a dual purpose: enhancing the final product's quality and contributing significantly to the cost-efficiency of the manufacturing company.

Even a minor defect that goes undetected during testing can lead to complete system failure, resulting in substantial financial losses and harm to a company's reputation. Therefore, **hardware testing is essential to make sure that a design is fault-free** that could adversely affect its operation in the field. The imperative for thorough testing has driven the evolution of sophisticated design methodologies that integrate test considerations from the earliest stages of development.

1.2 Design for Testability (DFT)

Modern design practices have embraced a proactive approach to testing known as **Design for Testability (DFT)**. Instead of treating testing as a post-design activity, the DFT methodology integrates test-related features directly into the circuit design itself. The goal of DFT is not to perform the testing, but rather to **design the circuit in a way that facilitates an efficient and effective test process later on**.

Based on the customer's specifications, the required attributes are defined and the designer develops the RTL. Once the RTL design is finalized, a gate-level netlist is generated and passed to the DFT team. At this stage, DFT structures are integrated into the netlist while respecting timing constraints. The DFT team also receives the SDF file from the STA team, which is created after successful placement and routing, along with results from static and dynamic timing analysis.

DFT plays a vital role in almost all stages of the ASIC design process shown in Figure 1.1. As depicted, the DFT flow is closely embedded within the standard VLSI design cycle. It starts at the Register-Transfer Level (RTL), extends through

synthesis and scan chain insertion, continues into physical design, and includes active participation during both wafer and final package testing.

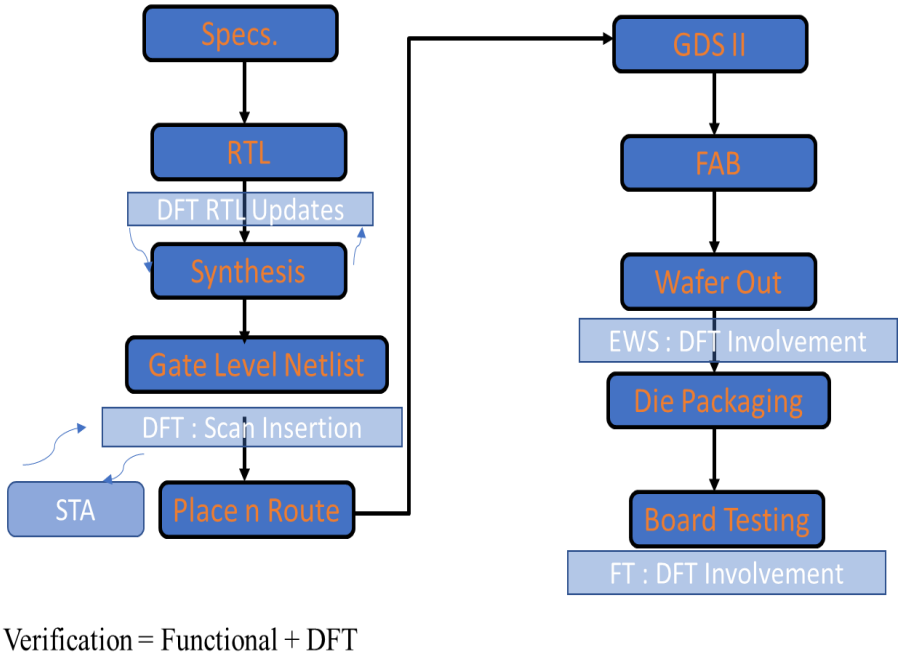


Figure 1.1: Standard Design for Testability (DFT) Flow in VLSI

The two parameters that determines how much a circuit is testable are: Controllability and Observability.

Controllability: The ability of the ATE to set a desired value (0 or 1) at an internal node of circuit under test by applying appropriate test vector to PIs.

Observability: The ability of the ATE to observe an effect of a fault on the primary outputs by propagating the faulty value (0 or 1) from the faulty node to the POs through the application of test vectors to the PIs.

1.3 Fault Modelling

A defect is defined as a physical or logical flaw that occurs in the manufactured integrated circuits. A fault is an abstract equivalent of a physical defect that leads to improper operation of a circuit. Fault causes an error in the circuit. A failure is a change in the system's behaviour from the desired one.

Thus, defects cause faults that lead to errors that ultimately cause failures. A fault model is the abstract description of manufacturing defects. It is simple to determine the

consequences of a fault using a fault model. A good fault model should meet the following two requirements:

• The behaviour should be accurately described.

• It should be efficient enough in terms of test pattern generation and fault simulation.

All the manufacturing defects cannot be captured by a structural test because test patterns are generated based on fault models. Therefore, a quantitative measure called Fault Coverage is used to indicate the ability of a fault model to capture manufacturing defects. It is expressed as:

Fault coverage = (no. of detected faults) / (total number of Faults) [1]

1.3.1 Fault Models

The predominant fault models often employed in the industry are:

- **Stuck-At Fault Model** - In this model, fault is defined by a particular location at which signal permanently takes value of logic 1 or 0. Node that is stuck at logic 0 is named as stuck-at-0 fault whereas node that is stuck at logic 1 is named as stuck-at-1 fault. The key point about such model is that if device under test (DUT) does not exhibit any stuck-at-fault during manufacturing tests, it is highly likely that there is no other fault present.

  The model above can be divided into two categories:

  - Single Stuck-At Fault.
  - Multiple Stuck-At Fault.

  The Single stuck-at-fault model presumes that only one node in circuit under test is stuck at particular logic value. Whereas multiple models allow several nodes to be simultaneously stuck at certain logic levels. Testing circuit under test for presence of single node that is stuck at particular logic level is much easier than detecting multiple such nodes.

- **Delay Fault Model** - This fault model is appropriate for the cases when hazard-induced slow transition of signal on the path leads to delay faults. This model is beneficial when process node scaling is involved. Gate-delay and path-delay are the two most common delay fault models. Gate-

delay fault model is used when the circuit under test has the delay of particular gate as a fault. Under this model, fault effects can be either early or late dependent on the arrival times of inputs signal at the gate. Gate delay can vary depending on the delays of the surrounding circuit. The second model is path-delay fault model. This model is used to describe the case when the cumulative delay along some signal path exceeds the allowed timing constraints.

- **Transistor Fault Model** - The MOSFET is a switch, and one of the most common transistor faults is the switch being stuck either open or closed. Such a fault is also commonly referred to as a "stuck-open" or "stuck-closed" transistor, respectively. One of the ways of detecting a transistor fault in a static complementary MOSFET gate isIDDQ testing. Basically,IDDQtesting checks for abnormal power supply currents due to stuck-short faults, as shorted transistors cause the current to saturate IDDQ. However, another method of checking for transistor faults is also used. The alternative method uses a special current sensor to detect transistor faults in CMOS circuits.
- **Transition Fault Model –** A transition fault occurs when the delay of a gate is so large that a signal transition initiated at an input along the fastest path cannot reach any of the gate's outputs within a clock cycle. For each gate, a transition fault may be either slow-to-rise or slow-to-fall. The total number of transition faults in a circuit, therefore, is 2 times the number of gates.
- **Path-Delay Fault Model –** A path-delay fault occurs when the propagation delay along a particular combinational path of the circuit under test is higher than the permissible delay. The propagation delay for a path-delay fault begins at a PI or clocked flip-flop, propagates through a set of connected logic gates, and ends at a PO or clocked flip-flop. For each such combinational path, two different delay faults can occur corresponding to the rising and falling transitions.
- **Bridging Fault Model –** As the name suggests, a bridging fault occurs when two nets are shorted together. The logic value of the shorted net depends on which of the two nets is dominant, which can result in either an OR-type bridge (shorted net has logic value 1) or an AND-type

bridge (shorted net has logic value 0). If there are no stuck-at faults in a circuit, then most probably there are no bridging faults.

1.4 Types of Test modes:

When a DFT-inserted core is integrated into a top-level design, the internal scan chain structures of the core are logically bound to the top-level scan chain architecture. To ensure testability of the core logic independent of the other design elements, the core must be wrapped in a wrapper structure. DFT Compiler provides two methods for implementing such a wrapper. The basic flow provides straightforward implementation whereas the optimized reuse flow reduces area and timing overhead by taking advantage of existing functional registers in the design.

1.4.1 Internal Scan

Internal scan mode is the extended version of hierarchical scan mode, in which short chains of compressed scan cells are combined to form long conventional scan chains. This DFT method provides maximum fault coverage. The decomposition of sequential circuits into maximal or near-maximal combinational circuits greatly simplifies the process because much less complex test patterns are required to capture faults. The internal scan mode is automatically generated when compressed scan logic is inserted into the design because it changes existing sequential circuits into ones that include a serial shift capability in addition to whatever functions they performed before. Internal scan mode is illustrated in Fig. 1.2.

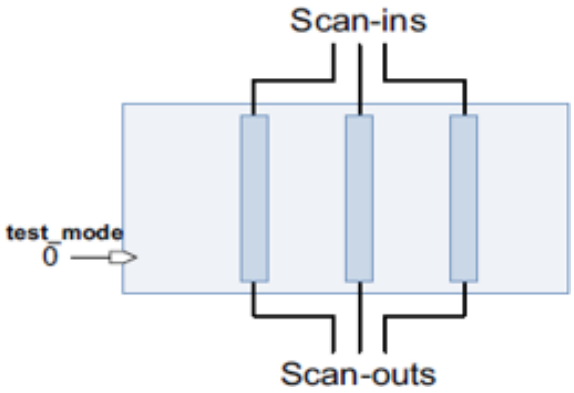


Figure 1.2 Internal Scan Mode

1.4.2 Scan compression

Scan Compression is an internal scan methodology that uses a compression codec—comprising a Compressor and Decompressor—to drive and compress data within the internal scan chains. This approach is employed to test core-internal logic while

substantially lowering the volume of test data and the time needed for test application. It is only implemented when scan compression is activated, and it is widely adopted to minimize both test duration and the number of required pins.

Figure 1.3 illustrates the scan compression method, which greatly reduces the simulation time necessary for validating test patterns. However, the added compression hardware introduced into the design also brings additional potential fault sites that must be detected during testing.

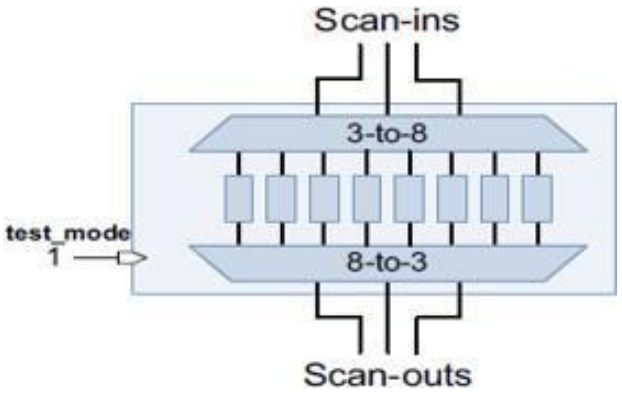


Figure 1.3 Scan Compression Mode

1.5 Problem Statement

As technology scales and the number of gates increases, the challenge of achieving high fault coverage in a limited amount of testing time and with acceptable costs becomes more difficult. For a complex System-on-Chip (SoC), testing all the internal nodes directly by the primary inputs and outputs is impractical. The problem of testing such a design is how to find high-quality test patterns for maximum fault coverage and minimum test cost, i.e., test time, test application power, and extra circuitry. This thesis addresses test pattern generation and simulation methods that take advantage of DFT features to improve fault coverage.

1.6 Thesis Organization

This thesis is structured into five chapters. **Chapter 1** provides an introduction to VLSI testing and Design for Testability. **Chapter 2** presents a detailed literature review of fundamental and advanced DFT techniques. **Chapter 3** describes the proposed methodology for ATPG and simulation. **Chapter 4** outlines the anticipated results and discussion. Finally, **Chapter 5** concludes the thesis and suggests directions for future work.

CHAPTER 2

LITERATURE REVIEW

This chapter focuses on the fundamental concepts, along with relevant research studies and methodologies, related to DFT flow, the challenges associated with DFT, and the scan insertion process. It also presents an overview of the key concepts and approaches used in ATPG.

2.1 DFT Flow

The first step in the DFT (Design for Testability) flow is RTL design, which involves designing DFT-specific structures and verifying their functionality. The next step is synthesizing the design for the first time to obtain a netlist for validation. The design then goes through the scan insertion process to produce the netlist for the next step [2]. In this stage, the SDF files and libraries from the Back-End team are required. In other words, the process of scan synthesis consists of reading the RTL, synthesizing the design, retesting it, implementing the scan chains, and re-evaluating the modified design. Then, ATPG (Automatic Test Pattern Generation) takes place, during which test patterns are generated. These patterns are then shifted into the scan chains, and simulations are performed to determine their effectiveness [3].

The simulations can either be performed as zero-delay (untimed) simulations or timed simulations using Standard Delay Format (SDF) with worst-case or best-case timing. The DFT design flow is shown in Figure 2.1.

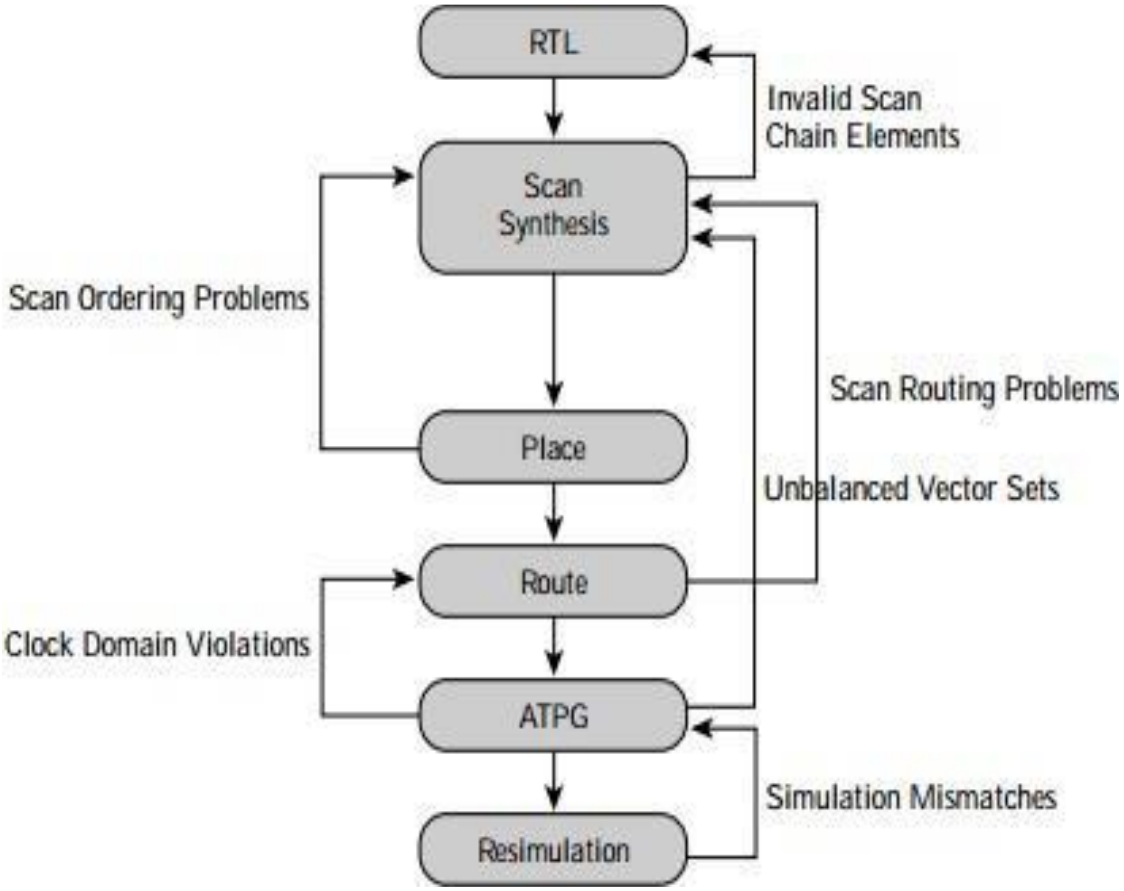


Figure 2.1 DFT Design Flow

2.2 DFT Challenges

With the rapid development of modern technology, the size of transistors in integrated circuits has been greatly reduced. As a consequence, the overall size of integrated circuits has been reduced, while the level of integration has increased, thus increasing the difficulty of testing and debugging circuits. Therefore, it is particularly important to select an appropriate optimal test solution.

2.2.1 Test Time: As the feature size decreases, the number of defects increases at smaller technology nodes. Thus, due to the limitations of test equipment, DFT engineers are responsible for finding design flaws rapidly and efficiently.

2.2.2 Pattern Count: To achieve the required high fault coverage, more test patterns have to be applied, which increases the volume of information to be stored on the tester's memory and I/O pins. This is a problem for DFT engineers because reducing the number of test patterns is critical due to the limited amount of memory and pin count of the tester. Moreover, it reduces the overall test application time [4].

2.2.3 Test Coverage: It is calculated as the ratio of the number of faults actually found to the total number of faults that could potentially be found in the design. [5].

2.2.4 Area: In order to make the design testable, Design for Testability (DFT) adds some additional circuitry to the design under test. This additional logic facilitates testing at different levels of integrated circuit design, but it also increases the overall chip area. The number of gates is raised due to DFT, and the area consumed by it should not, as a rule, exceed 5 percent of the design area.

2.2.5 Power Consumption: Power is one of the most crucial aspects of a circuit design. All design decisions are made with respect to the power requirements. Since power dissipation is related to switching activity, the higher the switching frequency, the higher the power dissipation which may damage the chip under test. [6].

2.2.6 Pin Count: As the device features get smaller and smaller the number of pins for testing purposes reduces. This poses a major challenge for engineers who have to develop means for thorough testing with lesser number of accessible points. [7].

2.3 Scan Insertion flow

Scan insertion is the first and the simplest step in Design for Testability, during which all the flip-flops and latches are replaced by special scan flip-flops and scan latches, respectively. These devices are referred to as scan cells, and they are connected together to form chains, also called scan chains. [8] It is too simplistic to describe the process by these few words, because practice shows that it is far from being trivial. Moreover, the quality of test directly depends on how correctly this step is performed. There are two ways of scan insertion, namely, Full Scan Design and Partial Scan Design. The former implies that all the flip-flops in a circuit are replaced by scan cells, while the latter implies only a certain number of them.

The partial scan method requires less chip area and provides less test coverage than a full scan design. However, this technique is cheaper and faster to implement than a full scan. In addition, due to the reduced number of test patterns, it also requires less memory. The flow of the scan insertion process is shown in Figure 2.2.

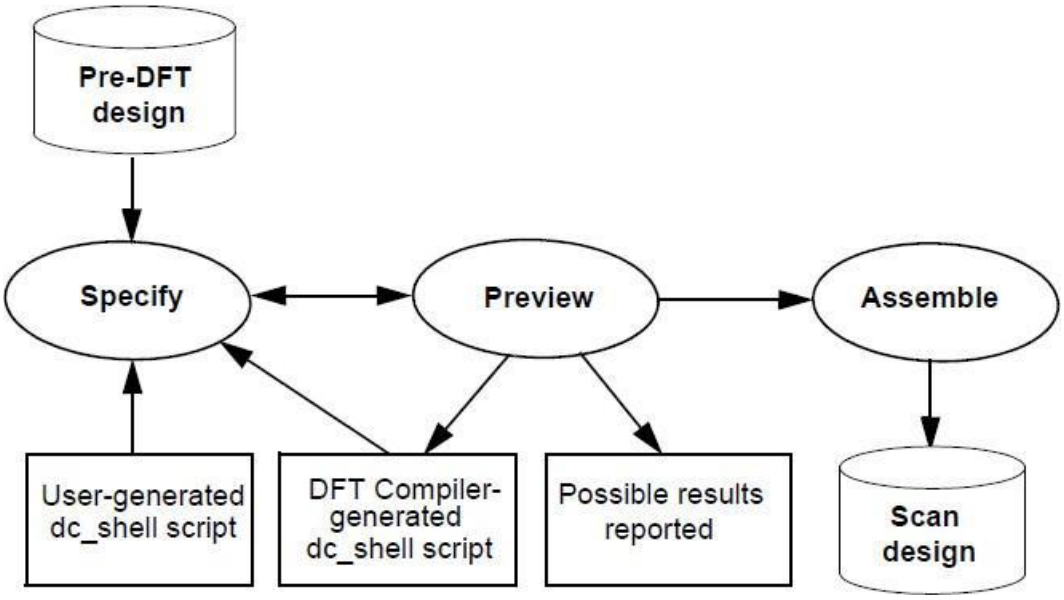


Figure 2.2 Scan Insertion Flow

2.4 ATPG Introduction

ATPG stands for Automatic Test Pattern Generation, a common practice in the Design for Test (DFT) community that aims to produce input patterns capable of detecting faults created during the fabrication process [9]. Typically, the effectiveness of ATPG tools is measured by their ability to identify a certain number of pre-defi ned faults and the overall number of patterns they produce. Those patterns are applied to the device under test by means of Automated Test Equipment (ATE). This tool checks if the circuit works correctly by comparing the measured response of the circuit under test to the expected correct response also encoded in the test pattern.

2.4.1 ATPG Technique

The ATPG process involves in general a three-step procedure called the Path Sensitization procedure. Those steps may be described as following:

- **Fault Sensitization**: This step is characterized by the assignment of 1s and 0s to circuit inputs to produce a 0 (or 1) at a fault site, opposite to the faulty value.
- **Fault Propagation**: The second step requires to assign 0s and 1s to the circuit inputs in such a way that the signal from the fault site is propagated through the circuit to some primary output.
- **Justification:** The third step is to assign values to the primary inputs so that the values used to propagate the fault are justified.

2.4.2 ATPG Process

The primary objective of ATPG is to generate the set of tests that achieve a specified level of fault coverage as defined by the following formula:

ATPG generally consists of two steps:

1) Test pattern generation step

2) Fault simulation step, which is used to evaluate the level of fault coverage and determine the set of detected faults.

There are generally three basic approaches that are used in test pattern generation [3]:

a) **Random Test Pattern Generation:** The tool under consideration has the ability to generate random patterns and save only those that identify a given fault. With this method, the process of testing takes place until such a set of patterns is obtained that does not reveal new flaws. One of the main disadvantages of this approach is the impossibility of completely replacing deterministic methods since such a method cannot detect some of the faults with a low probability of detection. However, this method is used in the early stages of pattern generation.

b) **Deterministic Test Pattern Generation:** This method is used when generating patterns for specific flaws. By selecting individual fault items from the list of faults, the user selects individual test patterns that must be used to detect them. Using the path sensitization method, the tool checks whether the chosen pattern detects the specific fault. In case of no detection, the fault is marked as untestable because it is redundant; otherwise, it is marked as detected.

c) **External Test Pattern Generation:** This method is used when the user has an externally generated set of test patterns. The tool checks which of these patterns will detect the given list of faults; therefore, using such patterns allows the user to achieve higher fault coverage for the design.

2.5 Research Gaps in DFT

Despite the significant advancements in DFT, several challenges remain. Key research gaps identified include the need for more efficient scan compression algorithms, automated test point insertion strategies, and ATPG tools that can effectively target new and emerging fault models without causing an explosion in test time and cost.

CHAPTER 3

PROPOSED METHODOLOGY

Moore's law dictates that the feature size of integrated circuits will continue to decrease, which ultimately makes perfect chips impossible. This necessitates extensive testing of chips to ensure that they are free of defects. The main criteria for testing are test cost, test quality, and test time.

Testing design for testability (design for testing) is a group of design methods that contain special structures to improve the testability of integrated circuits. The addition of these structures allows for faster and simpler testing during design and manufacturing. The manufacturing test is critical because it ensures that the finished product is free from manufacturing faults that could affect its performance in downstream applications. The most important design for testing technique is automatic test pattern generation (ATPG) simulation, which checks whether outputs responses match expected results after specific input patterns have been applied.

It has long been standard practice for ATPG to use logic gate abstractions for fault modeling. However, at the level of nanometers, new challenges arise in manufacturing tests. At the design verification stage, crosstalk noises and power supply variations can no longer be neglected and must be taken into account. Moreover, conventional methods of modeling and ATPG test pattern generation are being replaced by new approaches that take into account timing information, are scalable, and handle a wide range of operating conditions. Thus, just as design verification is encountering new obstacles, manufacturing testing is also being challenged. It becomes apparent that new fault models and ATPG techniques must be developed at the level of nanometers.

3.1 Overview of the Proposed Workflow

The methodology outlined adheres to a structured, industry-recognized process, starting from the initial RTL design and progressing through to a gate-level netlist ready for testing. This workflow is illustrated in Figure 3.1.

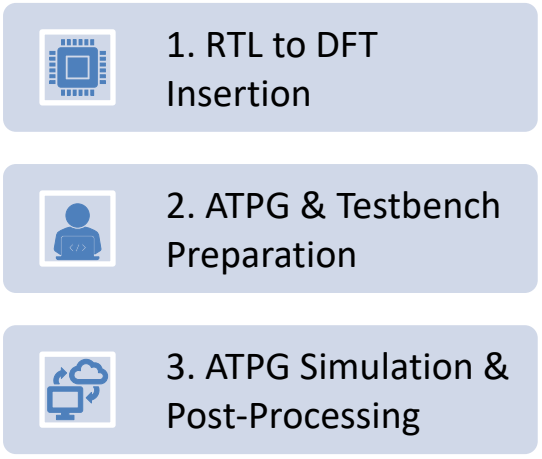


Figure 3.1: Proposed Workflow

Proposed Workflow from RTL to ATPG simulations the process is divided into following main stages:

1. RTL Design & Simulation
2. Synthesis
3. DFT Insertion and Analysis
4. ATPG pattern Generation
5. ATPG Simulations

The objective of DFT profiling is to ensure that the design under test can tolerate possible faults by generating a set of test patterns that targets the fault list with the highest fault coverage. After the pattern set is generated, which meets the desired coverage, the patterns are simulated. The following section describes the process step by step, starting with the RTL description of the design, followed by RTL synthesis to produce the netlist, which leads to the next step of scan chain insertion.

3.1.1. RTL Design and Simulation

The first step in the workflow is the creation of the digital circuit's functionality using a Hardware Description Language (HDL) i.e. Verilog. This involves writing the RTL code that describes the intended behavior of the circuit. Once the RTL code is written, it undergoes rigorous functional verification through simulation

3.1.2. Synthesis

After the RTL code has been functionally verified, the next stage is **synthesis**. In this process, the abstract RTL description is converted into a physical, technology-specific, gate-level netlist using a synthesis tool i.e. Cadence Xcelium. The tool maps the HDL code to a standard cell library provided by the semiconductor foundry.

Step 1.) **Synthesis of Netlist:** The RTL description of the design under test is synthesized to generate the corresponding gate-level netlist.

3.1.3 DFT Insertion and ATPG

This is the core stage where the design is made testable. The synthesized gate-level netlist is taken as input, and DFT structures are inserted into it. Key activities include Scan Chain Insertion, BIST Logic Insertion, and JTAG/Boundary Scan implementation. Once DFT insertion is complete, an **ATPG tool i.e. Synopsys TetraMAX** generates a compact set of test patterns which are simulated to determine fault coverage.

Step 2.) **Scan Insertion**: The scan insertion process utilizes a synthesized netlist in which the standard cells such as flip-flops and latches have been replaced by their testable counterparts named as scan flip-flops and scan latches respectively. In this process, the replaced cells are grouped into scan chains. Decisions on the scan inputs and outputs are taken and the estimated scan chain length is evaluated. The process of 'Scan or DFT insertion' consists of five steps which are listed below.

a) **Scan Configuration**: Configuring the number of scan chains along with the inclusion or exclusion of particular flip-flops from the scan chain.

b) **Scan Replacement**: Replacing of functional circuits such as flip-flops with their testable equivalents.

c) **Reordering**: Reordering of the logic gates taking into account the timing constraints of the circuit. Similarly, the scan cells are arranged in order to reduce the routing difficulties and

optimize the wire length. The step also helps in the arrangement of the functionally required circuits and scan chains.

d) **Scan Stitching**: Stitching the scan cells together in order to make complete scan chains.

e) **Scan Compression**: Scan Compression technique is adopted to minimize the test time by allowing parallel processing of test patterns. This is achieved by employing a compression architecture to partition the test patterns into groups of short chains instead of processing them individually.

Step 3.) **SPF Generation**: The scan inserted netlist is used to generate the SPF (STIL Procedure File) which stores all the relevant information about the circuit. STIL is basically a standard language used for test vectors representation and it is recommended by IEEE 1450 as a standard test language. STIL helps the automated test equipment (ATE) and computer aided engineering (CAE) tool providers to achieve standardization. TetraMAX ATPG utilizes a restricted set of STIL syntax to describe scan chains, clocking, constrained ports, stimuli and response data. When it comes to the usage of TetraMAX ATPG for pattern generation, it employs STIL for both design rule checking (DRC) and pattern formatting.

Step 4.) **Performing DRC:** Performing 'Design Rule Check' which checks the testability of the design. The ATPG tool performs a number of checks to ensure the correctness of scan architecture and its capability to support pattern generation and fault simulation. It checks the correct scan chain configuration and also ensures the presence of scan cells in the design. It also takes care of the DFT related clocking constraints and finally it moves the ATPG tool to Test Mode after successfully completing the DRC. It may be noted that the subsequent paragraphs will discuss general ATPG pattern generation flow as well as the inputs and outputs associated with it.

3.1.4 Test Pattern Generation

The present section provides a general description of the pattern generation flow that involves the Synopsys TetraMAX tool. Figure 3.1 below represents the general flowchart of the tool's usage. The pattern generation process involves several steps, namely: firstly, invoking the tool, secondly, including the design netlist and the DFT library and, thirdly, generating patterns and performing fault simulation.

1.) Firstly, the tool should be invoked. The tool's operation involves several modes and contexts that perform a variety of tasks. Each mode-context combination executes particular functions.

Usually, there are two basic contexts:

DFT: modification, edition, and introspection of the design files taking place, along with the process of scan insertion, scan compression, and scan analysis;
Pattern: test pattern generation and fault simulation are performed in this context;
The ATPG tool, in its turn, involves three basic modes that are combined with the contexts:
Setup: the mode gives access to the tool and allows setting the context and the design description;
Analysis: this mode is used for design analysis, pattern generation, and simulation;
Insertion: it allows performing design introspection and edition.

2.) Secondly, the design netlist and the DFT library should be included. All the design elements should have their definitions in the DFT library.

3.) Once the library file and netlist have been read in by the tool, it enters what is called setup mode. In this mode, a number of things can be done to setup the tool for test pattern generation. Things like define PI/PO, clocks, pins and cell constraints can be entered either manually or through a script. In addition, test procedure files must be provided for the tool to be able to perform shift and capture operations.

4.) Once the setup has been completed, the tool will leave setup mode and begin the processes necessary to begin the next mode, analysis mode. Before analysis mode can begin, the tool must flatten the design netlist. Flat design netlist is simply the netlist of the design using the internal primitives of the tool.

5.) Once the design netlist has been flattened, the tool will perform what is called a "Learning Analysis". The purpose of this is to allow the tool to analyze the design in an intelligent manner so as to allow for efficient fault simulation and test pattern generation with maximum fault coverage.

6.) At this point the tool will perform a Design Rule Checking (DRC). Once the netlists have been flattened and the design analysed, the tool will perform a Design Rule Checking to make sure that the design satisfies a number of design feasibility rules. Some of these rules if they are violated will cause the ATPG to fail due to errors. There are also some rules that if violated will produce only warnings that can be ignored.

7.) Assuming there were no errors or warnings that needed to be addressed, the tool then moves on to analysis mode. In this mode, simulation is performed on existing test patterns. Simulation includes both fault simulations which produce the output from the test patterns under fault conditions and good machine simulations which simulate the output of the design under test with no faults. Existing patterns can also be used or modified to increase the fault coverage.

8.) Once simulation is complete, the tool will prepare to begin test pattern generation. To do this, a fault list must be prepared for the tool to use. This fault list can be anything from a list of specific targets faults to be found to a list of all the faults that the tool can potentially target. Once this list has been prepared, test pattern generation can begin.

9.) Finally, the test patterns that have been generated can be exported from the tool either in Verilog format or binary or ASCII format

Tool inputs and outputs:

Figure 3.1 illustrates a simplified representation of the inputs to and outputs from the ATPG tool..

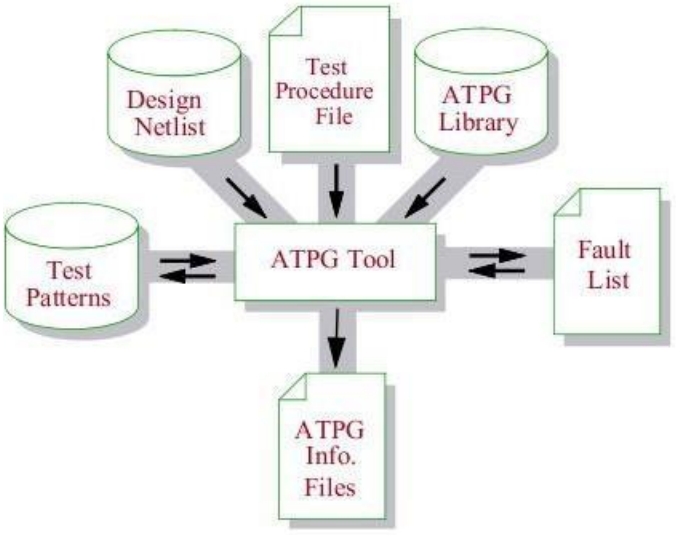


Figure 3.2 Inputs and output of ATPG Tool [3]

The tool requires the following inputs:

Design:

A gate-level netlist or flat representation of the design.

Test Procedure File:

This file describes the desired operation of the scan circuitry in the design under test. This file

is usually created after scan insertion, but it can also be manually-entered.

Library:

The design library contains cell information for the design. The library is used to create a flat, gate-level simulation model of the design for purposes of fault simulation and test pattern generation.

Fault List:

An optionally-supplied external fault list can be used to specify the faults of interest, together with their desired initial values. The tool uses the information in the fault list as a starting point for test pattern generation.

Test Patterns:

Externally-created test patterns can be supplied as input to the tool, for use in simulation.

The tool produces the following outputs:

Test Patterns:

A test bench plus test patterns.

ATPG Information File:

This file contains information specific to the ATPG, such as session log information, reports, DRC results, gate-level information, or other information.

Fault List:

An ASCII file containing internal fault information.

3.1.5 Pattern Simulation:

After the pattern has been successfully generated, it should be converted into the Verilog format so that it can be interpreted by the tester. This process is performed by Synopsys's stil2verilog utility, which produces three files: .v, .dat, psd.dat. The last file is generated only if the scan compression was used.

CHAPTER 4

Work done and Experimental Results

4.1 Steps for Simulation

The Cadence SimVision tool is used for simulation purposes. Several files are needed to set up a simulation, including a force file that defines the signals to be forced to specific values during the simulation. The simulation process has three phases: compilation, elaboration, and simulation.

4.1.1 RTL code :

A snippet of RTL code for b19 processor is shown in fig.4.1.

```
88          end else begin
89          rd <= 1'b0;
90          wr <= 1'b0;
91          case(state)
92              FETCH : begin
93                  MAR = reg3 % 2 ** 20;
94                  MBR = datai;
95                  addr <= MAR;
96                  rd <= 1'b1;
97
98                  IR <= MBR;
99                  state = EXEC;
100             end
101             EXEC : begin
102                 if(IR < 0) begin
103                     IR =  -IR;
104                 end
105                 mf = (IR / 2 ** 27) % 4;
106                 df = (IR / 2 ** 24) % 2 ** 3;
107                 ff = (IR / 2 ** 19) % 2 ** 4;
108                 cf = (IR / 2 ** 23) % 2;
109                 tail = IR % 2 ** 20;
110                 reg3 = (reg3 % 2 ** 29) + 8;
111                 s = (IR / 2 ** 29) % 4;
112                 case(s)
113                     ZERO : begin
114                         r = reg0;
115                     end
116                     ONE : begin
117                         r = reg1;
118                     end
```

Figure 4.1 RTL code for b19

4.2 Environment

Synopsys Design Compiler was used for synthesized netlist generation of the design under test, while Synopsys TetraMAX was used for test pattern generation. The generated test patterns were then simulated with Cadence Simvision (Xcelium 24.10.071) for timing checking purposes. All the tools were operated on the Linux server provided by STMicroelectronics.

The invoking of Synopsis of DC shell is shown in Fig.4.2

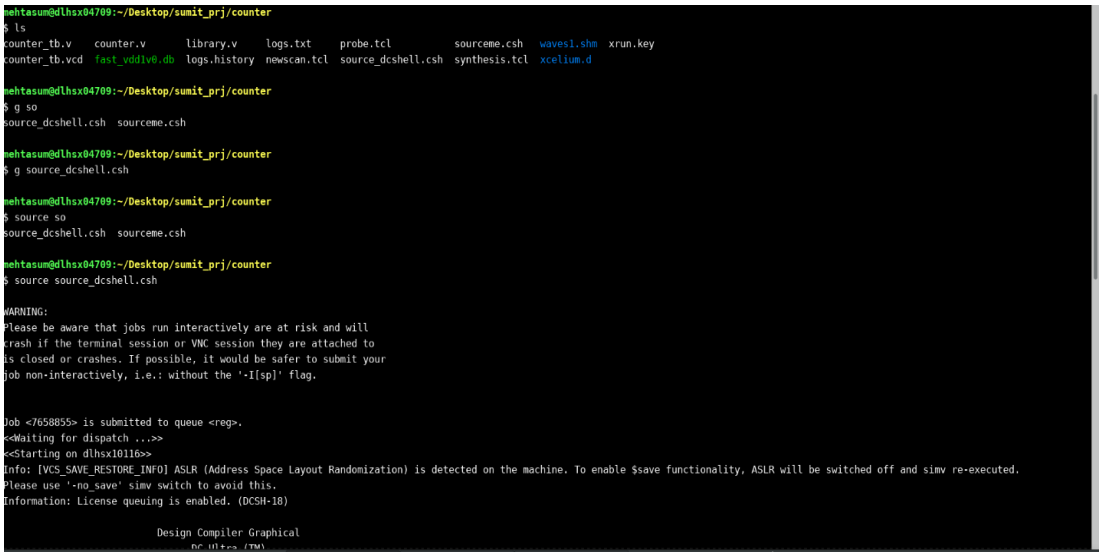


Figure 4.2 Invoking Synopsis DC shell

The output of the synthesis is shown in Fig.4.3.

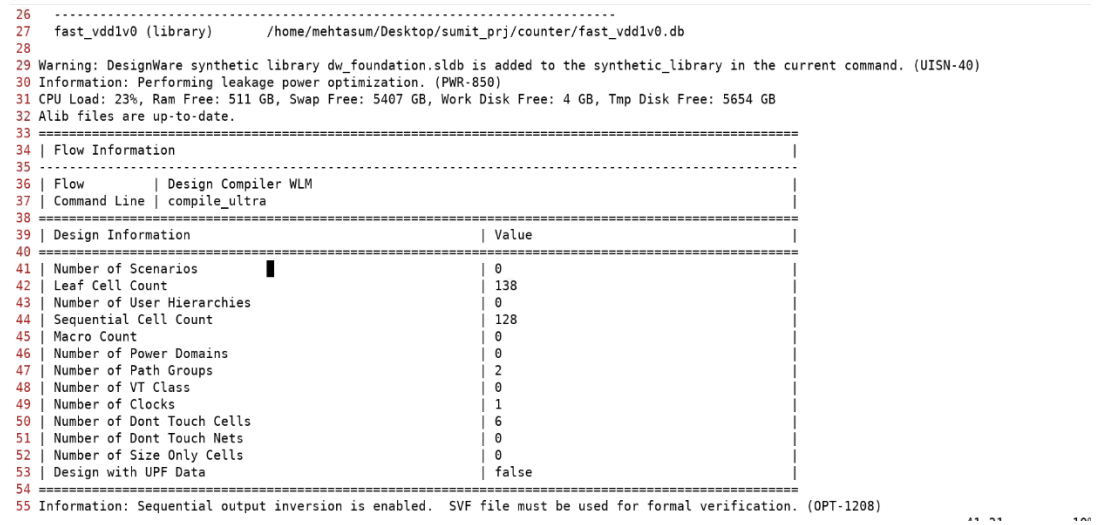


Figure 4.3 Synthesis Output

The various output files as a result of synthesis are shown in Fig.4.4.

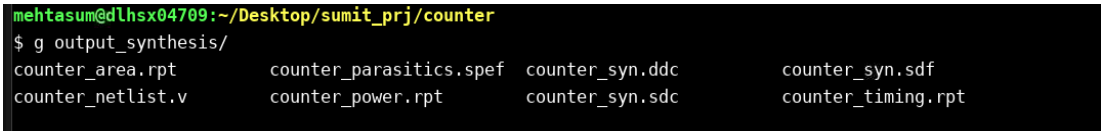


Figure 4.4 Output Files after synthesis

4.3 Scan Insertion

The following is the output files generated after scan insertion as shown in Fig.4.5.

```
mehtasum@dlhsx04741:/work/stellar_x8_dft/sumit/learning/b19/b19_tr_1/sumit/clg_1/REPORTS_Scan_compression
$ ls *80*
area_post_scan_80.rpt  dft_80.rpt                    power_post_scan_80.rpt  preview_dft_80.rpt              preview_test_points_80.rpt
area_pre_scan_80.rpt   dft_drc_pre_verbose_flow_80.rpt  power_pre_scan_80.rpt   preview_dft_test_wrappers_80.rpt  report_scan_path_scan_comp_80.rpt

mehtasum@dlhsx04741:/work/stellar_x8_dft/sumit/learning/b19/b19_tr_1/sumit/clg_1/REPORTS_Scan_compression
```

Figure 4.5 Output Files after scan insertion

The scan coverage and insertion reports are shown in Fig.4.6.

```
-----------------------------------------------------------------------------------------
 Sequential Cell Report:       Sequential       Core             Core
                               Cells            Segments         Segment Cells
-----------------------------------------------------------------------------------------
 Sequential Elements Detected:  6138               0                   0
 Clock Gating Cells           :  0
 Synchronizing Cells          :  0
 Non-Scan Elements            :  0
 Excluded Scan Elements       :  0                 0                   0
 Violated Scan Elements       :  0                 0                   0
 (Traced) Scan Elements       :  6138    (100.0%)  0       (  0.0%)    0       (  0.0%)
-----------------------------------------------------------------------------------------
```

T

```
Architecting scan compression mode ScanCompression_mode with base mode Internal_scan
Architecting Load Decompressor (version 5.8)
  Number of inputs/chains/internal modes = 7/77/4
Architecting Unload compressor (version 5.8)
  Number of outputs/chains = 7/77
  Information: Compressor will have 100% x-tolerance
Info: Completed Post Scan Architect Callbacks.

****************************************
Preview_dft report
For    : 'Insert_dft' command
Design : b19
Version: X-2025.06-SP3
Date   : Thu Feb 26 11:20:15 2026
****************************************



****************************************

Current mode: ScanCompression_mode
****************************************

Number of chains: 77
Scan methodology: full_scan
Scan style: multiplexed_flip_flop
Clock domain: mix_clocks_not_edges
Scan enable: ipt_hm_se (no hookup pin)
```

Figure 4.6 Scan coverage and Insertion report

The Scan Testmodes are shown in Fig.4.7.

```
Test Mode Controller Ports
--------------------------
test_mode: ipt_hm_atpg_wrpof_mode
test_mode: ipt_hm_atpg_wrpif_mode
test_mode: ipt_hm_atpg_mode
test_mode: ipt_hm_atpg_comp_mode

Test Mode Controller Index (MSB --> LSB)
----------------------------------------
ipt_hm_atpg_wrpof_mode, ipt_hm_atpg_wrpif_mode, ipt_hm_atpg_mode, ipt_hm_atpg_comp_mode

Control signal value - Test Mode
--------------------------------
0011 ScanCompression_mode - InternalTest

0110 Internal_scan - InternalTest

1
```

Figure 4.7 Scan Testmodes

The Post scan reports are shown in Fig.4.8.

```
mehtasum@dlhsx04741:/work/stellar_x8_dft/sumit/learning/b19/b19_tr_1/sumit/clg_1/REPORTS_Scan_compression
$ ls *80*
area_post_scan_80.rpt  dft_80.rpt                     power_post_scan_80.rpt  preview_dft_80.rpt              preview_test_points_80.rpt
area_pre_scan_80.rpt   dft_drc_pre_verbose_flow_80.rpt  power_pre_scan_80.rpt   preview_dft_test_wrappers_80.rpt  report_scan_path_scan_comp_80.rpt

mehtasum@dlhsx04741:/work/stellar_x8_dft/sumit/learning/b19/b19_tr_1/sumit/clg_1/REPORTS_Scan_compression
```

Figure 4.8 Post scan reports

4.4 Comparison of Pre-Scan and Post-Scan Design

The pre-scan schematics of the B19 processor are shown in Fig.4.9.

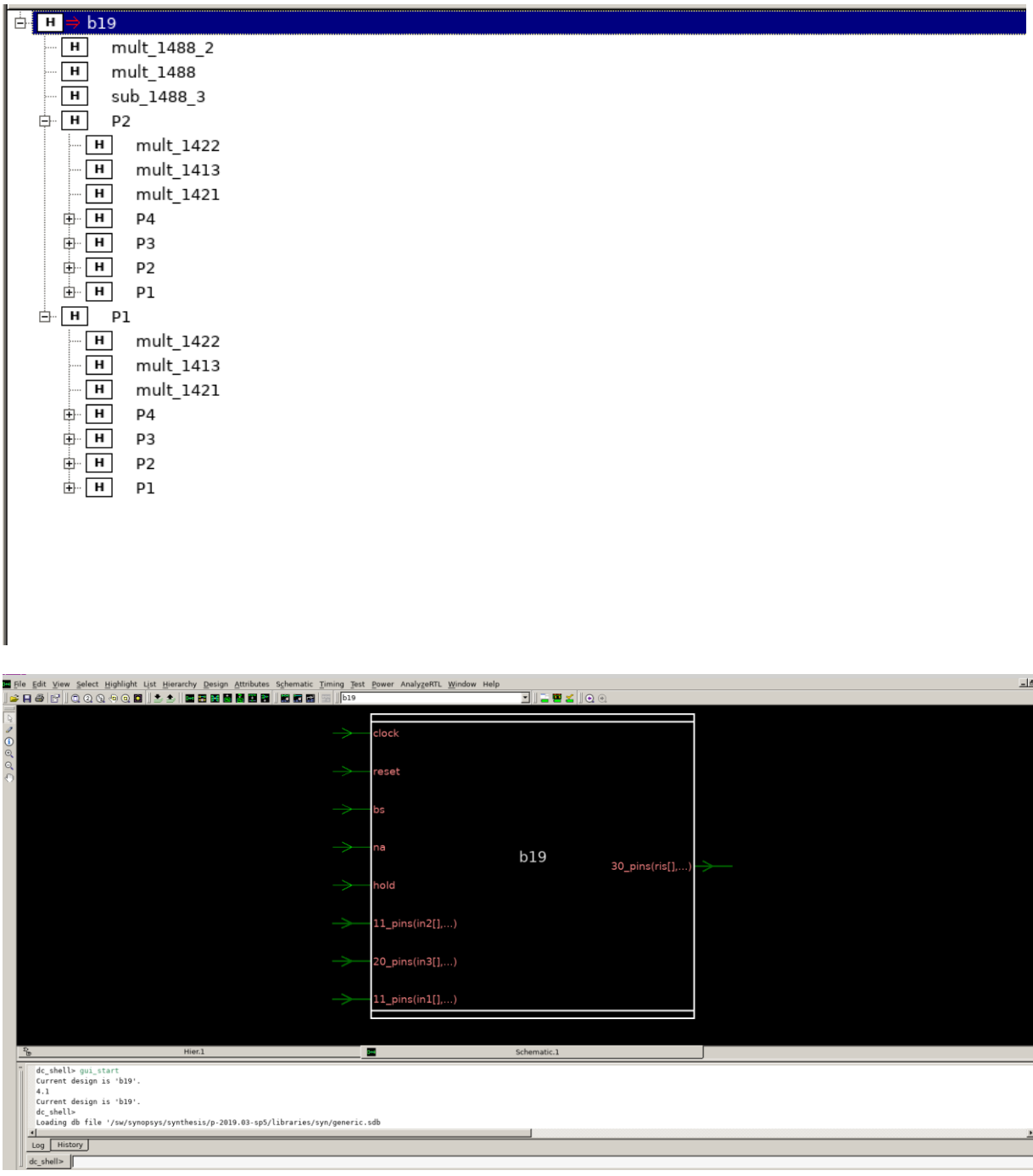


Figure 4.9 B19 schematics pre scan

The post-scan schematics of the B19 processor are shown in Fig.4.10.1 and Fig.4.10.2.

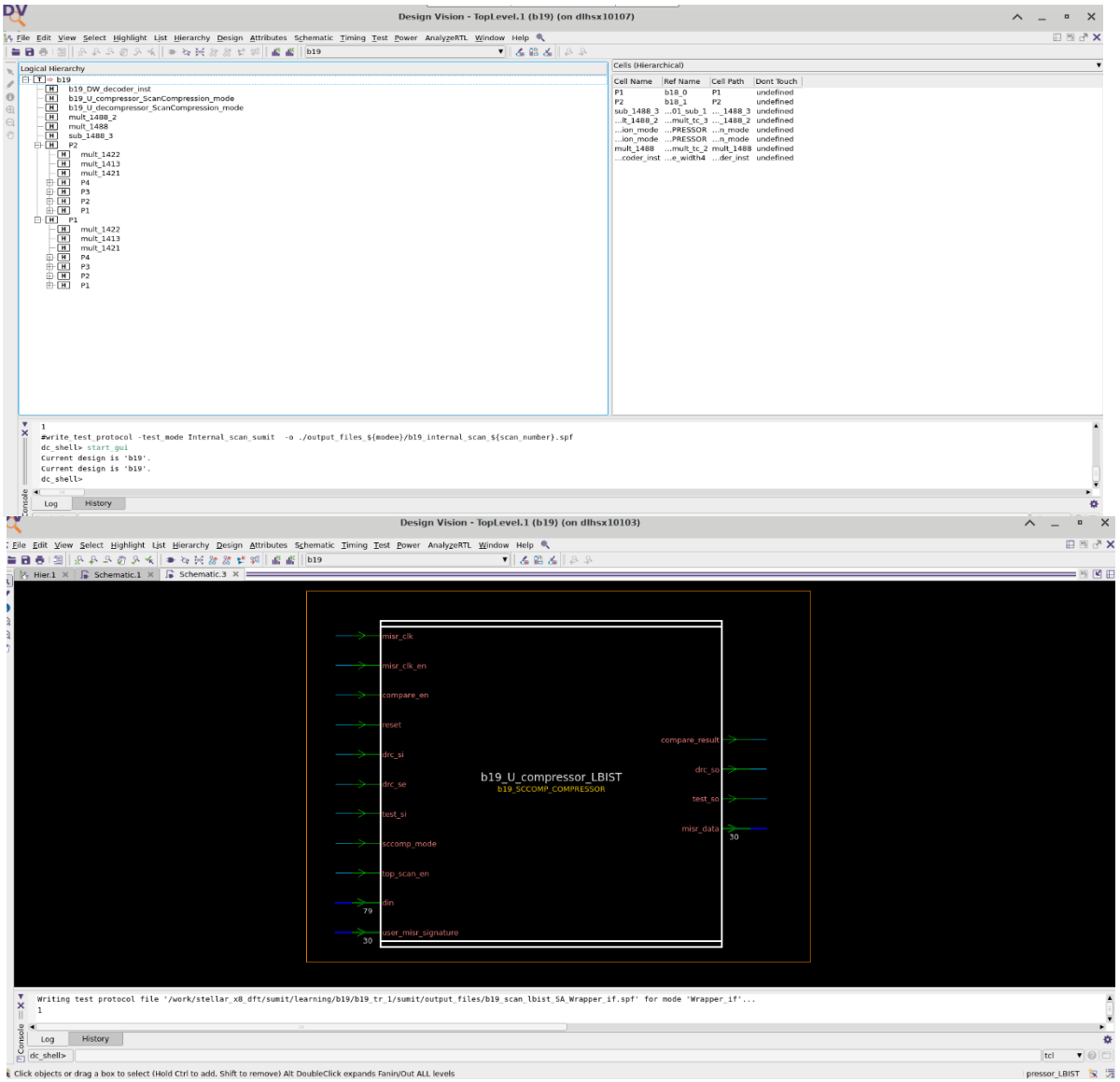


Figure 4.10.1 b19 schematic post scan

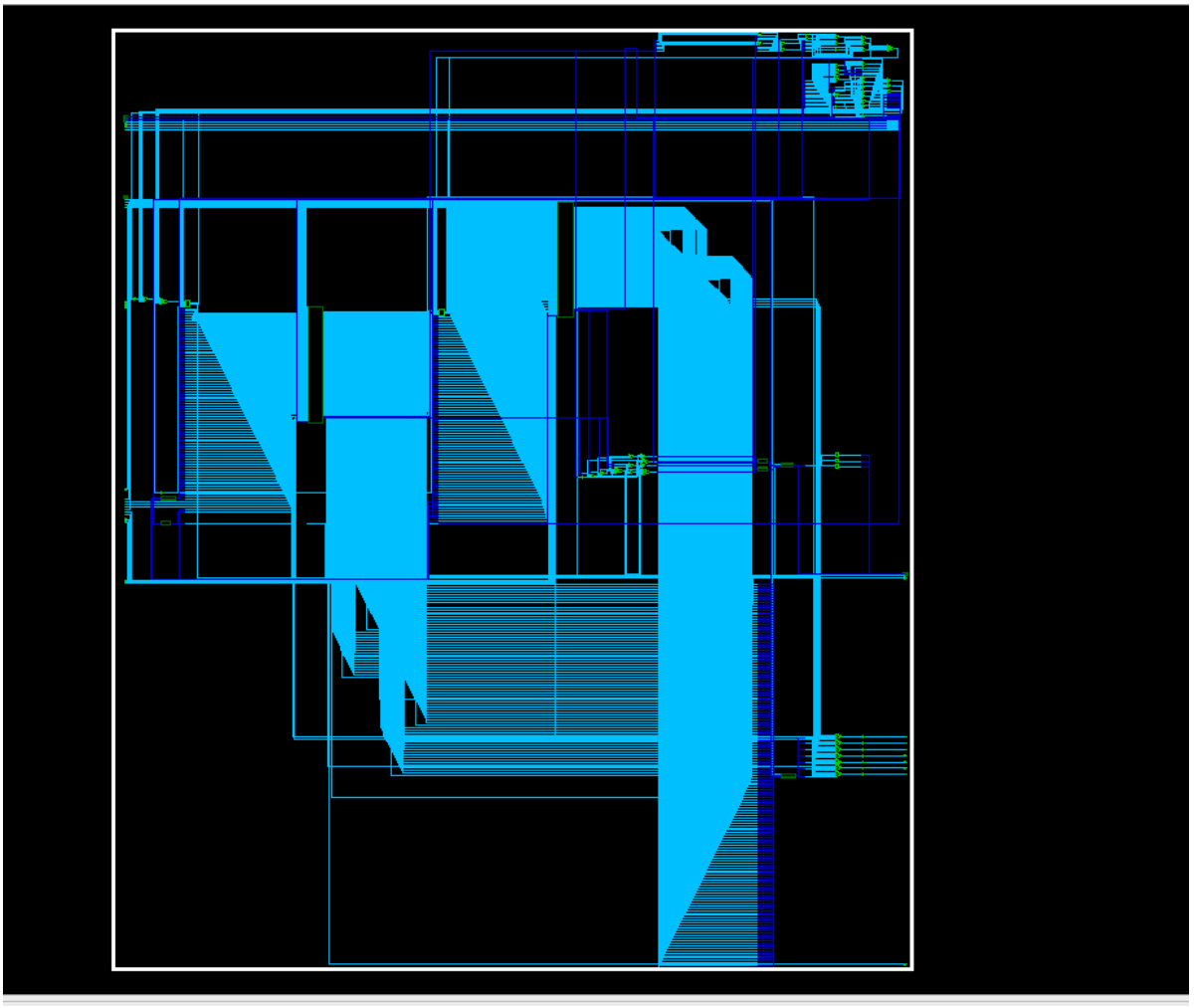


Figure 4.10.2 B19 schematic post scan

The schematic of Scan chain is depicted in the following fig.4.11

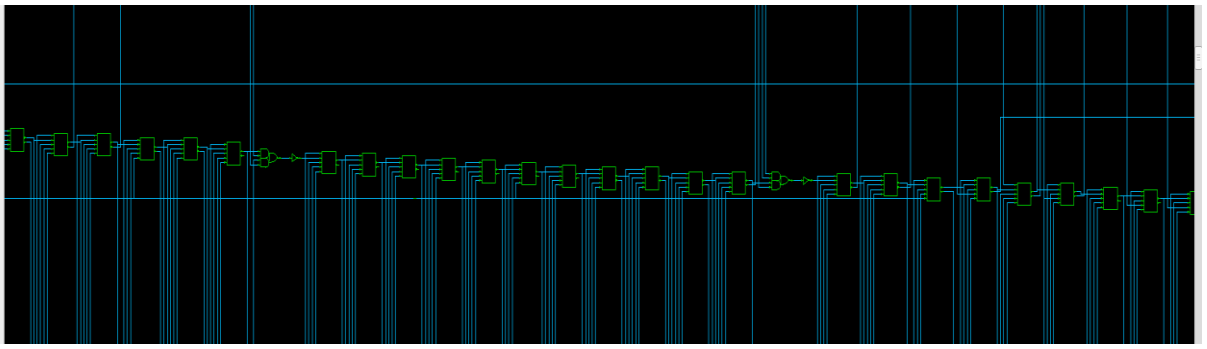


Figure 4.11 Scan chain with variable length of flip flops (here 77)

The Timing pre scan and post scan reports are shown in Fig.4.12 and Fig.4.13 respectively.

```
clock clk (rise edge)                    10.00      10.00
clock network delay (ideal)               0.00      10.00
count_reg_127_/CK (DFFQXL)                0.00      10.00 r
library setup time                       -0.02       9.98
data required time                                   9.98
-----------------------------------------------------------
data required time                                   9.98
data arrival time                                   -2.83
-----------------------------------------------------------
slack (MET)                                          7.15
```

Figure 4.12 Timing pre scan

```
clock clk (rise edge)                   10.00      10.00
clock network delay (ideal)              0.00      10.00
count_reg_127_/CK (SDFFQX1)              0.00      10.00 r
library setup time                      -0.04       9.96
data required time                                  9.96
-----------------------------------------------------------
data required time                                  9.96
data arrival time                                  -2.83
-----------------------------------------------------------
slack (MET)                                         7.13
```

Figure 4.13 Timing post scan

The net interconnect area reports of pre scan and post scan designs are shown in Fig.4.14 and Fig.4.15 respectively.

```
-----------------------------------
Hierarchical Cell Count:        187
Hierarchical Port Count:      16069
Leaf Cell Count:              76571
Buf/Inv Cell Count:           11420
Buf Cell Count:                   0
Inv Cell Count:               11420
CT Buf/Inv Cell Count:            0
Combinational Cell Count:     70433
Sequential Cell Count:         6138
Macro Count:                      0
-----------------------------------


Area
-----------------------------------
Combinational Area:   137095.000000
Noncombinational Area: 61380.000000
Buf/Inv Area:          11420.000000
Total Buffer Area:             0.00
Total Inverter Area:       11420.00
Macro/Black Box Area:      0.000000
Net Area:                  0.000000
-----------------------------------
Cell Area:            198475.000000
Design Area:          198475.000000
```

Figure 4.14 Net interconnect area pre scan

```
Cell Count
----------------------------------
Hierarchical Cell Count:        190
Hierarchical Port Count:      16837
Leaf Cell Count:              77319
Buf/Inv Cell Count:           11618
Buf Cell Count:                  31
Inv Cell Count:               11618
CT Buf/Inv Cell Count:            0
Combinational Cell Count:     71181
Sequential Cell Count:         6138
Macro Count:                      0
----------------------------------


Area
----------------------------------
Combinational Area:   138710.000000
Noncombinational Area: 61380.000000
Buf/Inv Area:          11618.000000
Total Buffer Area:            31.00
Total Inverter Area:       11618.00
Macro/Black Box Area:      0.000000
Net Area:                  0.000000
----------------------------------
Cell Area:            200090.000000
Design Area:          200090.000000
```

Figure 4.15. Net interconnect area post scan

The pre scan and post scan dynamic power reports are shown in Fig.4.16 and Fig.4.17 respectively.

```
  Cell Internal Power  =   0.0000 nW    (0%)
  Net Switching Power  =   4.3945 mW  (100%)
                         ---------
Total Dynamic Power    =   4.3945 mW  (100%)
```

Figure 4.16 Pre Scan dynamic Power

```
Attributes
----------
i - Including register clock pin internal power


  Cell Internal Power  =   0.0000 nW    (0%)
  Net Switching Power  =   4.7148 mW  (100%)
                         ---------
Total Dynamic Power    =   4.7148 mW  (100%)
```

Figure 4.17 Post Scan dynamic Power

Below is a brief comparison shown of pre-scan and post-scan design for b19.

Table 4.1 Comparison of Pre-scan and Post Scan design

Metric	Pre-Scan	Post-Scan	difference
Total Cells	76571	77319	+0.9% (scan overhead)
Chain Length	N/A	77	N/A
Max Delay (ns)	2.5	2.7	+0.2

4.5 ATPG Pattern Generation:

Below is the Test pattern generation of b19 design with Transition Faults with 80 maximum chain length and 7 scan Ins and 7 Scan outs.

The Test pattern generation window of b19 design is shown in Fig.4.18.

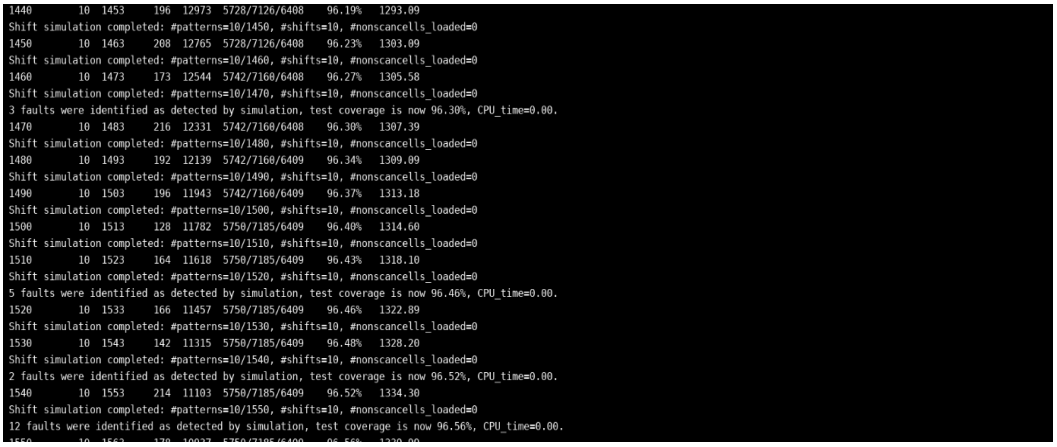


Figure 4.18 Test pattern generation of B19

We observed that we are getting S1 errors during the DRC phase of the pattern generation.

"S1" errors in Synopsys test pattern generation (ATPG) typically refer to Scan DRC violations (Design Rule Check) related to scan cell controllability or scan chain configuration. These are common in TestMAX ATPG or TetraMAX when generating patterns after DFT insertion.

S1 is specifically the rule code for "Scan cell not controllable" or "Invalid scan cell in chain" — meaning some scan flops cannot be loaded/shifted properly, leading to low coverage or pattern generation failures.[3]

The DRC violations are shown in Fig.4.19.

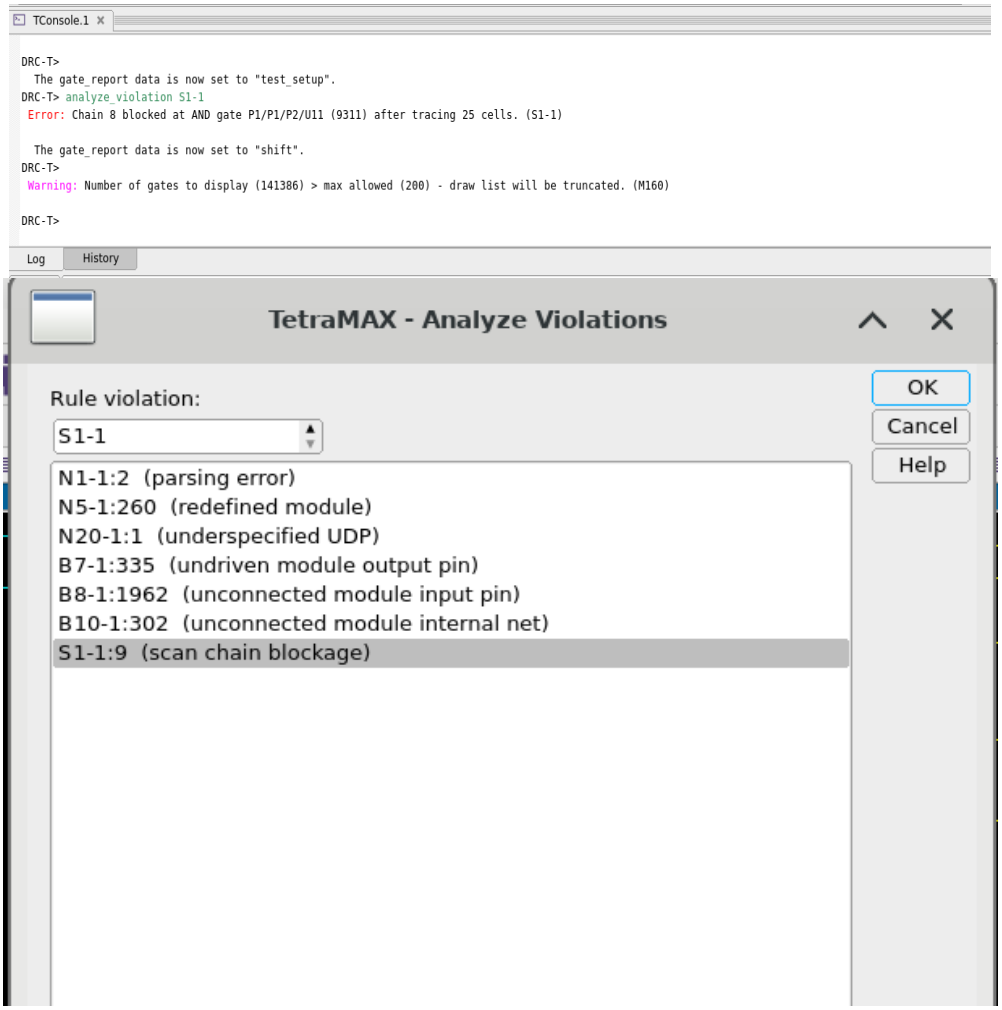


Figure 4.19 DRC violations during the Test pattern generation

Why S1 Errors Occur

- Scan cells (flops) are not fully controllable from scan inputs (e.g., missing scan enable, clock issues, or hierarchy problems).
- During ATPG, the tool detects this and marks faults as untestable, causing errors or warnings like "S1 violation in chain X".
- Common triggers:
    - Incomplete scan replacement (some flops not converted to scan flops).
    - Clock skew or missing clock during shift.
    - Wrapper isolation blocking chains in hierarchical designs.

So after the debugging is done we managed to trace back the chains and found the cause of the S1 errors and managed to add the constraints in order to resolve the errors.

The chain blockages in TetraMAX gui are shown in fig.4.20 and fig.4.21 respectively.

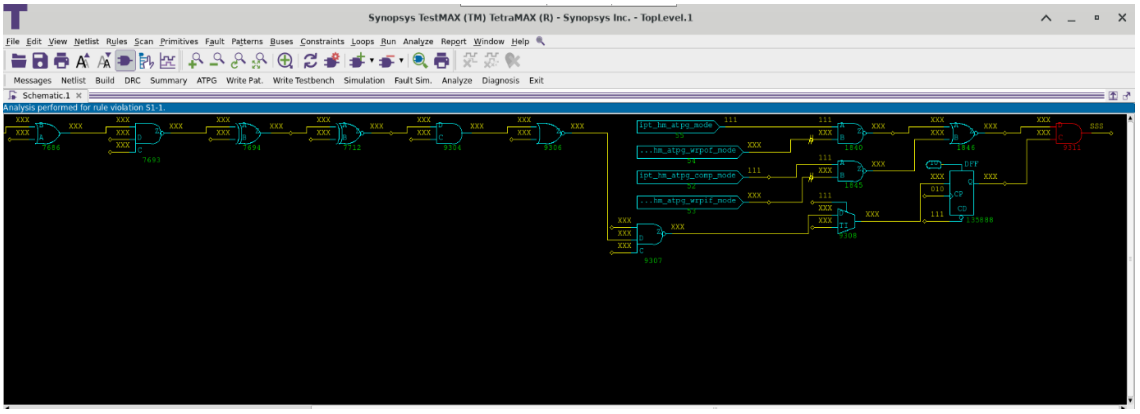


Figure 4.20 TetraMAX gui view of the chain blockages

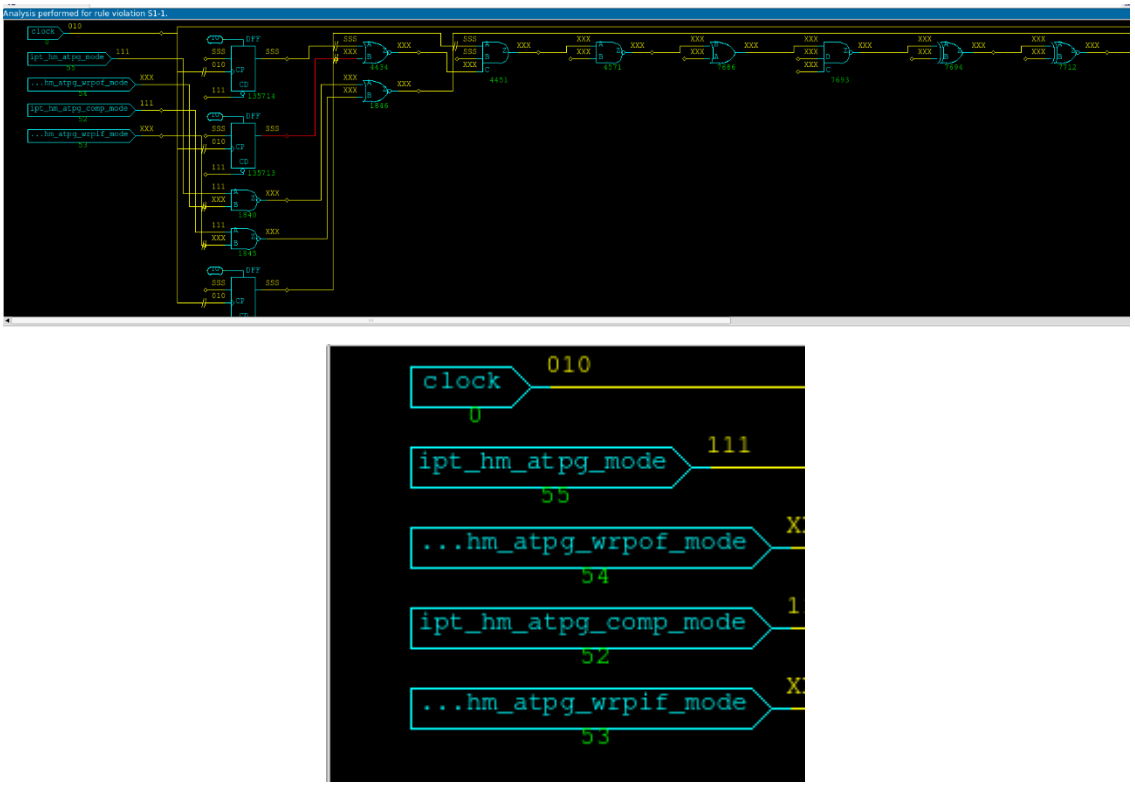


Figure 4.21 Cause of the chain blockage

The ATPG constraints are shown in Fig.4.22.

```
add_pi_constraint 0 reset
add_pi_constraints 0 { ipt_hm_atpg_wrpif_mode  ipt_hm_atpg_wrpof_mode }
```

Figure 4.22 ATPG Constraints

The constraints are added and hence the test patterns are generated successfully.

4.6 Combined Results and analysis of Scan Insertion and Pattern Generation:

The analysis is done in Scan Compression Mode with base mode as Internal Scan mode.

4.6.1 For SA(stuck-at) Faults:

The Analysis of B19 for SA faults is shown is tabular format in Table 4.2.

Table 4.2 Analysis of B19 for Stuck-at Faults

no.of SI/SO	10 si /10 so	7 si / 7 so
max length of chain	80	80
no. of scan chains	77	77
pre scan power Pdyn	4.3945	4.3945 mW
post scan power Pdyn	4.7500 mW	4.7148 mW
power % increase	8.08%	7.28%
ATPG coverage	98.48%	98.47%
faults detected / total	586439/606792	585530 / 605884
patterns	637	763
test time (test cycles * time period)	5.295ms	6.3413ms

The SA ATPG coverage comparison is shown in Fig. 4.23.

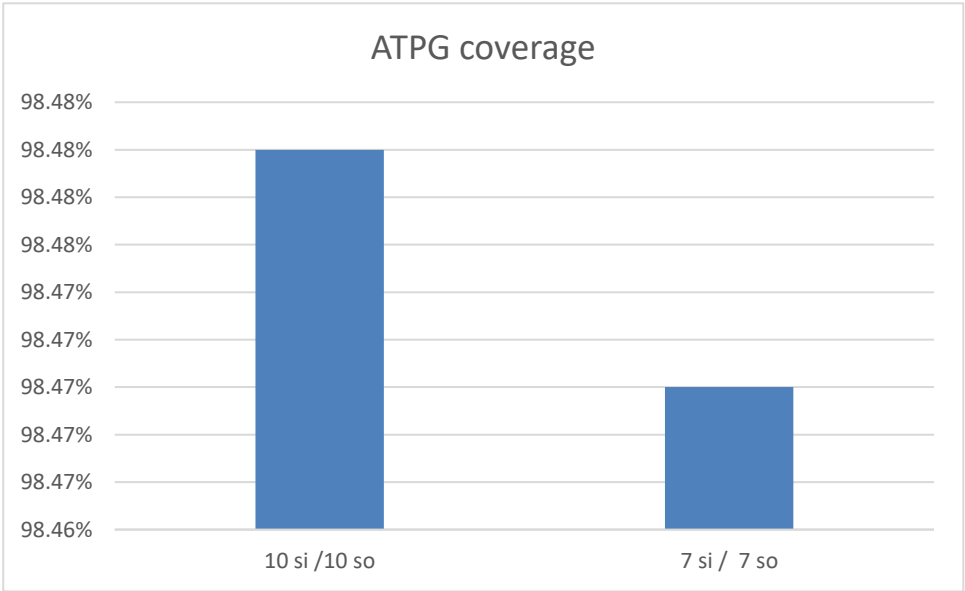


Figure 4.23 SA ATPG coverage comparison

The SA ATPG patterns comparison is shown in Fig. 4.24.

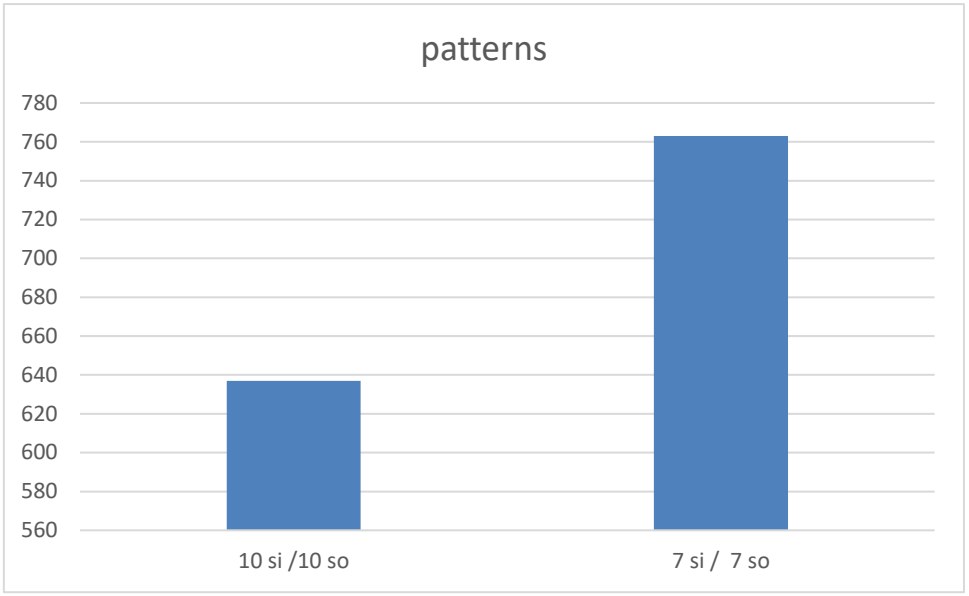
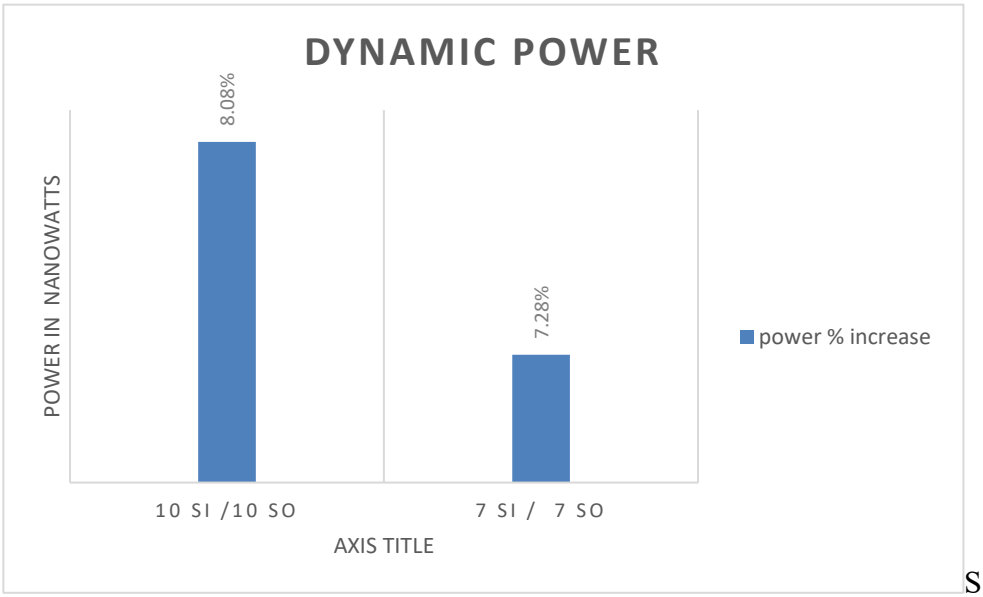


Figure 4.24 SA ATPG patterns generated

The SA ATPG dynamic power increase is shown in Fig. 4.25.



S

Figure 4.25 SA ATPG dynamic power increase

The SA Test time with different SI/SO is shown in Fig. 4.26.

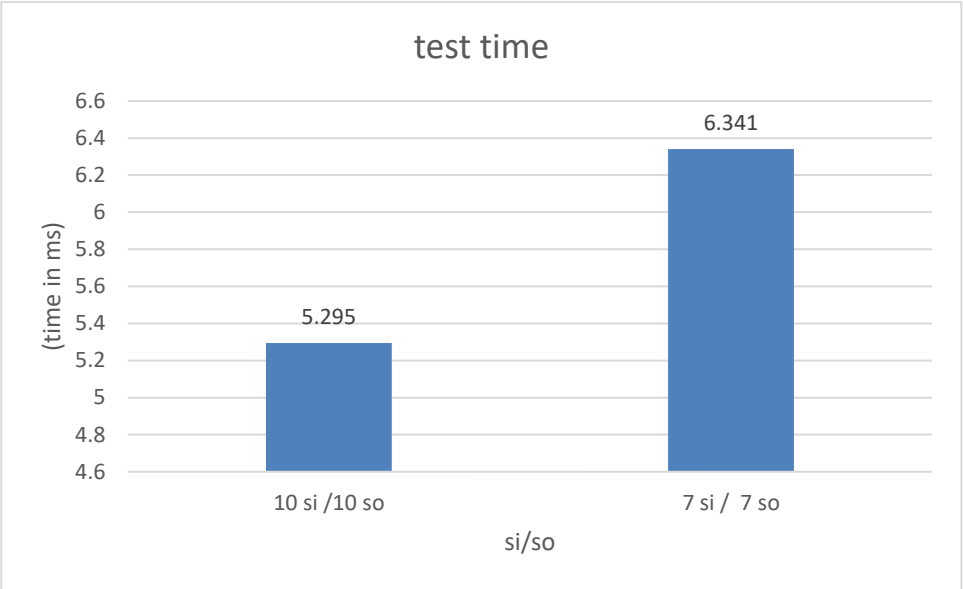


Figure 4.26 SA Test Time with different SI/SO

The above results show that :

1.) The increase in no. of Scan ins and scan out lead to increase in dynamic Power consumption

2.) The increase in no. of Scan ins and scan out lead to can also lead to increase in Test coverage

3.) Lower scan ins and outs can cause the test patterns to increase which takes a lot more time .

4.) The Test time which is most important in DFT decreases with increases Scan ins and Scan out but you have to keep in mind the dynamic power increase.

4.6.2 For Transition Faults:

The Analysis of B19 for transition faults is depicted in tabular form in table.4.3.

Table 4.3 Analysis of B19 for Transition Faults

max length-si-so	80 - 10 - 10	80-7-7	40-10-10	40 -7-7
no.of SI/SO	10 si /10 so	7 si / 7 so	10 si /10 so	7 si / 7 so
max length of chain	80	80	40	40
scan chain	77	77	154	154
pre scan power Pdyn	4.3945	4.3945 mW	4.3945 mW	4.3945 mW
post scan power Pdyn	4.7500 mW	4.7148 mW	4.8466 mW	4.8082 mW
power % increase	8.09%	7.20%	10.20%	9.40%
ATPG coverage	98.06%	98.06%	98.05%	98.04%
fault det/ total	539339/562912	538438/562004	544123/567872	541669/565380
patterns	1297	1551	1962	2597
test time (test cycles * time period)	10.902ms	10.903ms	8.637ms	11.429ms

The TF ATPG coverage comparison is shown in Fig. 4.27.

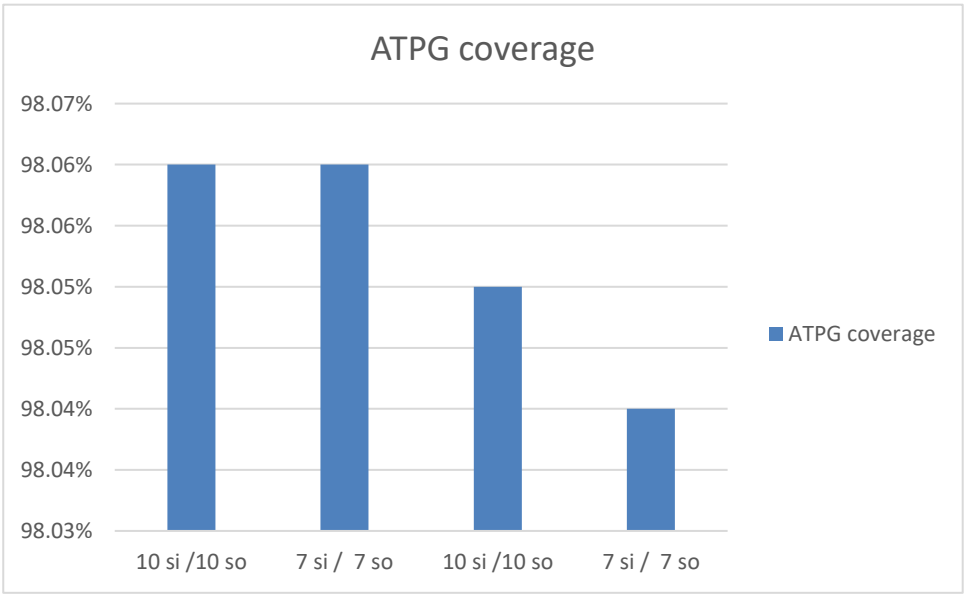


Figure 4.27 TF ATPG coverage comparison

The TF ATPG patterns comparison is shown in Fig. 4.28.

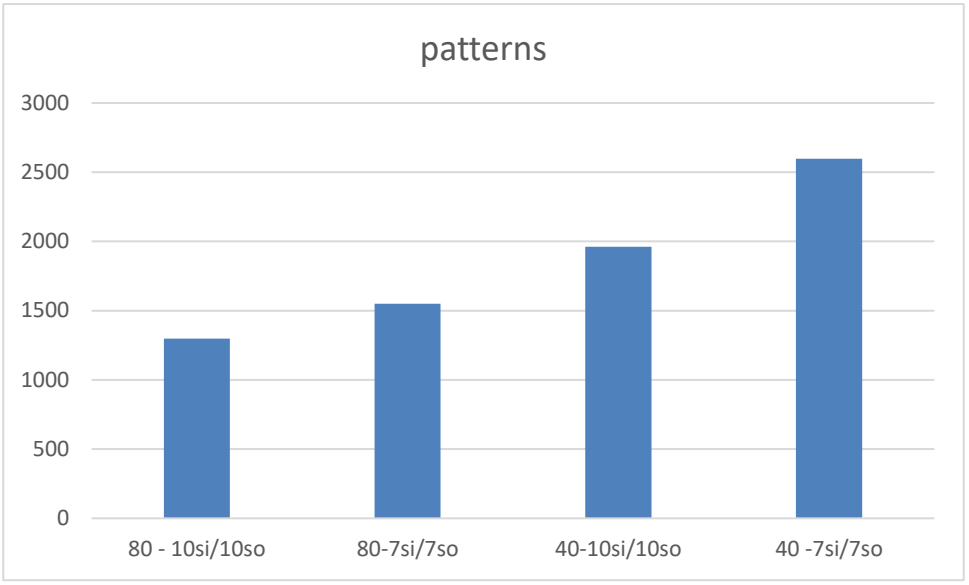


Figure 4.28 TF ATPG patterns generated

The TF ATPG increase in dynamic power is shown in Fig. 4.29.

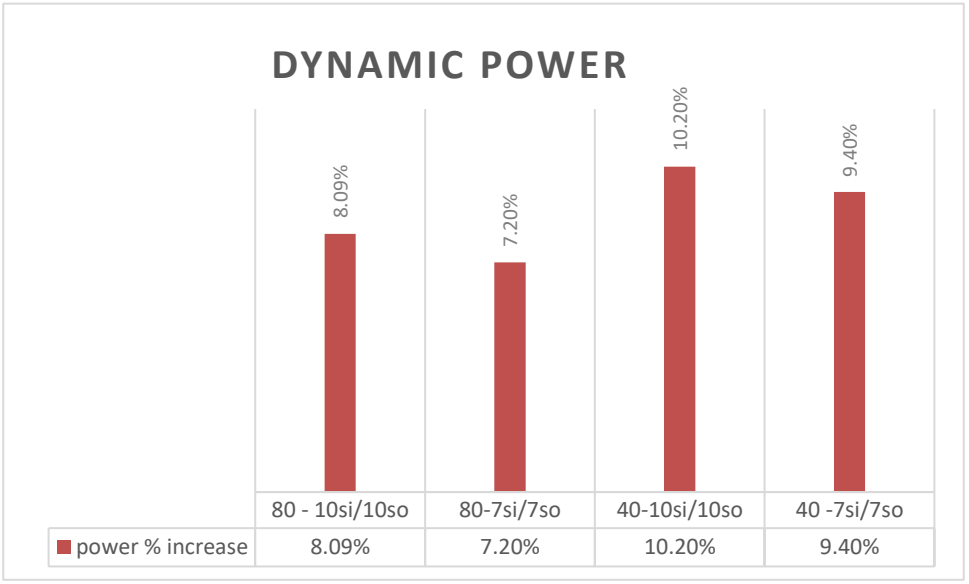


Figure 4.29 TF ATPG dynamic power increase

The TF test time comparison with different chain length and SI/SO is shown in Fig. 4.30.

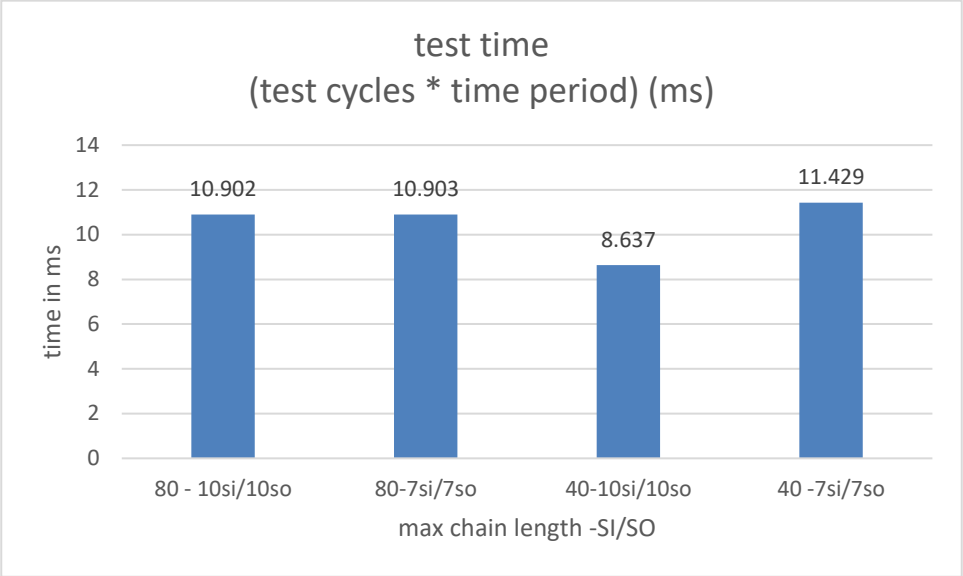


Figure 4.30 TF Test Time with different chain length and SI/SO

The above results show that :

1.) The increase in no. of Scan ins and scan out lead to increase in dynamic Power consumption

2.) The increase in no. of Scan ins and scan out lead to can also lead to increase in Test coverage

3.) Lower scan ins and outs can cause the test patterns to increase which takes a lot more pattern generation time.

4.) The Test time which is most important in DFT decreases with increases Scan ins and Scan out but you have to keep in mind the area and power increase.

5.) Reduction in chain length leads to increase in the no. of scan chains.

6.) The best and lowest Test time is in the case of 40 max length of Scan chain and 10 Scan ins and Scan outs. This means that the best test time comes in the condition where the scan ins and outs are more and the length of scan chains is less and the no. of scan chains are more. However a drawback of this configuration would be the maximum area usage and maximum dynamic power usage.

4.7 Pattern Simulation:

The patterns are generated in a .STIL(Standard Test Interface Language) format.

We then convert file to .STIL zip format and then give a command to generate Testbench/Patterns i.e. (.v, .psd and .dat files) which are then used in the ATPG simulations to check the correctness of the patterns generated.

Below is the window that converts .STIL file to Testbench/ Patterns in fig.4.31.

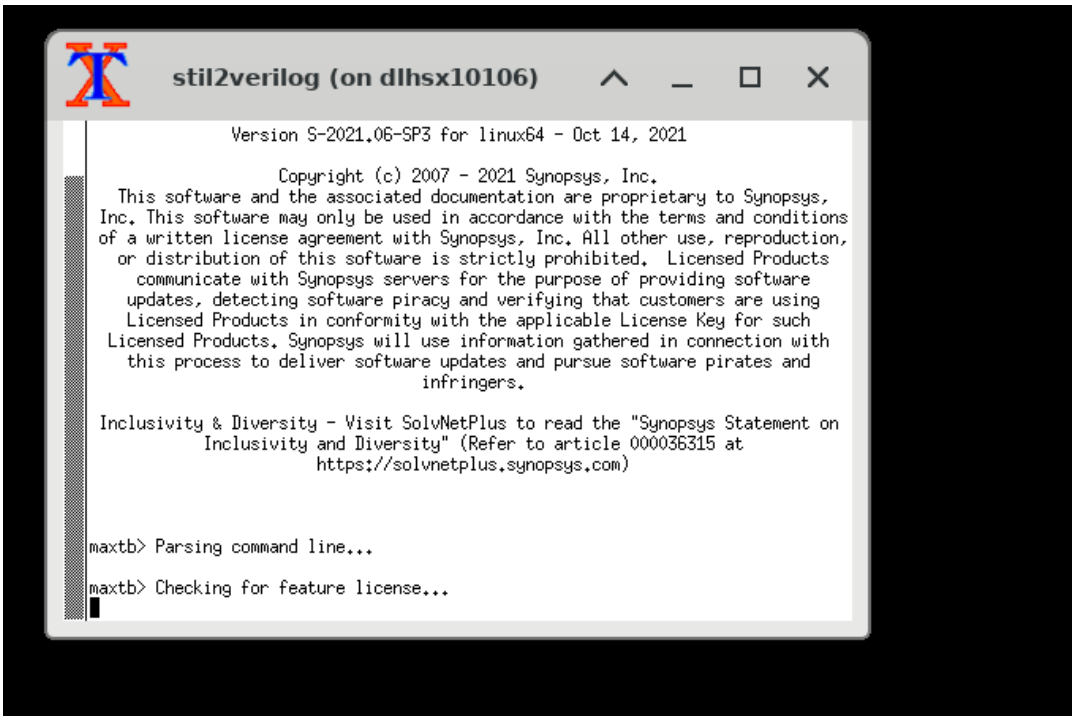


Figure 4.31 STIL to Verilog window

The .STIL file is then successfully converted to Testbench i.e. a .v file and .dat files, these are collectively known as patterns.

We will now simulate the generated patterns with the library file and the netlist on the shell as shown below in fig.4.32 and fig.4.33 respectively.

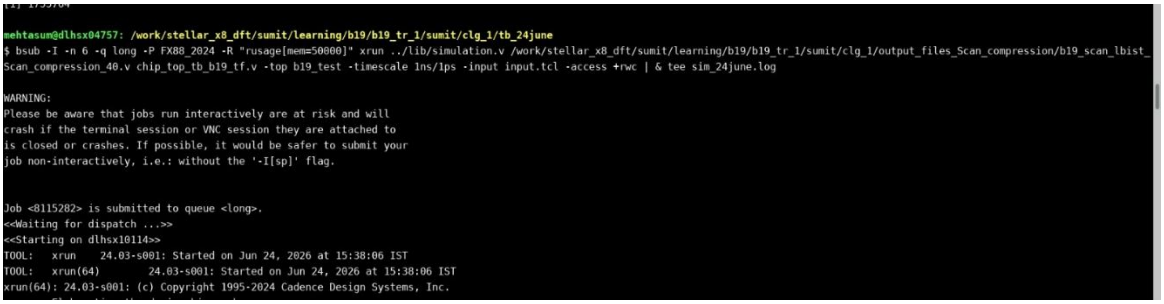


Figure 4.32 Pattern simulation window

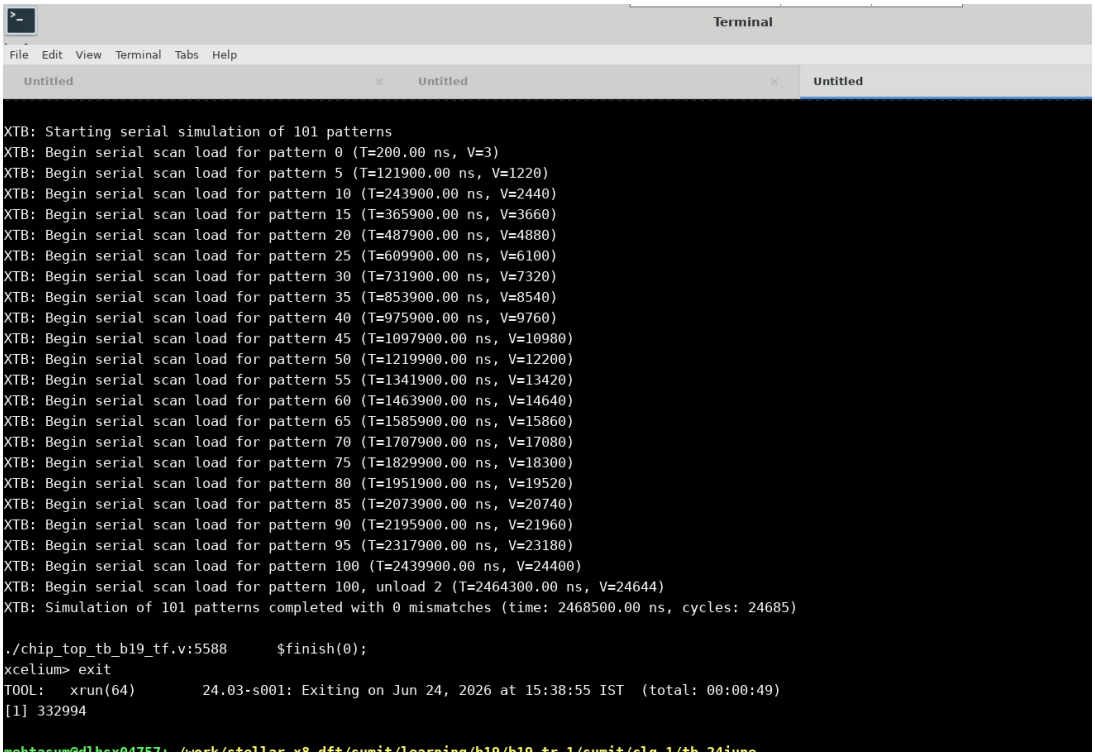


Figure 4.33 Serial simulation of patterns

Here in our design all the generated patterns are hereby successfully simulated and are passing on the intended design as shown in figure 4.25. Hence the DFT work on this design is done and this design can be referred to further stage of ASIC design and can move to Physical Design process.

We have also taken a dump/an output of the simulations and all the simulated signals as a .shm file. It stores all the signal transitions that occur during simulation so that you can debug the design later without rerunning the simulation.

We open the .shm file using Cadence Xcelium to visually see and verify all the simulated signals. We can verify the scan by looking at the signals like Scan_enable,

scan_clock, scan_in, scan_out, test_mode, etc. This is also shown in Fig.4.34.

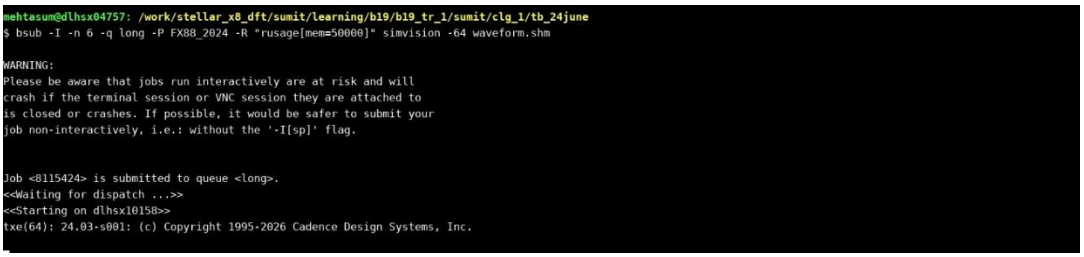


Figure 4.34 Opening the waveform in Cadence Xcelium

The fig.4.35 is the depiction of the simulation environment in the Cadence Xcelium:

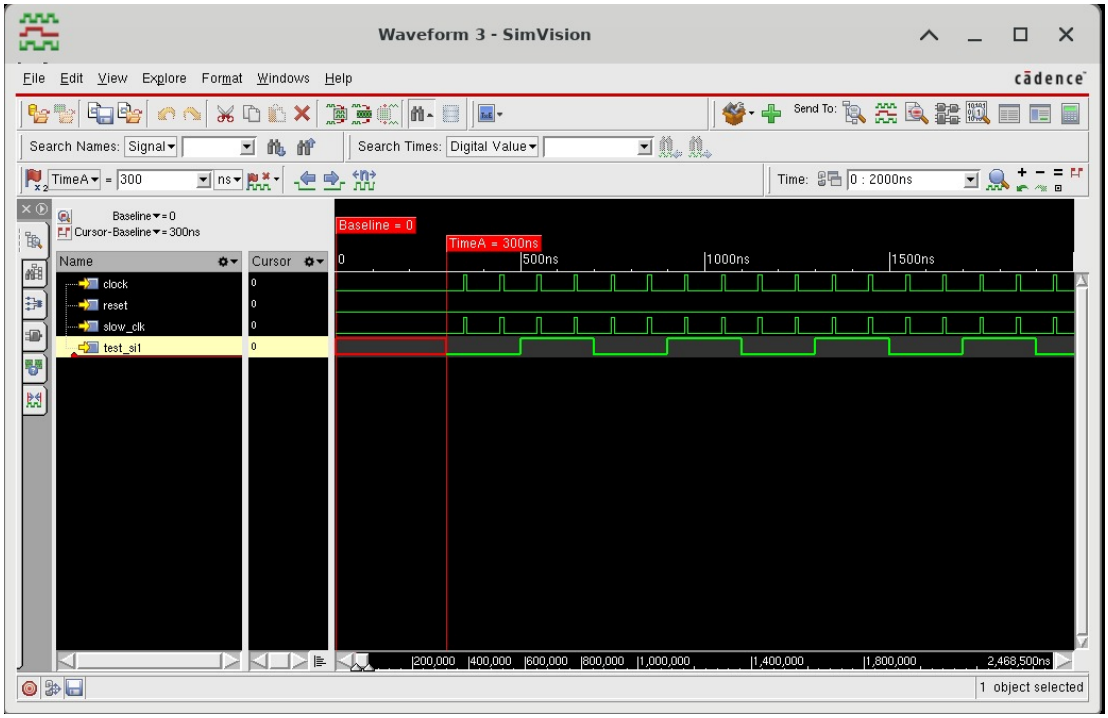


Figure 4.35 Cadence Xcelium window with all the important scan signals

Also the design can also be verified by looking and tracing the scan signals in the schematic in Cadence Xcelium as shown in Fig.4.36.

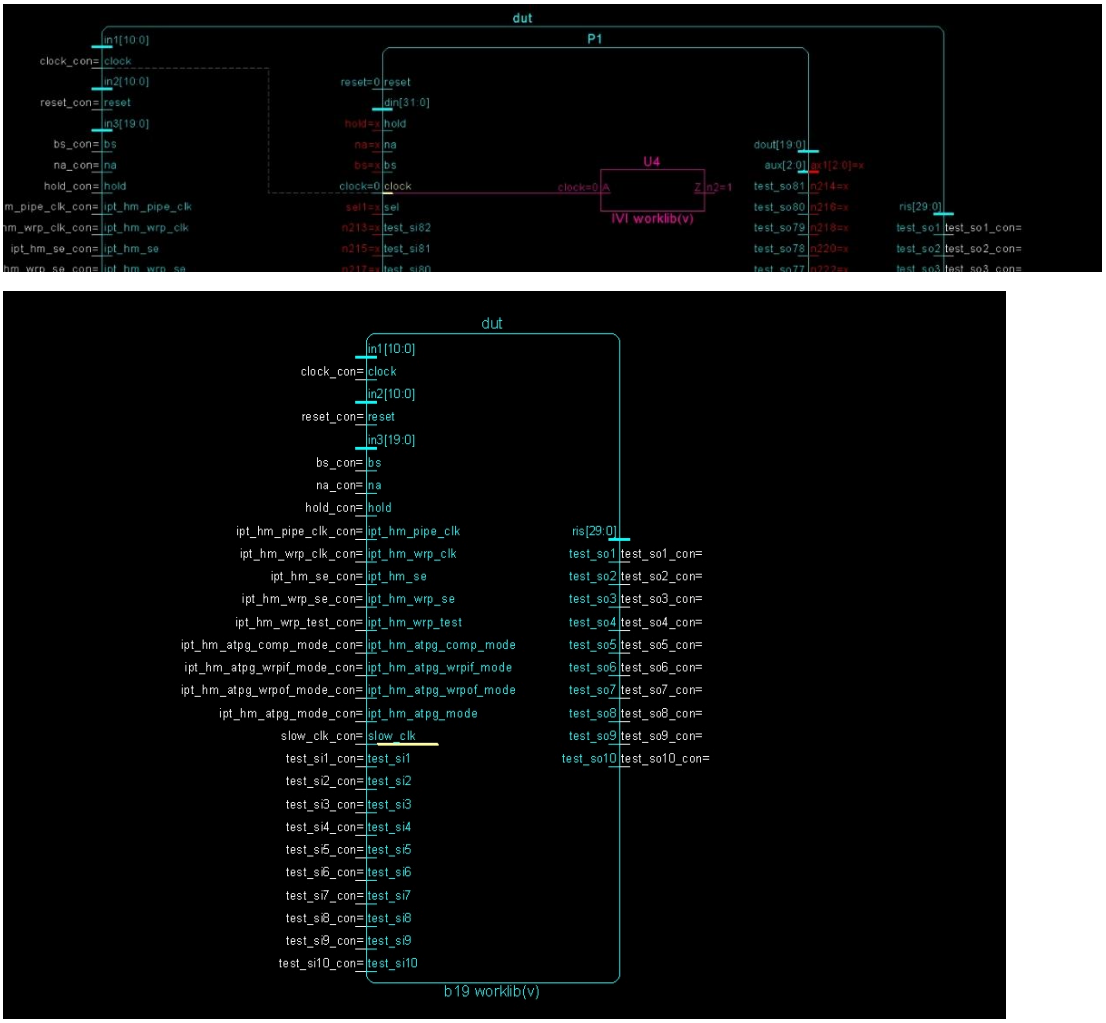


Figure 4.36 Schematic showing all the scan signals

CHAPTER 5

Conclusion

This thesis presents the work completed toward implementing a structured Design for Testability (DFT) flow for a digital design, with progress including scan insertion, pattern generation and fault coverage analysis and simulation. As modern VLSI systems grow increasingly complex, incorporating DFT methodologies early in the design process becomes essential for ensuring controllability, observability, and efficient testability. The work carried out in this thesis demonstrates the initial yet critical phases of this process, beginning from RTL design and functional verification and progressing through synthesis to scan insertion followed by pattern generation and fault coverage analysis and the finally the ATPG Pattern Simulations.

The B19 design was synthesized using Synopsys Design Compiler, generating a gate-level netlist and associated reports. Following synthesis, the scan insertion step was performed, wherein the sequential elements of the design were converted into scan cells and integrated into a structured scan chain. The comparison between pre-scan and post-scan schematics illustrates the expected architectural modifications introduced by scan insertion, including additional scan ports, scan enable signals, and chaining of flip-flops. Further the ATPG Pattern Generation has also been completed with an in depth analysis of the scan configurations with both stuck at and transition faults judging various parameters. Finally the ATPG Simulations have been completed with all the generated patterns successfully passing on the scan inserted design verifying the successful DFT insertion and functioning in the design.

These results validate that the scan architecture was correctly incorporated into the design, forming a foundational requirement for subsequent ATPG pattern generation and fault simulation. And further the ATPG patterns were generated successfully and correctly followed by the successful simulation of these patterns on the design validating the correctness of the inserted DFT.

In summary, the thesis successfully demonstrates the RTL-to-scan-insertion-to-ATPG-simulations flow and highlights the importance of structured DFT practices in enabling high-quality test generation.

References

[1] E Renold Sam Vethamuthu; S Sivanantham; R Sakthivel, "*Implementation of Hierarchical DFT Approach for Better Testability*," in International Conference on Emerging Trends and Innovations In Engineering And Technological Research (ICETIETR), 2018.

[2] Laung-Terng Wang, Cheng-Wen Wu, and Xiaoqing Wen., "VLSI Test Principles and Architectures: Design for Testability", Academic Press, 2006.

[3] Synopsys – TestMAX DFT User Guide, Version -P 2019.03-SP4, September 2019.

[4] Kun-Han Tsai, Yu Huang, Wu-Tung Cheng, Ting-Pu Tai, and Augusli Kifli, "*Test cycle power optimization for scan-based designs*," in 2010 IEEE International Test Conference, pages 1–10. IEEE, 2010.

[5] Bushnell and Agarwal, a book on "Essentials of Electronic Testing"

[6] Ivano Indino and Ciaran MacNamee, "*Dft: Scan testing issues and current research*," in Irish Signals & Systems Conference 2014 and 2014 China-Ireland International Conference on Information and Communications Technologies (ISSC 2014/CIICT 2014). 25th IET, pages 227–232. IET, 2013.

[7] https://www.design-reuse.com/articles/39857/atpg-challenges-at-lower-technology-nodes.html

[8] Kun-Han Tsai and Shuo Sheng, "*Design rule check on the clock gating logic for testability and beyond*," in 2013 IEEE International Test Conference (ITC), pages 1–8. IEEE, 2013.

[9] Synopsys – TestMAX ATPG and TestMAX Diagnosis User Guide, Version Q-2019.12, December 2019.

[10] Xiaoming Yu, M. Abramovici, "*Sequential circuit ATPG using combinational algorithms*," IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, volume: 24, Issue: 8, Aug. 2005.

[11] Miczo, "*The sequential ATPG: a theoretical limit*," in Proc. Int. Test Conf., Oct. 1983, pp. 143–147.

[12] N Burgess, RI Damper, SJ Shaw, and DRJ Wilkins, "*Faults and fault effects in nmos circuits-impact on design for testability*," in IEE Proceedings G- Electronic Circuits and Systems, volume 132, pages 82–89. IET, 1985.

18% Overall Similarity

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

Match Groups

114 Not Cited or Quoted 17%
Matches with neither in-text citation nor quotation marks

1 Missing Quotations 0%
Matches that are still very similar to source material

6 Missing Citation 1%
Matches that have quotation marks, but no in-text citation

0 Cited and Quoted 0%
Matches with in-text citation present, but no quotation marks

Top Sources

2% Internet sources
4% Publications
16% Submitted works (Student Papers)

Integrity Flags

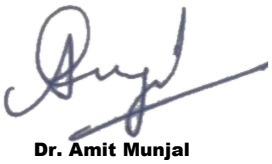
0 Integrity Flags for Review

Our system's algorithms look deeply at a document for any inconsistencies that would set it apart from a normal submission. If we notice something strange, we flag it for you to review.

A Flag is not necessarily an indicator of a problem. However, we'd recommend you focus your attention there for further review.

Sumit Mehta

Candidate signature



Dr. Amit Munjal

Supervisor signature

Match Groups

114 Not Cited or Quoted 17%
Matches with neither in-text citation nor quotation marks

1 Missing Quotations 0%
Matches that are still very similar to source material

6 Missing Citation 1%
Matches that have quotation marks, but no in-text citation

0 Cited and Quoted 0%
Matches with in-text citation present, but no quotation marks

Top Sources

2% Internet sources
4% Publications
16% Submitted works (Student Papers)

Top Sources

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

1	Student papers	
Maulana Azad National Institute of Technology Bhopal on 2022-04-27		13%
2	Publication	
"Power-Aware Testing and Test Strategies for Low Power Devices", Springer Scien...		<1%
3	Publication	
Zhang, Sheng. "Built-in self-test design optimization for scan-based circuits", Pro...		<1%
4	Publication	
Madhumithaa S. P. M, Aravind S, Harish S. P, Ramakrishna Prabhu Ch, Anita J. P. "...		<1%
5	Internet	
www.diva-portal.org		<1%
6	Student papers	
Visvesvaraya Technological University on 2014-11-22		<1%
7	Student papers	
Visvesvaraya Technological University on 2014-11-22		<1%
8	Student papers	
Visvesvaraya Technological University on 2014-11-24		<1%
9	Internet	
pdxscholar.library.pdx.edu		<1%
10	Publication	
"Design for AT-Speed Test, Diagnosis and Measurement", Springer Nature, 2002		<1%

11	Publication	"CTL for Test Information of Digital ICS", Springer Science and Business Media LL... <1%
12	Internet	eprints.soton.ac.uk <1%
13	Internet	download.bibis.ir <1%
14	Internet	www.coursehero.com <1%
15	Publication	Ashutosh Kumar Singh, Hari Mohan Gaur, Umesh Ghanekar. "Fault detection in ... <1%
16	Student papers	ITESM: Instituto Tecnologico y de Estudios Superiores de Monterrey on 2026-03-19 <1%
17	Publication	H. Fujiwara. "A new class of sequential circuits with combinational test generatio... <1%
18	Internet	gsajournals.org <1%
19	Student papers	Ho Chi Minh University of Technology and Education on 2025-06-22 <1%
20	Student papers	Infile on 2021-06-02 <1%
21	Student papers	Institute of Technology, Nirma University on 2017-05-31 <1%
22	Student papers	Maulana Azad National Institute of Technology Bhopal on 2022-05-04 <1%
23	Student papers	National Institute of Technology, Kurukshetra on 2026-06-22 <1%
24	Student papers	Universiti Teknologi MARA on 2019-07-08 <1%

25 Student papers
Universiti Teknologi Malaysia on 2021-02-12 <1%

26 Student papers
Universiti Teknologi Petronas on 2011-03-18 <1%

27 Internet
etheses.bham.ac.uk <1%

28 Publication
Joe, Jerin. "Incremental Test Pattern Generation for Structurally Similar Circuits", ... <1%

29 Student papers
Visvesvaraya Technological University on 2014-11-24 <1%

30 Student papers
Department of Industrial Management on 2026-05-25 <1%

31 Student papers
Manipal University on 2019-07-19 <1%

32 Student papers
Visvesvaraya Technological University on 2014-11-22 <1%