"Pre-Silicon Scan Validation and Pattern Transformation"

Project Report

Submitted in Partial Fulfillment of the
Requirements for the Degree of

MASTER OF TECHNOLOGY

IN

VLSI DESIGN

By

Anju M Katarmal
(19MECV08)

Department of Electronics and Communication Engineering,
Institute of Technology,
Nirma University,
Ahmedabad 382 481

May 2021

Declaration

This is to certify that

1. The thesis comprises my original work towards the degree of Master of Technology in VLSI Design at Nirma University and has not been submitted elsewhere for a degree.
2. Due acknowledgment has been made in the text to all other material used.

Anju Katarmal

19MECV08

Certificate

This is to certify that the project entitled "**Pre-Silicon Scan Validation and Pattern Transformation**" submitted by **Anju Katarmal (19MECV08)**, towards the partial fulfillment of the requirements for the degree of Master of Technology in VLSI Design, Nirma University, Ahmedabad. The record of work carried out by her under our supervision and guidance. In our opinion, the submitted work has reached a level required for being accepted for examination. The results embodied in this major project, to the best of my knowledge, haven't been submitted to any other university or institution for award of any degree or diploma.

Dr. Manish Patel
Internal Guide

Dr. Usha Mehta
PG Coordinator – VLSI Design

Dr. Dhaval Pujara
Head, EC Dept. Institute of Technology

Date:
Place: Ahmedabad

Certificate

This is to certify that the project entitled "**Pre-Silicon Scan Validation and Pattern Transformation**" submitted by **Anju Katarmal (19MECV08)**, towards the partial fulfillment of the requirements for the degree of Master of Technology in VLSI Design, Nirma University, Ahmedabad. The record of work carried out by her under our supervision and guidance at **Intel Technology India Pvt. Ltd.** In my opinion, the submitted work has reached a level required for being accepted for examination.

External Guide:
Ramesh Thota
Pre-Si Valid/Verif Engineer
Intel Technology India
Bangalore
Date:

Acknowledgement

Developing and evolving this project has been very critical and joyful throughout the year. The inner gratification for the accomplishment of the project successfully would not be complete without mentioning organization and mentors who helped me in completing this project work on time. Their support and motivation made my efforts fruitful.

Foremost, I extend my deepest gratitude to “Sc Parthasarathy”, “Ramayanam Balaji” and “Ramesh Thota” for giving me the opportunity to work with their group and guiding me in this challenging project.

I do sincerely appreciate my teammates “N Madhan”, “Ravi Kumar, G” and “B, Arun” for their constant assistance, support and constructive suggestions in the better-ment of this project, without which this would not have been possible.

I cordially extend my profound gratitude to: “Dr. Manish Patel” and “Dr. Usha Mehta”, for their motivation and valuable advice during the project work.

Last but not the least, I would like to take this opportunity to thank my family, friends and colleagues for their help and valuable suggestions from time to time.

Anju M Katarmal

(19MECV08)

Abstract

As the transistor size reduces, we can accommodate more complex logic in a very small chip. Testing of such a small chip is a challenging task. Today testing of the chip takes around 30% to 50% of the total product cost. Test techniques have become crucial part in success of the product in marketplace. Design for testability is a technique used to detect the manufacturing defects using test generation and test application. Test generation is accomplished by Automatic test pattern generation using fault models and algorithms. Test generation for sequential circuits is difficult due to poor initializibilty, controllability and observability of the circuit. Scan design is one of the most widely used approach for improving the intializibility, controllability and observability of the circuit by converting the storage element in the circuit into combination of the mux and flipflop. This way sequential circuit becomes virtual combinational circuit, for which combinational ATPG can be used for test pattern generation. In this work, scan validation architecture has been explored. RTL based scan validation strategy is useful in detecting faults in the early stages of the cycle. Pattern transformation for converting the block level test pattern into full chip level pattern has been discussed in the later part. After that, as part of scan validation, evil validation is introduced, where the effect of corrupted scan cell is analyzed over the critical signals of the systems like powergoods, clock signals etc. As a part of project, automation was implemented to make the scan validation more efficient by reducing manual efforts.

Table of Contents

Declaration ..... **i**

University Certificate ..... **ii**

Company Certificate ..... **iii**

Acknowledgement ..... **iv**

Abstract ..... **v**

Table of Contents ..... **vi**

List of Figures ..... **viii**

Nomenclature ..... **ix**

1. Introduction **1**

1.1 Company Profile 1
1.2 Introduction 1
1.3 Scope of Training/Project 2
1.4 Orgainization of the Rest of the Project 2

2. Literature Review **3**

2.1 Scan Design 3
2.1.1 Mux-D Scan cell 3
2.1.2 Scan Operation 4
2.1.3 Scan Compression 5
2.2 Test Access Port 6
2.2.1 TAP Controller 8

3. Scan Validation **11**

3.1 Scan Architecture 11
3.2 Pattern Transformation 12
3.3 Evil Validation 13
3.4 Automtion in Scan Validation 15

4. Simulation and Result Discussion **19**

4.1 Pattern Transformation: Single Shift Cycle Issue 19

4.2 Evil Validation 20
Conclusion 22
References 23

List of Figures

Figure 2.1Mux-D scan cell and scan chain with mux-D cells 3
Figure 2.2 Scan Operation with shift and capture mode 4
Figure 2.3 Mentor's Embedded Deterministic Test compression system 5
Figure 2.4 IEEE 1149.1 Test Logic[2] 7
Figure 2.5 TAP controller Finite State Machine[2] 8
Figure 3.1 Scan Architecture with scan controllers,TAPs and test fabric 11
Figure 3.2 Evil validation test flow 14
Figure 3.3 Prepared excel sheet after automation with Regions and tests 17
Figure 3.4 Flow chart of the working of the automation script for comparing two init sequences 18
Figure 4.1 Single Shift Cycle pattern issue 19
Figure 4.2 Summary of the output of pattern test 20
Figure 4.3 Evil Validation corrupted clk signal and driver signals 20
Figure 4.4 Evil Validation corrupted clkout signal with traced signals zoomed in view 21

Nomenclature

Subscripts

mux	Multiplexer

Abbreviations

ATPG	Automatic Test Pattern Generator
DFT	Design For testability
EDT	Embedded Deterministic test
TAP	Test Access Port

1. Introduction

1.1 Company Profile

Intel Corporation is an American multinational corporation and technology company headquartered in Santa Clara in the city of California, at the heart of Silicon Valley. Based on revenue it is the world's second largest semiconductor chip manufacturer. It is also the inventor of the x86 series of microprocessors, the processors found in most personal computers (PCs). Intel ranked No. 46 in the 2018 Fortune 500 list of the largest United States corporations by total revenue. Incorporated in Delaware, Intel supplies processors for computer system manufacturers such as Apple, Lenovo, HP, and Dell. Motherboard chipsets, flash memory, graphics chips, embedded processors, network-interface controllers, and integrated circuits are also a part of the diverse portfolio of hardware manufacturing at Intel. It also produces many computation and communication ancillaries which have become an integral part of our day to day lives.

1.2 Introduction

As the transistor size is reducing millions of transistors are accommodated on a small chip to achieve the complex functionality within smaller area. As the complexity of the system increases, it becomes difficult to test the chip. As the test costs takes almost 40% of the total costs, test technologies have become crucial in market success of the product.

When the transistors are fabricated in the foundaries, many reasons like coefficient mismatch, mask misalignment can introduce manufacturing defects. Design for testing (DFT) is the technique which model the manufacturing defects in fault models and provides different approaches to test the defects.

Most common fault models are stuck-at and at-speed. Stuck-at model implies the defects which occurs due to any input or output is permanently connected to ground or the supply voltage. At-speed model implies to the transition faults (slow to rise or slow to fall). Test generation and test application are the two main process of VLSI testing. Automatic test pattern generator (ATPG) is a tool that generates the test patterns for the given fault model.

Testing of the sequential circuits is difficult due to poor initializibility, poor controllability and observability. No. of test cycles required to test such circuits are large. To mitigate this problem, flip flops in the system are converted into combination of flip flop and mux and later they are connected

into serial manner like shift registers to make scan chains. Through select input of the mux, this scan chain can be initialized. Thus, scan system makes the sequential circuit into virtual combinational circuit. Combinational ATPG can be used to generate patterns for such circuits.

Scan validation in RTL facilitates the detection of the issues in early stage of the product cycle, which can save test time and test efforts. Also, validation at the gate level netlist is a complex task. So, scan-based RTL validation is performed to catch the issues in the early design cycle.

1.3 Scope of Training/project

Scope of the training includes:

- Understanding scan concepts, scan Architecture, scan validation environment
- Performing a pre-silicon scan validation to ensure the post silicon testing capability.
- Performing pattern transformation in order to prepare scan content to be ready for chip level testing

1.4 Organization of the rest of the project

Rest of the report is organized as follows: First concept of scan is introduced, and scan operation is explained. After that scan compression using mentor's Embedded deterministic test compression tool is explained. Test Access port (JTAG 1149 standard) is described later in the chapter 2. Scan architecture is introduced in chapter 3. Pattern transformation flow responsible for converting the block level netlist patterns into chip level netlists, is shown later in chapter 3. As part of scan validation, introduction to evil validation is given. An automation in the scan validation flow was explained in the later part of the chapter. Debug issues faced during pattern transformation and evil validation are discussed in chapter 4. At last, the report is concluded with references.

2. Literature Review

2.1 Scan Design

As the transistor size reduces, whole system designs become more complex. Observability and controllability of the circuit reduces as the complexity of combinational circuit increases. Observability and controllability have become a serious issue for testing sequential circuits. More no. of test patterns are required to put the sequential elements into known state for fault simulation, which increases the test time by a large amount. Also, more powerful ATPG tool is required to generate the patterns for sequential circuits. Scan is one of the most used structural approaches to improve the testability of the sequential circuits.

Scan design improves the testability of the circuit by increasing the observability and controllability of storage elements. Here storage elements such as flip flops are converted into scan flops and are stitched together in serial manner resulting in chain. As a result, sequential circuit will be converted into virtual combinational circuit. Combinational ATPG can be used to generate patterns for such circuits. This chain can be loaded serially from the tester, allowing each sequential element to be controlled.

Scan design has three mode of operations: normal(functional) mode, shift mode and capture mode. In normal mode, all test signals are deasserted and system works in functional configuration. In shift mode and capture mode, data can be serially loaded and unloaded into scan chain providing the better observability and controllability. Capture mode will allow the output of the combinational circuit to latch into the nearby connected flip flop [1].

2.1.1 Mux-D Scan cell

Here flip flop in the sequential circuit is replaced by the combination of the 2X1 mux and D flip flop and are connected serially as shown in figure 2.1

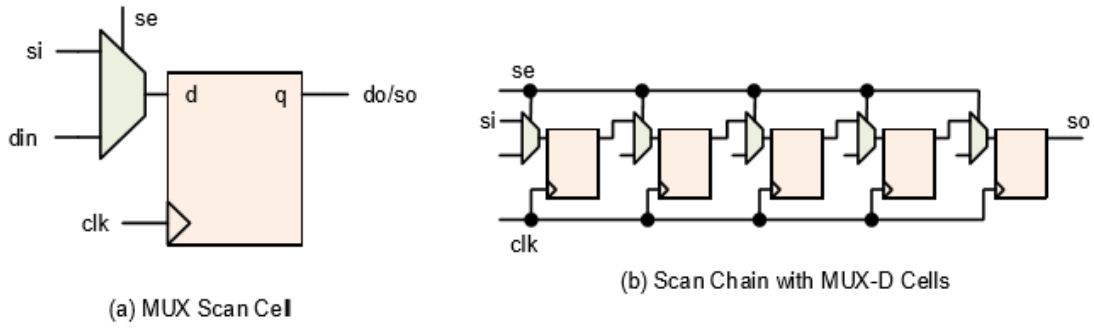

Figure 2.1Mux-D scan cell and scan chain with mux-D cells

To select between data input(din) and scan input(si), Multiplexer has scan enable(se) signal. In normal/capture mode se is set to zero. The value present at the din will be loaded into the D flip flop. During scan mode, se is set to one, allowing value present at the si input to be shifted into scan chain while data is shifted out from the so at the other hand.

Mux-D scan design is straight forward concept and used widely, but it has a disadvantage of added delay and area of mux.

2.1.2 Scan Operation

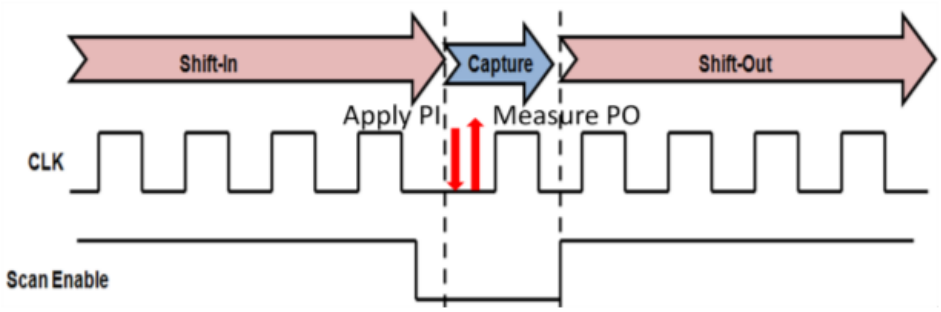

Figure 2.2 Scan Operation with shift and capture mode

Scan operation can be defined as follows:

First the scan enable signal is set to have the system in shift mode. Scan data bits will be shifted in through scan-in inputs in scan chains.

Then scan enable signals turns low, which will make the system to work in functional mode. Here primary inputs are applied. After that primary output is measured at the output of the combinational circuit.

Now in the capture mode, scan enable will be zero. Output of the combination circuit is captured by the nearby flip flop at the negative edge of the clock.

Again, the scan enable is set to make the system in shift mode, shifting out the data through scan out. Here next test pattern is serially shifted in through SI input [1].

2.1.3 Scan Compression

One of the big challenges in the scan ATPG patterns are large data volume. There could be thousands of flops in each partition. Resulting large patterns would increase the test time, or we will have to drive patterns at high frequencies. That will result in excessive power consumption. To reduce data volume, scan compression is used. Below figure 2.3 is the block diagram of Mentor's Embedded Deterministic Test compression system used in scan system.

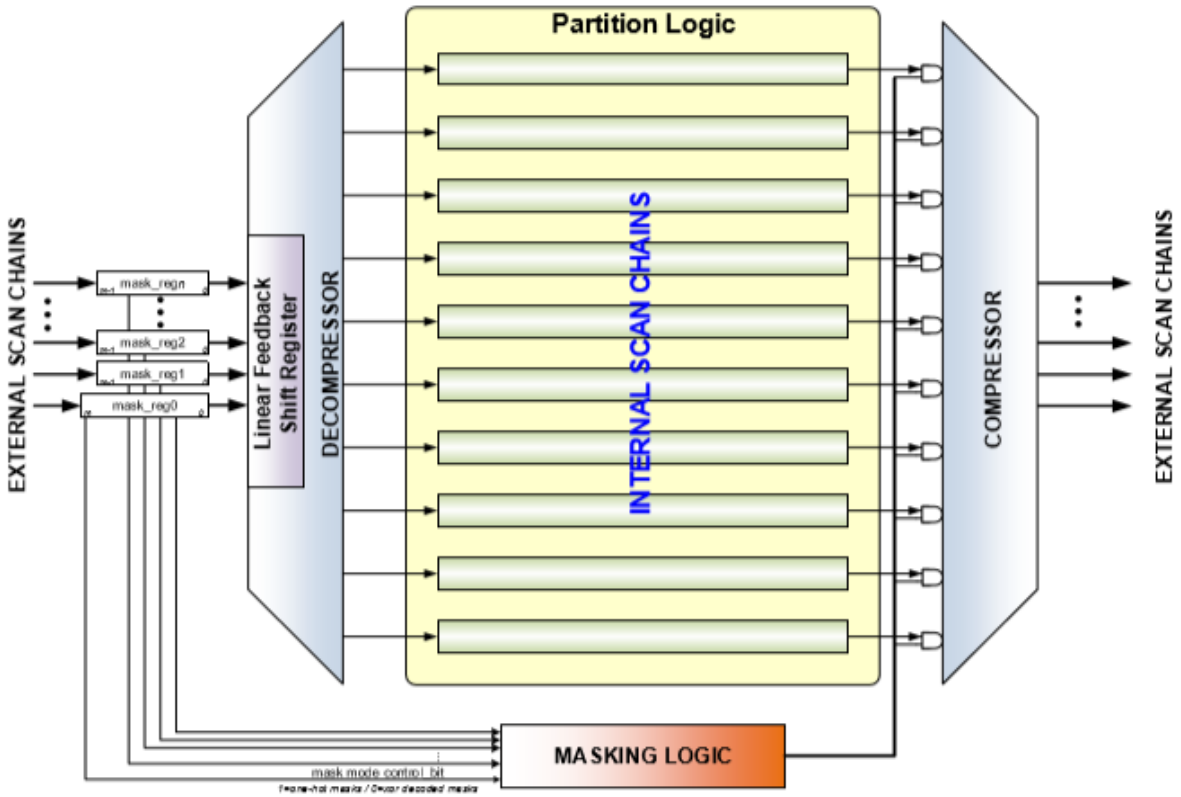

Figure 2.3 Mentor's Embedded Deterministic Test compression system

ATPG tools generally target on the small set of faults per run and generate patterns for those faults. Untargeted bits are filled with random data and simulated to get the coverage for random fill. As the large amount of data in ATPG patterns is random, If the random data generator is implemented in hardware, only small amount of data needs to be specified to the tester. This is the main concept behind

scan compression. If the same random generator used in ATPG tool can be implemented in the hardware, great reduction can be achieved.

Random sequence generation is achieved using linear frequency shift register (LFSR), which is simple shift register with XOR taps in the feedback to create maximum length sequence. But the traditional LFSR has high fanout, leading to timing challenges. To resolve this, mentor's EDT tool uses ring LFSR structure, where the LFSR is folded back on itself to form the ring. Since state of the LFSR must match exactly with ATPG simulation, LFSR is reset at every load/unload vector. This is accomplished by setting the EDT_UPDATE high in mentor's EDT tool.

Input to the EDT is referred to as external chains, which are fed to LFSR to steer the LFSR. Output of LFSR drives the decompressor logic called phase shifter. Phase shifter is constructed using XOR gates and is responsible for generating the controlled random values into each of the partition chains referred to as internal chains.In conclusion, small no. of input control bits can control a very large number of scan chains.

On the output side, scan data from the internal chains are compressed back to small number of external chains using parity style XOR tree. But X created in the logic by some sources like uninitialized array, uncontrolled primary inputs may impact the entire compressor. To mitigate this problem, mentor's EDT uses maskable compressor XOR logic tree.

Mask information is the last bit in external chain in the mask register when the EDT_UPDATE is true. Mask registers are in series with shift registers. Mask information is loaded from the external chains to mask register via shift register.

2.2 Test Access Port

JTAG stands Joint Test Access Group also known as IEEE 1149.1 is a standard used to overcome the lack of standardization in testing individual chips at the board level. Test access port consists of three input pins TDI (test data input) and TCK (test clock) and TMS (test mode select). There is optional third pin TRST (test reset) to provide asynchronous initialization to test logic. It has an output pin called TDO (test data output).

Test clock is included to make sure that the test data path can be used independently between devices having different frequencies and to accommodate the concurrent shifting of the test data with normal operation of the device[2].

TMS (test mode select) at the edge of the test clock will decide the state of the TAP state machine. TDI is the serial input for test instruction and data to test logic. TDO is the serial output for test data and instruction.

Block diagram of the test logic architecture is shown below in Figure 2.4. It consists of TAP controller, instruction register and set of data registers. Instruction register and data registers are shift registers connected in parallel. They are connected to TDI and TDO via common serial path. Selection between data register and instruction register is made by state of TAP controller[3,4].

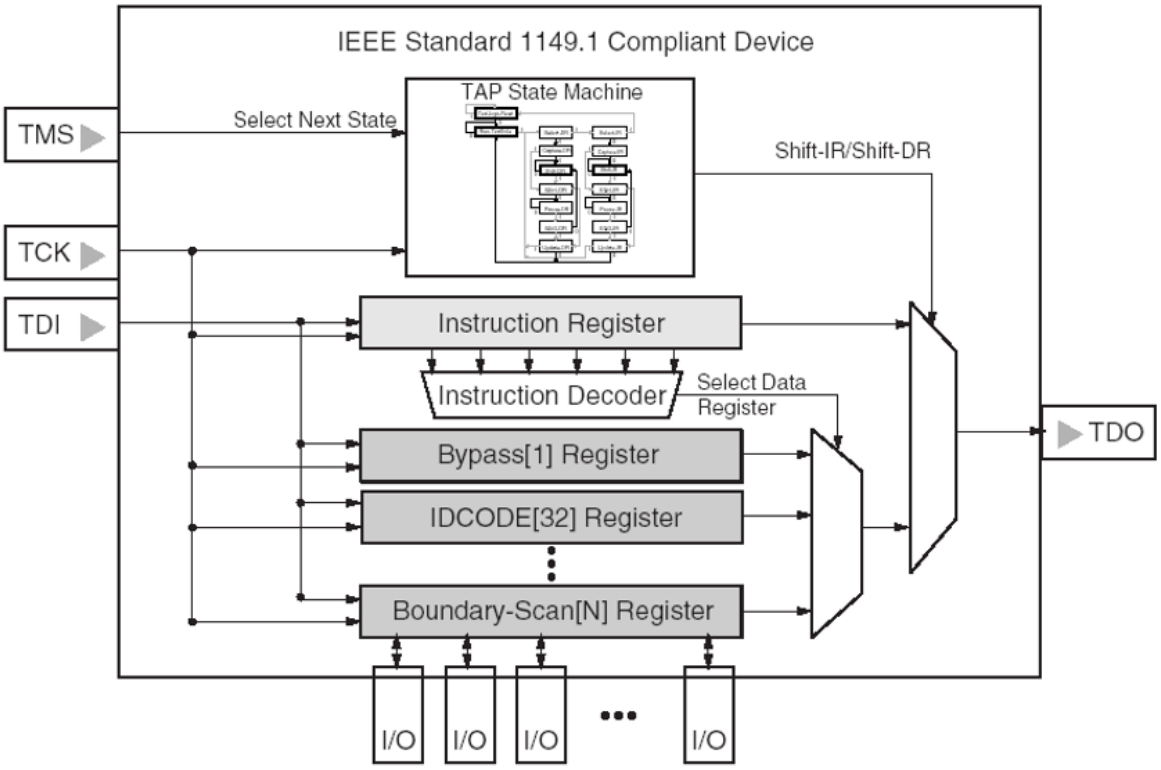

Figure 2.4 IEEE 1149.1 Test Logic[2]

TAP controller generates clock and control signals for instruction and data registers. Instructions are shifted into design through instruction register, providing the information about what test to be performed. Test data registers include bypass register, boundary scan register and sometimes may include device identification register etc.

2.2.1 TAP Controller

TAP controller is a 16-state finite state machine that changes the state on the positive edge of the TCK based on the value of TMS to control the behavior of the test logic. TAP controller FSM is shown in the figure 2.5 below[3].

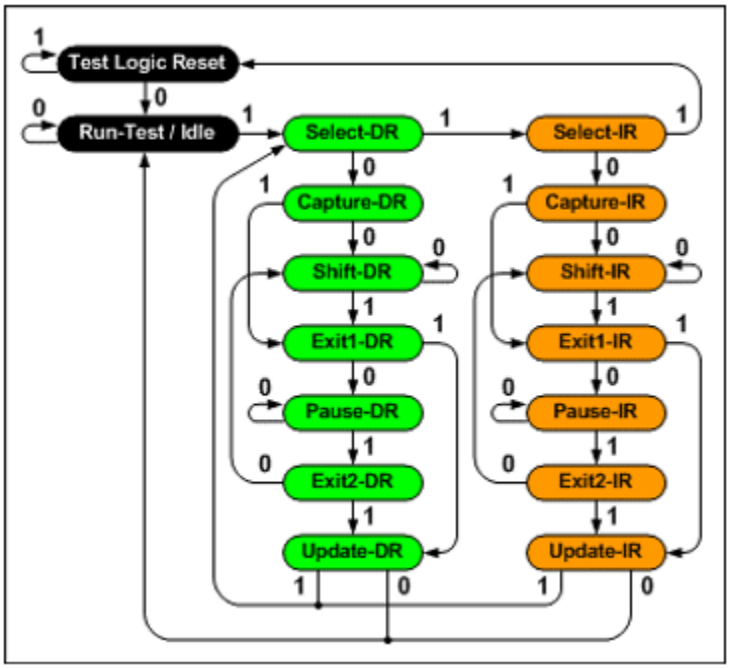

Figure 2.5 TAP controller Finite State Machine[2]

On reset, TAP controller goes into the test logic reset. Below all states have been describes in brief.

Test Logic Reset:

This is a reset state. Here the test logic is disabled to let the normal operation of the circuit to proceed. When TMS is high for consecutive five clock pulses, controller will enter this state regardless of the current state. It will remain in this state till TMS is asserted. When the current state of test logic is low at the positive edge of the TCK, controller goes into Run-Test/idle state[2].

Run-Test/Idle:

It's the state between scan operation. Selected activity in the test logic will only occur in this state if there is instruction present. When the controller is in run-test/idle state, and TMS is asserted on the rising edge of the TCK, TAP controller will move to select-DR state.

Select-DR:

It is a temporary state. When select-DR is current state and TMS is asserted at positive edge of the TCK, controller will move to the select-IR state. If TMS is low at the rising edge of TCK, TAP controller will move to capture-DR state, where the scan sequence for data register is initiated. TMS value during the select-DR state will decide whether the TAP controller will perform the actions for data register or for instruction register.

Select-IR:

Select-IR state is again a temporary state. When the current state of the controller is select-IR, value of the TMS at the positive edge of the TCK will decide whether tap controller will continue to go further for instruction or will move to Test logic reset state as shown in state diagram.

Capture-DR:

During the capture-DR state, test data is loaded parallelly into shift-capture path. This is temporary state. If the TMS is asserted at the positive edge of the TCK, controller moves to exit1-DR state, otherwise, it moves to shift-DR state.

Shift-DR:

Here data is serially shifted from the TDI towards TDO through data registers at every positive edge of the TCK.Shifting operation will continue till the TMS is held low. When the TMS is high at the positive edge of TCK, controller will move to exit1-DR state.

Exit1-DR:

When controller is in exit1-DR state and TMS is asserted, controller will enter update-DR state, otherwise if TMS is low then controller will enter pause-DR state.

Pause-DR:

In this state shifting of the data is temporarily put on hold. This is needed when ATPG tool may need to load long sequences from the memory which requires time. TAP controller will remain in this state till TMS is held low.

Exit2-DR:

When TAP controller is in pause-DR state and TMS is high at the positive edge of TCK, controller will enter the temporary state exit2-DR. Here if the TMS becomes low, controller will enter shift-DR state and shifting of data will be resumed. If TMS turns high, controller will move to update-DR state.

Update-DR:

In this state, data will be parallelly latched-out from the data register outputs on the negative edge of the TCK. When controller is in this state and if the TMS is held high at the rising edge of the TCK, controller will move to select-DR state. Otherwise if the TMS is held low, it will move to run test/idle state.

Capture-IR:

Here patterns of the fixed logic are parallelly loaded into the specific bits of the instruction register shift-capture path. This is a temporary state, and controller will move to shift-IR state as TMS will be held low during the positive edge of the TCK. At the positive edge of the TCK, if TMS is asserted, controller will enter exit1-IR state.

Exit1-IR:

This is again a temporary state, from where controller will move to pause-IR mode or update-IR mode depending on the value of test mode select at the positive edge of the TCK as shown in figure.

Shift-IR:

Here data will be shifted through instruction register connected between TDI and TDO at each rising edge of the TCK.When controller is in this state, if TMS is held high at positive edge of the TCK, controller will move to exit1-IR state. Controller will stay in this state till TMS is held low.

Pause-IR:

In this state, shifting of the data will be temporarily halted. This is required to accommodate the time in case ATE needs to load data from memory.

Exit2-IR:

This is again a temporary state which controller enters in when controller is in pause-IR state, and TMS is asserted during the positive edge of the TCK. When the controller is in exit2-IR state and TMS is low, controller will move to shift-IR state and resume shifting. But if TMS is kept high, controller will enter update-IR state.

3. Scan Validation

3.1 Scan Architecture

Scan system divides the whole SOC into scan regions and has a JTAG+IJTAG network. This scan regions are further divided into partitions. Each partition has its own scan controller responsible for generating control signals including scan control signals, clock signals, synchronous and asynchronous reset signals locally according to partition's specifications. Scan controller has EDT included into it to provide scan compression locally to each partition. Test fabric is used to accommodate the parallel transmission among scan regions. There is a bridge between the all partition of the scan region and test fabric, which will be responsible for transferring the data between partition scan controller and fabric and generating global control signals as shown in figure 3.1.

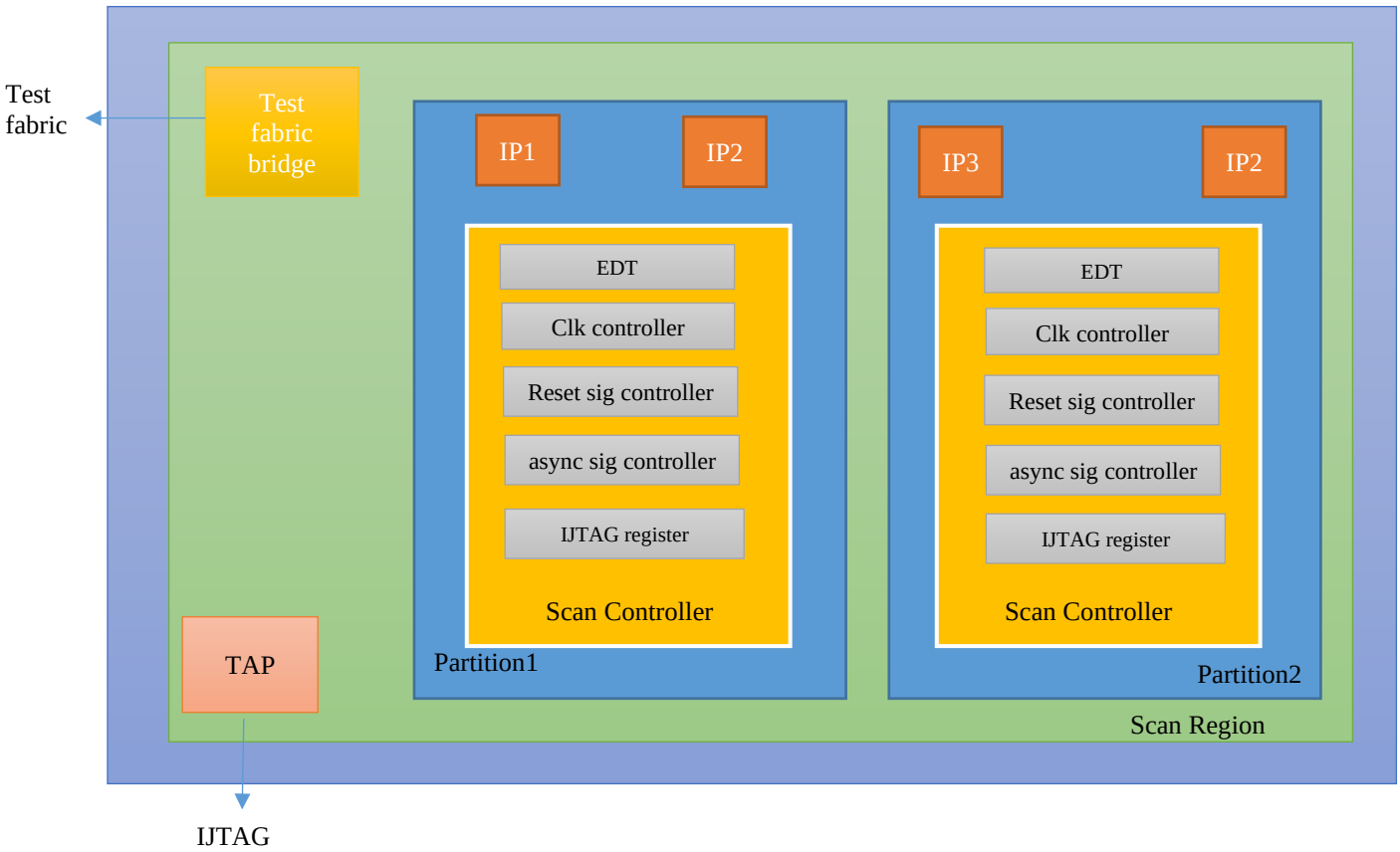

Figure 3.1 Scan Architecture with scan controllers,TAPs and test fabric

3.2 Pattern Transformation

This topic contains the flow of pattern transformation which is used to convert the block level ATPG patterns generated by mentor tools into full chip level patterns.

ASCII patterns generated by ATPG flows are either at the block netlist level, IP level or at the partition levels. Reason behind generating the patterns at the such level rather than at full chip level is the requirement of large level of resources and the test time. Also, full chip level netlists are usually available at the later stages. In order to apply these patterns at the full chip level we need to have transformation process to have a pattern ready for the full chip application.

Patterns applied at the full chip level have mainly three distinguishable section: preamble, midamble and main test. The sequence of the preamble + midamble, is the test Reset. Preamble is responsible for generating the appropriate reset sequences. The midamble is responsible for some overhead needed to set up a new main test. Reset sequences are the first step in the test patterns. Reset sequences can be termed as the set of events that happens before the any main test application. Main goal of reset is to make sure:

- All the IPs are rebooted and have come out of reset state
- All the clocks are toggling at the expected frequencies
- Have necessary DFX features setup and ready to be utilize

Reset sequences are generated by recording the activity on the DUT stimulus, which is later converted to the tester pattern format by pattern transformation tool. The tester can drive the Silicon Pins and mimic the test that was running in the simulation. RESET will be developed as a test which will be simulated in design. In a later phase all the tested contents will use this flow as a common part to initiate the SOC Platform (Power On & Reset).

After the reset, there will be sequences having all scan related programming which is common across all tests such as most of the TAP configurations, initialization of scoreboard and scratchpad etc. These sequences will be run once after reset but before main test. Main test contains the init sequences and main test content, which is the scope of this document.

Block/partition level test pattern file generation of the scan test content is owned by ATPG team executed using mentor tessent tools. It basically marks the starting-point for the transformation, for all scan content types (such as stuck-at, at-speed) and scan system configuration (EDT on or bypassed etc.)

There are additional sequences that contain final configuration of scan system. ATPG tool generates the patterns with set of constraints referred to as ATPG assumptions. It is crucial that on Silicon, these conditions are satisfied.

Creation of these sequences and main test transformation are accomplished by the supporting perl scripts. There are three perl scripts, which are responsible for:

- Parsing the data for all scan regions into single hash and creating the configuration template for all the scan regions.
- Collecting the pre-determined force statements from block level test patterns files and generating the sequence file according to it.
- Creating the scan patterns for given scan region and generating chip level test pattern files for block level test pattern files given in the directory.

3.3 Evil validation

Scan cells are stitched during synthesis to improve the controllability and observability of the system. Scan validation is performed to ensure the proper functionality of the system during scan operation such as to check clocks are properly enabled, to check power related issues etc. Introduction of the scan should not change the original functionality.

Scan cells are stitched in a serial manner like shift registers to make the scan chains. If any scan cell is corrupted, we lose the entire coverage of the chain, we can't test it. Scan cell can get corrupted due to tie up or improper stitching. In order to make sure that even if one scan cell is corrupted, entire scan functionality does not get affected, evil validation is performed.

Flow chart of evil validation is shown below in the figure 3.2:

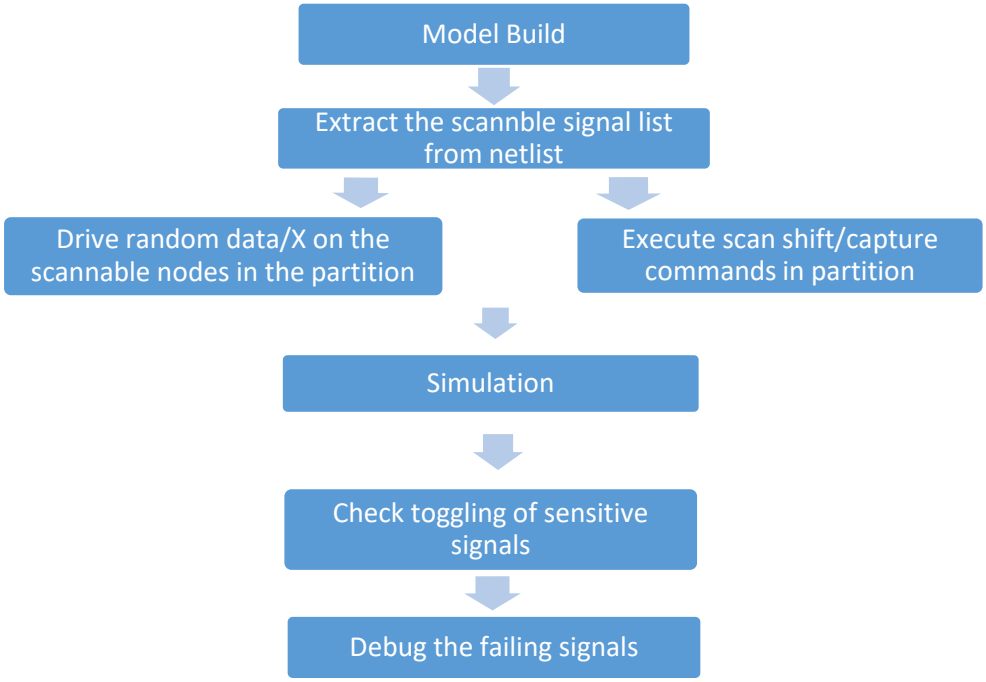

Figure 3.2 Evil validation test flow

First the model is built up. As we do the scan validation at the RTL level and scan cells are stitched on gate level netlist, we don't know where the scan cells are present. So, we take the scan cell report given by the ATPG team and map it with the RTL hierarchy to get the scannable signals. RTL hierarchy generated from regression run, contains all design signals with their hierarchy information and information regarding how signal is toggling.

In order to validate the scan functionality when the scan cell is corrupted, these scannable signals are deliberately driven by X or random data and their effect on the critical signals are analyzed. A tool will replicate the signal behavior from normal regression till the specified time. After the scan mode is asserted, tool will force the scannable signals given in the input to X/random data as specified by the user. After the regression is run, sensitive signals are extracted from RTL hierarchy and analyzed.

These sensitive signals mainly include global clock signals, clock enable signals and powergood signals etc. If any of such sensitive signals are found to be X or random toggling, then debug is carried out for those signals.

3.4 Automation in Scan Validation

The scan test plan includes several pre-written tests and a library of reusable sequences. Users can use the pre-written tests to validate a large majority of their design.

The test content covers mainly the following aspects of design:

- Network DOA tests
- Scan proxy tests
- Scandump tests

- **Network DOA tests**

  Network Dead or Alive (DOA) tests are a set of basic IJTAG network and test fabric network register read/write tests and reset value check tests.

- **Proxy-based scan tests**

  Proxy-based scan tests are a set of tests that focus on actual scan functionality such as scan continuity and scan capture. These tests use the proxy circuit to mimic functional logic, providing a mechanism to perform RTL-based scan validation.

  This set of tests comprises various scan usage models:

  - Scan chain continuity
  - Stuck-at capture
  - At-speed capture
  - EDT modes: Intest (default) and Extest

  The general structure for proxy-based scan tests is as follows:

  - ❖ The test runs initial design reset sequence to ensure all clocks are running before entering test mode
  - ❖ Programs IJTAG or TAP network for test setup
  - ❖ Loads partition wise scan chains to program an initial value for proxy circuit and control registers
  - ❖ Scans capture sequence to start the capture clocks
  - ❖ Unloads scan chains for proxy circuit and compares the post-capture values

- **Scan Dump:**

  All scan chains are concatenated into a single long chain, apart from the disabled chains. The tests confirm that the chain connectivity is correctly described in the spec files, and the chain start/end marker flops are in the correct position in the serial chain readout.

- **Automation script to check the status of the tests run**

  A script was developed to check the status of the test and print that data in a excel sheet, this sheet will be mailed to the email id given in the input of the script.

  Here are the inputs, outputs and the command line of the script:

  Perl grep_content_and_excel.pl --region_path abc –partition_path xyz --id anjukatarmal@yahoo.in

  Inputs:

  - Absolute Path where the region level tests were run.
  - Absolute Path where the partition level tests were run.
  - Email id where we want to send the prepared excel sheet.

  First a script will take the region path and for each test directory, it will check if the log file exists.

  Then it will read the log file to grep the patterns like ‘Driver check expected packet Failed’ and ‘Driver check expected packet Passed’ etc. to identify the status of the tests and save the status “PASS” or “FAIL” in the hashes of hashes, where primary key will be region or partition name and secondary key will be the test name.

  These hashes will be used to populate the data in the excel sheet.

  Here perl script was used to implement the above logic, where Excel: Writer::XLSX package was used to enter the data in the excel sheet.

region name	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
a	PASS	PASS	PASS	FAIL	PASS	PASS	FAIL	PASS	logfile not found	FAIL	PASS	PASS	PASS	PASS	PASS	FAIL	FAIL	FAIL	FAIL	
b	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS	
c	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS	
d	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS	
e	FAIL	PASS	PASS	PASS	PASS	PASS	PASS	FAIL	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS	
f	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS	
g	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS	
h	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS	FAIL	FAIL	FAIL	FAIL	
i	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS	
j	PASS	PASS	PASS	PASS	PASS	FAIL	FAIL	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS	
k	PASS	PASS	PASS	PASS	PASS	FAIL	FAIL	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS	
l	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS	
m	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS	
n	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS	
o																				FAIL
p																				PASS
q																				FAIL
r																				FAIL
s																				FAIL
t																				PASS
u																				FAIL
v																				FAIL

Figure 3.3 Prepared excel sheet after automation with Regions and tests

- **Automation in silicon test sequence enablement**

  An automation script was developed to make the silicon test content enablement more efficient and less time consuming. Init sequences are delivered to post silicon team for content enablement on silicon. Previously Init creation used to take manual efforts to perform some checks and compare it with previous dies. This whole procedure was done in no. of iterations to get the final init sequences with zero failures.

  Following flow chart represents working of the automation script:

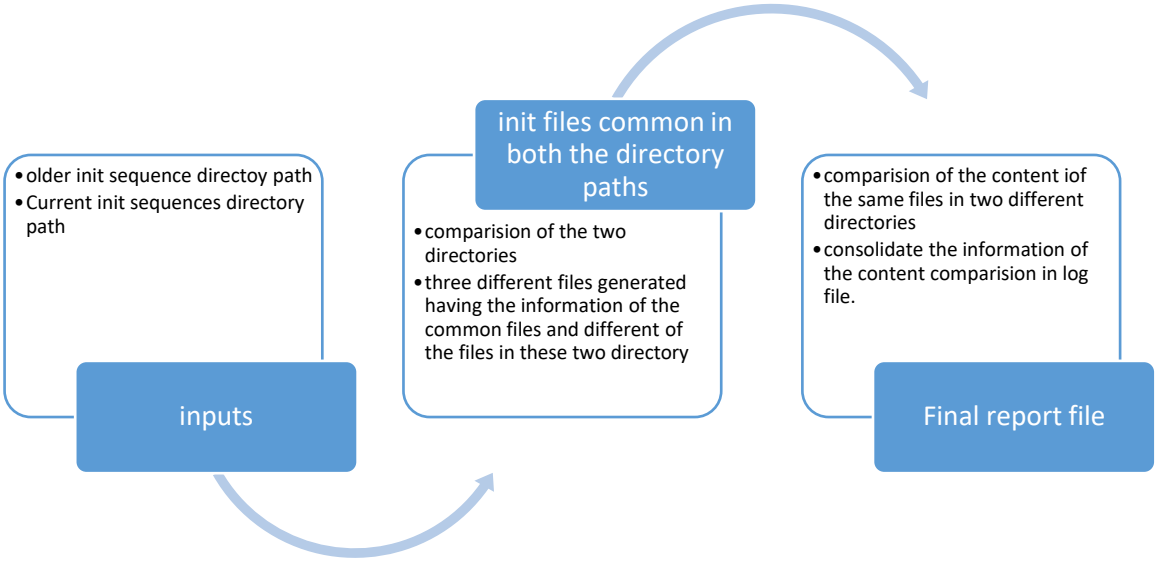

Figure 3.4 Flow chart of the working of the automation script for comparing two init sequences

Introduction of automation in silicon content enablement reduced the init sequence generation and delivery from manual efforts of 2-3 days to script execution of 10 mins. This way, it also helped in reducing human errors to almost zero.

Similarly, a script was developed to have a comparison of the configuration files of the different dies.

4. Simulation and Result discussion

4.1 Pattern Transformation: Single Shift Cycle Issue

Transformed patterns are tested by running regression. Here first full model reset test is run, during which the snapshot is taken. On this snapshot, main tests are run which usually takes 2-3 hours to be completed.

After the main test is run, perl script is used to analyze the log file to get the status of the patterns. Below figure 4.1 is the screenshot of the output of the perl script showing the expected patterns and actual patterns applied to the DUT for every cycle. Actual patterns are coming one clock cycle late from the expected ones as highlighted in the figure.

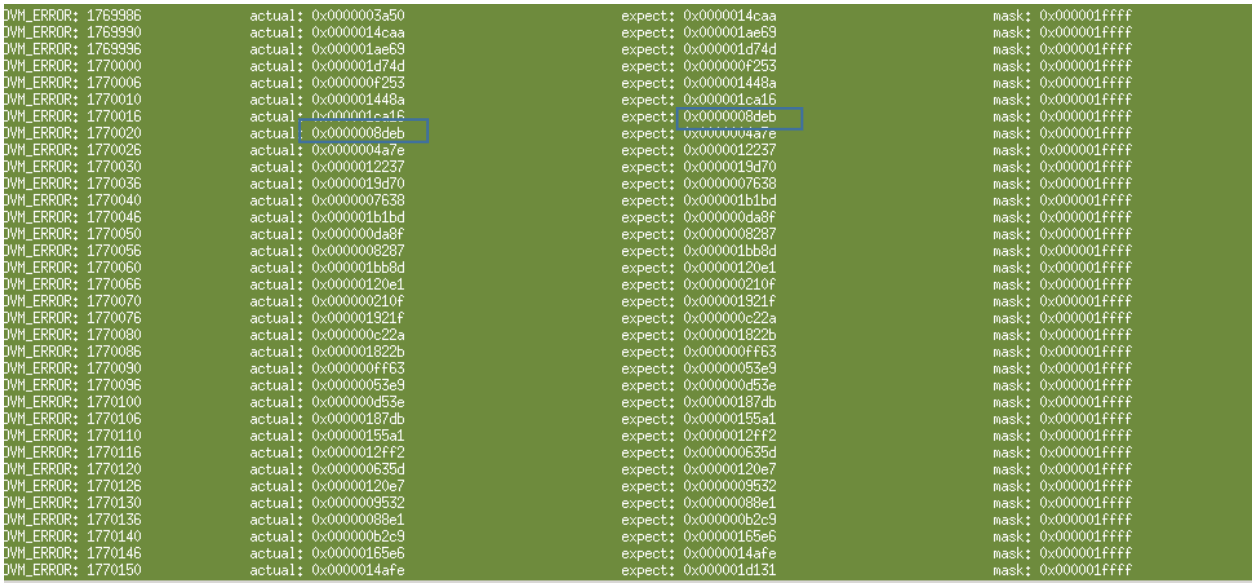

Figure 4.1 Single Shift Cycle pattern issue

Here figure 4.2 shows the output of the Perl script applied on the logfile of the simulation. It gives the total number of tests ran and their status.

Figure 4.2 Summary of the output of pattern test

As a solution we manually changed the latency to find if there is another issue than one cycle delay. And then the ATPG team was asked to generate patterns with changed delay.

4.2 Evil Validation:

Below figure 4.3 is the screenshot of CLKIN signal in the X regression getting X toggling after scan mode assertion. After further debugging and back tracing, it was due to one of the scannable signals which we are forcing deliberately X/random for the analysis. This issue is further reported and passed on to concerned team to take appropriate actions.

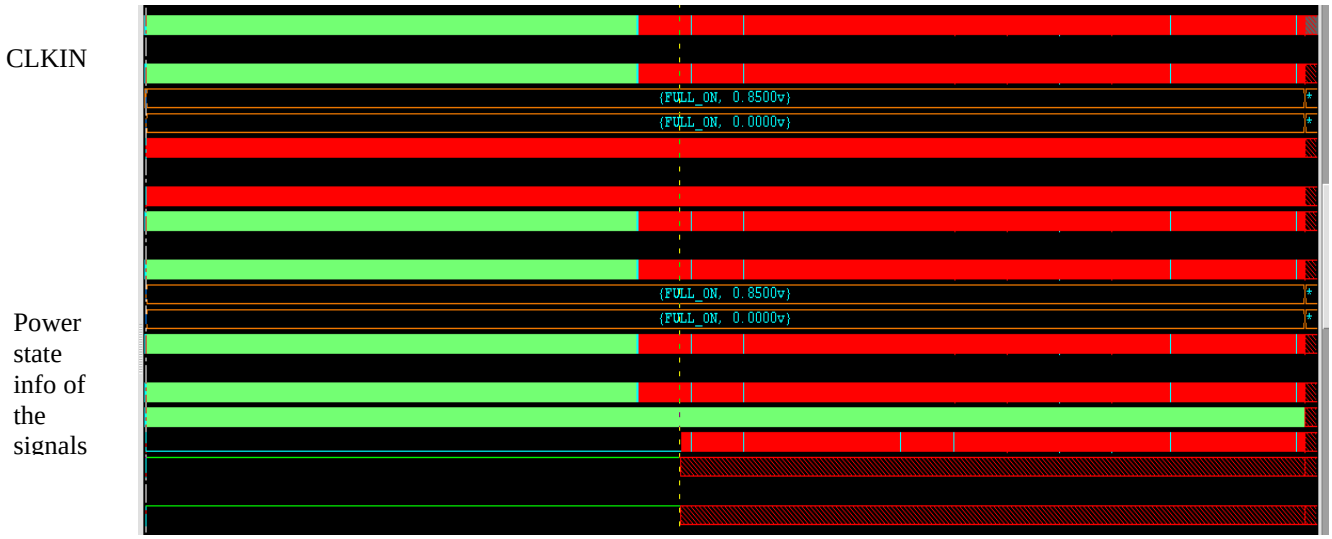

Figure 4.3 Evil Validation corrupted clk signal and driver signals

Zoomed in view of the clock in signal can be seen below in figure 4.4, X toggling after scan mode assertion.

Clkin

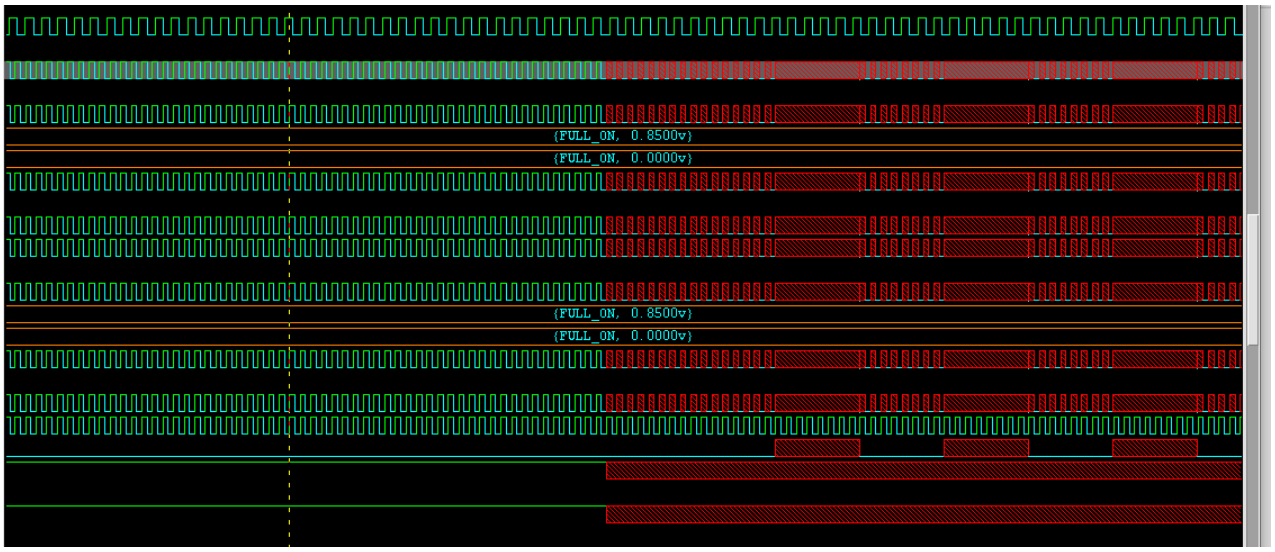

Figure 4.4 Evil Validation corrupted clkout signal with traced signals zoomed in view

Conclusion

In this work, pattern transformation flow to convert the block level netlist patterns to full chip level patterns was explored. Issues while testing these patterns such as single shift cycle were discussed. Evil validation required to check the impact of any scan cell corruption was explored and exercised in project. As a part of the ongoing team intiative, automation to get the status of the tests in excel sheet was implemented to improve the scan validation flow. Also, an automation script was developed to improve the silicon content enablement within lesser time and minimum discrepancies which reduced the time required for generation of silicon content to 2-3 hrs.

References

- [1] Prof Virendra Singh,Advance VLSI Design.NPTEL online course,Topic:"VLSI Testing:Design for Test(DFT),February 2016.[Online].Available: https://nptel.ac.in/courses/117/101/117101004/
- [2] "IEEE Standard for Test Access Port and Boundary-Scan Architecture", in IEEE Std 1149.1-2013 (Revision of IEEE Std 1149.1-2001) , vol., no., pp.1-444, 13 May 2013, doi: 10.1109/IEEESTD.2013.6515989.
- [3]Parker, Kenneth,(2003), The Boundary — Scan Handbook. 10.1007/978-1-4615-0367-5.
- [4] Gholami, M., Baleghi, Y. & Ardeshir, G. Design of Novel Testable and Diagnosable Phase-Frequency Detector. *Circuits Syst Signal Process* 33**,** 999–1018 (2014).
- [5]"IEEE Standard for Access and Control of Instrumentation Embedded within a Semiconductor Device," in IEEE Std 1687-2014 , vol., no., pp.1-283, 5 Dec. 2014, doi: 10.1109/IEEESTD.2014.6974961.
- [6]I. Indino and C. MacNamee, "DFT: Scan testing issues and current research," 25th IET Irish Signals & Systems Conference 2014 and 2014 China-Ireland International Conference on Information and Communications Technologies (ISSC 2014/CIICT 2014), Limerick, 2014, pp. 227-232, doi: 10.1049/cp.2014.0690.
- [7] F. G. Zadegan, G. Carlsson and E. Larsson, "Robustness of TAP-based scan networks," 2014 International Test Conference, Seattle, WA, 2014, pp. 1-10, doi: 10.1109/TEST.2014.7035321.
- [8] H. Ahrens, R. Schlagenhaft, H. Lang, V. Srinivasan and E. Bruzzano, "DFT Architecture for Automotive Microprocessors using On-Chip Scan Compression supporting Dual Vendor ATPG," 2008 IEEE International Test Conference, Santa Clara, CA, 2008, pp. 1-10, doi: 10.1109/TEST.2008.4700600.
- [9] N. Reddy, S. Menon and P. D. Joshi, "Validation Challenges in Recent Trends of Power Management in Microprocessors," 2020 IEEE International Symposium on Defect and Fault Tolerance in VLSI and Nanotechnology Systems (DFT), Frascati, Italy, 2020, pp. 1-6, doi: 10.1109/DFT50435.2020.9250842.