At-Speed ATPG aware scan stitching

Major Project Report

Submitted in partial fulfillment of the requirements
For the degree of
Master of Technology
In
Electronics & Communication Engineering
(VLSI Design)
By

Dipen Dave
(15MECV07)

Electronics & Communication Engineering Department
Institute of Technology
Nirma University
Ahmedabad-382481
May 2017

At-speed ATPG aware scan stitching

Major Project Report

Submitted in partial fulfillment of the requirements

For the degree of

Master of Technology

In

Electronics & Communication Engineering

(VLSI Design)

By

DIPEN DAVE

(15MECV07)

Under the Guidance of

Internal Guide	**External Guide**
Prof. (Dr.) Usha Mehta	Mr. Pradeep Bharadwaj
Professor, (EC Dept.)	Component Design Engineer
ITNU – Nirma University	Intel Technologies Pvt. Ltd.

Electronics & Communication Engineering Department

Institute of Technology

Nirma University

Ahmedabad-382481

May, 2017

DECLARATION

This is to certify that,

a. 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.

b. Due acknowledgment has been made in the text to all other material used.

Dipen Dave
(15MECV07)

Certificate

This is to certify that the Major Project report entitled **"At-speed ATPG aware scan stitching"** **s**ubmitted by **Dipen Dave** (15MECV07), towards the partial fulfillment of the requirements for the degree of **Master of Technology** in **VLSI Design** (Electronics and Communication Engineering Department) of Nirma University, Ahmedabad is the record of work carried out by him 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 our knowledge, haven't been submitted to any other university or institution for award of any degree or diploma.

Prof. (Dr.) Usha Mehta
Internal Guide
Professor, EC Dept.

Prof. (Dr.) N. M. Devashrayee
PG Co-coordinator (VLSI DESIGN)
Professor, EC Dept.

Prof. (Dr.) D. K. Kothari
Head, EC Dept.

Dr. Alka Mahajan
Director, ITNU

Date:

Place: Ahmedabad

Certificate

This is to certify that the Major Project (Phase-I) entitled "**At-Speed ATPG aware scan stitching**" submitted by Dipen Dave (15MECV07), towards the partial fulfillment of the requirements for the degree of Master of Technology in VLSI Design, Nirma University, Ahmedabad is the record of work carried out by his under our supervision and guidance. In our opinion, the submitted work has reached a level required for being accepted for examination.

Mr. Pradeep Bharadwaj
Component Design Engineer
Intel India Technologies Pvt.LTD
Bangalore

Acknowledgement

I take immense pleasure in thanking of my thesis coordinator Mr. Abraham Jais, Silicon Architecture Engineer, Intel Technologies Pvt. Ltd, Bangalore, India, my Mentor Mr. Pradeep Bharadwaj V R, Component Design Engineer, Intel Technologies India Pvt. Ltd., Bangalore, for having permitted me to carry out this project work. Through-out the training, he had given me much valuable advice on project work which I am very lucky to benefit from.

I wish to express my deep sense of gratitude to my guide Dr. Usha Mehta, Institute of Technology, Nirma University for being a source of inspiration and for timely guidance during the project.

I would like to express my gratitude and sincere thanks to our Director Dr. Alka Mahajan, Head of Electronics and Communication Engineering Dr. D. K. Kothari for allowing me to undertake this thesis work and for his guidelines during the review process.

I would like to express my gratitude and sincere thanks to Mr. Amey Shah, Mr. Jagadeesh Kumar, for helping me to carry out my final year thesis work at Intel, and all other team members for supporting and guiding me during the thesis.

-With Sincere Regards,
Dipen Dave
15MECV07

.

Abstract

With today's design size in millions of gates and working frequencies in gigahertz range, testing has become an important and an integral part of the VLSI design flow. Metrics like Fault Coverage have become one of the important parameter to be considered while designing a chip. Complex and efficient machines like the ATPG and ATE are developed to reduce the time and efforts required in testing. Many fault models are developed to represent the physical defects. Though the traditional Stuck-at faults cover most of the defects, defects like high impedance metal, high impedance shorts, and crosstalk are not caught by traditional stuck at scan vectors. They are only detected by Transition fault model which is tested by At-Speed testing. Also with the decreasing transistor size, the number of these defects tends to increase. Hence At-Speed testing plays a crucial role in testing of modern day chips.

The test inputs to a chip are applied by using a Scan Chain in which the flip-flops are stitched to form a continuous chain. Most of current algorithms does the stitching to reduce the wire length of the inserted scan chain i.e. physical aware stitching. In this project, efforts are made to stitch the scan chain in such a way that it improves the at-speed coverage and also try to reduce the test patterns required for testing the chip.

Contents

Declaration **3**
Certificate **4**
Intel Certificate **5**
Acknowledgement **6**
Abstract **7**
Contents **8**
List of figures **9**
List of tables **10**
Abbreviations and Key terms **11**

Title	Page No.
1. Introduction	**12**
1.1 Need for At-Speed testing	12
1.2 Objective	12
1.3 Organization of report	13
2. Literature survey	**13**
2.1 Design For Testability	13
2.2 Transition Faults	19
2.3 Fault coverage and ATPG	19
3. At-Speed testing techniques	**21**
3.1 Launch On Shift	21
3.2 Launch On Capture	22
3.3 Hybrid testing	23
4. Scan flop coverage	**24**
4.1 Effect of scan flop correlation on coverage	24
4.2 Algorithm to get correlation array	25
5. Shadow flop technique	**28**
5.1 Technique explained	28

5.2 Shadow flop implementation 29

6. Scan chain reordering 31
6.1 Travelling Salesman Problem (TSP) 31
6.2 Types of TSP 32
6.3 Using TSP for scan chain reordering 33

7. ATPG execution and results 36
7.1 ATPG execution 36
7.2 Results 36

Conclusion and future work 38

References 39

List of Figures

Sr. No.	Page No.
2.1 Typical Scan Chain	15
2.2 Modes of Scan Chain	16
2.3 Boundary Scan Cell	17
2.4 Boundary Scan Architecture	18
2.5 Rate of change of coverage with number of patterns	20
3.1 Launch On Shift waveform	22
3.2 Launch On Capture waveform	22
3.3 Hybrid test architecture	23
4.1 Transition fault testing example	24
4.2 Correlation algorithm flowchart	25
4.3 Flowchart for other approach of correlation algorithm	27
5.1 Shadow flop technique	29
5.2 Shadow flop implementation flowchart	30
6.1 Travelling Salesman Problem example	31
6.2 TSP implementation flowchart	33

List of Tables

Sr. No.	Page No.
3.1 Comparison between LOS and LOC	23
7.1 ATPG results for normal run	36
7.2 ATPG results for shadow flop technique	37
7.3 ATPG results for partial shadow flop technique	37
7.4 ATPG results for shadowing based on correlation data	38

Abbreviations

DFT	Design For Testability
ATPG	Automatic Test Pattern Generator
ATE	Automatic Test Equipment
LOS	Launch On Shift
LOC	Launch On Capture

Key terms

At-Speed - Combinational Clock speed

ATPG Aware – Coverage aware

Correlation between flops – Number of combinational cells shared by the flops

Chapter 1

Introduction

The at-speed testing for transition faults is an important and integral part of VLSI IC testing. No chip is passed by the chip fabricator until the required at-speed testing coverage numbers are met. In this chapter we will look at the need for at-speed testing and the objective for this project.

1.1 Need for At-Speed testing

As advancements are made to design, development and fabrication of ICs, chips are made to run faster and are designed to contain more and more transistors on a single chip having high packing density. The drive for faster performance and System-On-Chip (SOC) structure pushes the boundary of IC fabrication, reducing the transistor size to Deep Sub-Micron (DSM) technologies. This ever-shrinking transistor size has caused the emergence of new faults like the resistive via, high impedance via etc. These faults have a tendency to increase with the decreasing transistor size. The gates with these types of faults exhibit behavior of slow data transistors i.e. the gate output is correct but is delayed. The gate makes a transition but just not at the required speed. Due to this, there may be errors occurring in transitions at that gate. To detect these kinds of faults, a new dft technique is required that tests the device at functional speed. This technique is the At-Speed testing technique. The transition fault model is required for this test which is different from the traditional stuck-at faults. In this fault model, the gate is modeled to detect the transition faults i.e. 1 → 0 and 0 → 1 transition. The ATPG tool running this type of modeling creates test patterns to launch a transition and capture the response at the gate output and shifts through the scan chain. Other faults like the path delay faults which are prominent in today's complex chips can also be detected similarly by the at-speed testing. Hence for a thorough testing of the chip, the at-speed is necessary.

1.2 Objective

The scan chain is a prominent dft technique used in vlsi testing. The flip flops are converted to scan flip flops and a chain is formed connecting these scan flip flops. This process is called scan stitching. There are various factors governing this stitching process like the power domain of the scan flops, scan clocks associated with the scan flop etc. The synthesis tool which performs the stitching mostly stitches the scan flops in same power domain and having the same scan clock in one chain. Also the metric the tool considers while stitching is wire length of the scan chain. It tries to reduce it as low as possible so as to reduce the area and the delay overhead added due to stitching. Because of this, it stitches the scan flops near to each other in one chain. This may create a problem while testing. When the test patterns are applied to detect a fault at a combinational cell, if the cell is shared between two scan cells, there is a high chance that the cell is not controllable and the fault cannot be tested there. Thus the fault becomes redundant and

decreases the coverage for the design. To avoid this, the scan chain should be stitched in such a way that if here are combinational cells common between two scan flops, they should be stitched in different chains to avoid the development of above case. Also the added advantage is that the controllability of the cells increases so the test patterns can be generated more effectively and the number of patterns required to test the chip can be reduced.

1.3 Organization of report

Chapter 2: Literature survey – The basic concepts of dft like the test point insertion, scan chain, boundary sealing are explained. Along with that, topics like fault coverage, ATPG and transition faults and their testing are explained.

Chapter 3: At-Speed testing techniques –The different techniques for at-speed testing along with their advantages and disadvantages are explained.

Chapter 4: Scan flop correlation - The algorithm used to get the correlation between two scan flops i.e. to find the common combinational cells between them is explained.

Chapter 5: Shadow flop technique – The shadow flop technique and its implementation is discussed in this chapter along with its advantages and disadvantages

Chapter 6: Scan chain reordering – The travelling salesman problem and how its used for scan chain reordering is explained.

Chapter 7: ATPG execution and results – The coverage results for different run configurations are put.

Conclusion and future work

Chapter 2

Literature Survey

The testing of a chip is made easier by the process called Design For Test (DFT) and different fault models. There are many methods that can be implemented for reducing the testing efforts and improve the coverage numbers. Some of them are scan chain insertion, core-wrapping, boundary scan architecture (JTAG) etc. Some of them are discussed below along with the transition fault model and some basics of Automatic Test Pattern Generation (ATPG).

2.1 Design For Testability

Design For Testability (DFT) is a technique which makes the design testable after production. It's the extra synthesizable logic which we put in the design, during design process, which helps the post-processing testing. Post-processing testing is necessary and important because the process of manufacturing is not 100% error free. There are always some defects in silicon which contribute towards errors introduced in physical device. To make the detection of these defects easier, DFT is employed. Also testing of a chip takes up around 70%of the time from RTL to final packaging. DFT also makes the testing faster and hence helps in reducing the Time To Market for a chip.

There are two types of DFT techniques:

a) **Adhoc DFT approach:** Adhoc DFT techniques involve application of good design practices gained through years of experience or replacing a bad design practice with a good one or a better one.
   i) **Test Point Insertion:** Test Point Insertion (TPI) is a commonly used ad-hoc DFT technique so as to improve the controllability and observability of internal nodes. Testability analysis is used for identification of the internal nodes where test points should be inserted in form of control or observable points.
   ii) **Partitioning:** Another common method used in ad-hoc dft is partitioning. The entire design is divided into smaller testable blocks. This is done to reduce the complexity of testing larger designs. The smaller testable blocks are connected through muxes to each other.

   The difficulty with ad hoc testability techniques is that it requires additional control inputs and observation outputs. So it is not used nowadays.

b) **Structured DFT approach:** This approach improves the overall testability of a circuit with a test-oriented design methodology. This approach is methodical and systematic and gives predictable results. It gives access to internal nodes of a circuit without requiring a separate external connection for each node accessed. But this comes at a cost of additional internal

logic circuitry used for testing. So there is an added area overhead. The most widely used techniques are:

i) Scan Chain: Scan design is most preferable structured DFT methodology. It helps to improve the overall testability of a circuit by improving the controllability and observability of storage elements in a sequential design.

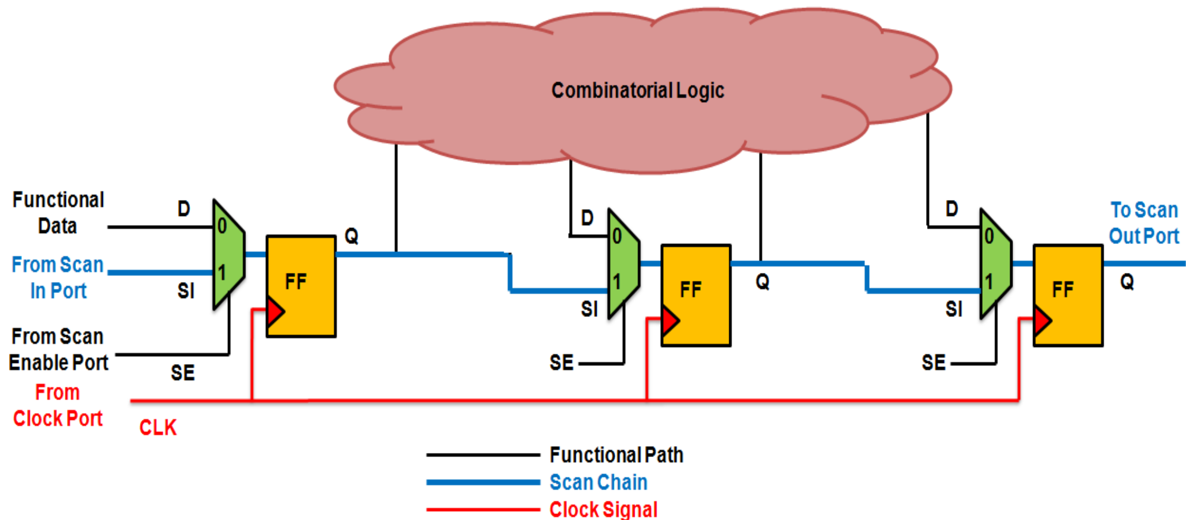


Fig 2.1 Typical Scan Chain

The scan design is a novel technique in which all the D flip-flops in the design are converted into Scan Flip Flops (SFF) by adding a multiplexer at the input D pin on the flip flop. The SFFs are stitched in a manner of a chain such that the output of one is connected to the input of other. This chain of the SFFs is a scan chain. One input of the multiplexer is the output of the combinational logic (highlighted in black in above figure). The other input is the scan path (highlighted in blue in above figure).

While forming the scan chain, the following ports are also created:

i. Scan In (SI)
ii. Scan Enable (SE)
iii. Scan Out (SO)
iv. Test mode

The SI and SO ports are the input and output ports of a scan flop respectively. The SE signal is used to control the shifting of test patterns and capturing the response of combinational logic. The test mode signal signifies whether the design is working in functional mode or in test mode. 0 for functional and 1 for test mode.

The scan path is the connection from SI of a scan flop to SO of another scan flop. The testing is done by applying test patterns to the combinational logic and then capturing their output. These test patterns are applied by the scan path. They get shifted through the chain.

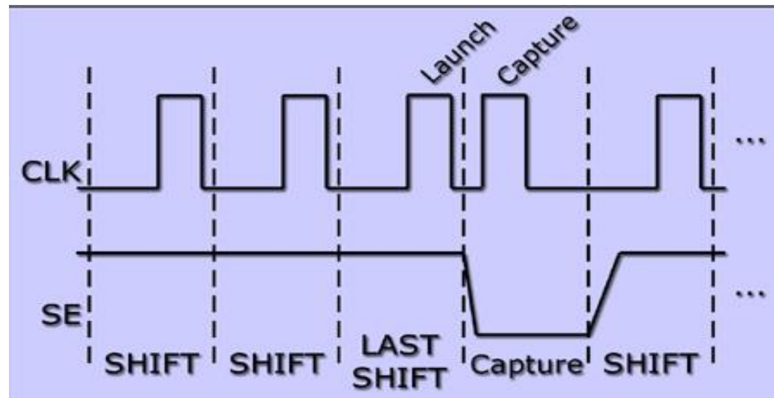


Fig. 2.2 Modes of Scan Chain

Based on the value of the SE signal, the scan chain can work in different modes as shown in the figure above:

Shift: This is the mode where the shifting of test pattern occurs. The SE signal is high during this process and hence the scan path is active.
Launch: In this mode the test patterns are applied to the combinational logic. The SE remains low during this process.
Capture: The output of the combinational logic exited by the test patterns is captured in the capture mode. The SE is low in this mode and hence the functional path is active in this mode.

ii) **Boundary scan:** The boundary scan is a special type of scan path with a register added at every I/O pin on a chip. It is an IEEE Standard: IEEE Std 1149.1 – 1990, IEEE Standard Test Access Port And Boundary-Scan Architecture. It was started by a group known as Joint Test Action Group also known as JTAG. Around 200 major electronics companies like IBM, TI, Siemens, Philips, AT&T etc. are a part of the development of this standard.
The boundary scan creates a scan path around the input and output pins of the device for increasing the controllability and observability through scan operation. Another important benefit of the boundary scan architecture is that it provides isolation to the part of the design it is applied on, from the rest of the design. So each partition which has the boundary scan architecture applied can be tested separately without affecting the function of the other partitions in the design. External testing of wiring interconnects and neighboring ICs on a board assembly is accomplished by applying test stimulus from the output BSCs and capturing test response at the input BSCs.

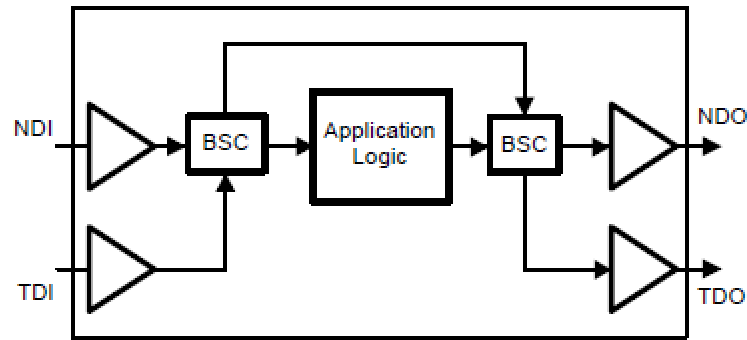


Fig 2.3 Boundary Scan Cell

The above figure shows the Boundary Scan Cell (BSC) applied to the functional logic represented as application logic. The BSCs make a separate chain around the core or functional logic from the normal scan chain containing scan flops. The required signals are:

NDI: Normal Data In – It is the functional data input pin.

NDO: Normal Data Output – It is the functional data output pin.

TDI: Test Data In – It is the test input pin where the test patterns are applied.

TDO: Test Data Out – It is the test output pin where the response of the test patterns applied is captured.

The TDI and TDO for a boundary chain are separate from the internal scan chain's test input output pins. The BSCs are interconnected to each other to form a chain such that the TDO of one BSC connects to the TDI of the next BSC. During functional mode operation, the BSC is transparent and the input and output signals pass through it freely. During the test mode, the test patterns are shifted through the BSC chain until they reach their required point. They are then applied and the response is captured and shifted at the output port for inspection.

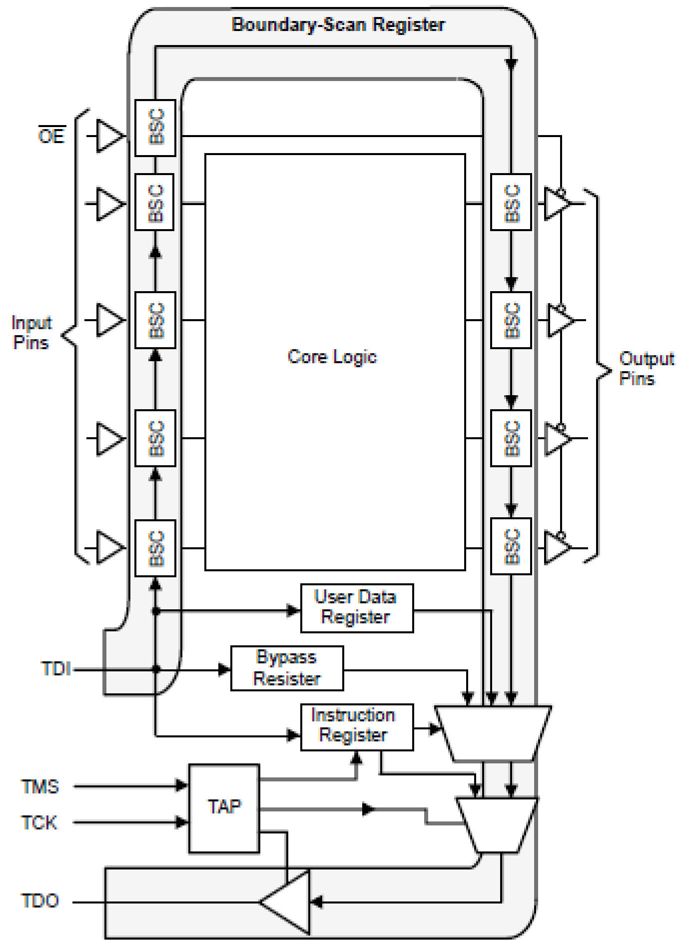


Fig 2.4 Boundary Scan Architecture

Figure shows the IEEE Std 1149.1 architecture. The architecture consists of an instruction register, a bypass register, a boundary-scan register (highlighted), optional user data register(s), and a test interface referred to as the test access port (TAP). In the figure, the boundary-scan register (BSR), a serially accessed data register made up of a series of boundary-scan cells (BSCs), is shown at the input and output boundary of the IC. The instruction register and data registers are separate scan paths arranged between the primary test data input (TDI) pin and primary test data output (TDO) pin. This architecture allows the TAP to select and shift data through one of the two types of scan paths, instruction or data, without accessing the other scan path.

2.2 Transition faults

The transition faults are the faults that occur in the transitions of a signal in a design. They occur due to various defects that are produced during the manufacturing process. The rise time or the fall time of a signal is affected at a defect with transition fault. They are of two types:

i) **Slow to Rise:** The slow to rise fault at a defect indicates a faulty 0 → 1 transition i.e. fault in rise time of a signal at the maximum operating speed of the device.
ii) **Slow to Fall:** Similarly, the slow to fall fault at a defect indicates a faulty 1 → 0 transition i.e. fault in fall time of a signal at the maximum operating speed of the device.

The testing of one transition fault requires two test patterns to be applied. They are namely:

Initialization pattern: This is the first pattern to be applied to test a transition fault. For a 0 → 1 transition, the initialization pattern would be 0 and similarly for a 1 → 0 fault, it would be 1.

Launch pattern: The launch pattern is applied after the initialization pattern to test a transition fault. Its value will be opposite to the initialization pattern.

The response of the launch pattern of the chip must be captured at functional speed.

2.3 Fault coverage and ATPG

It is the percentage of faults detected during testing of a chip. It is given by the following relation:

Fault coverage (%) = (Number of faults detected/ Total faults) * 100

Higher the coverage, better is the testing considered. Some of the faults are not detectable by applying test patterns. They are called redundant faults. There is another type of coverage called test coverage which only considers the detectable faults. It is given by:

Test coverage (%) = (Number of faults detected / Total detectable faults) * 100

The detectable faults are given by (total faults – redundant faults).

The faults are detected by applying the test patterns to the chip. The coverage increases with the number of patterns applied. Initially the rate of increase of coverage with number of test patterns applied is very high but goes on decreasing with time. Below is the graph showing relation between coverage and number of applied patterns.

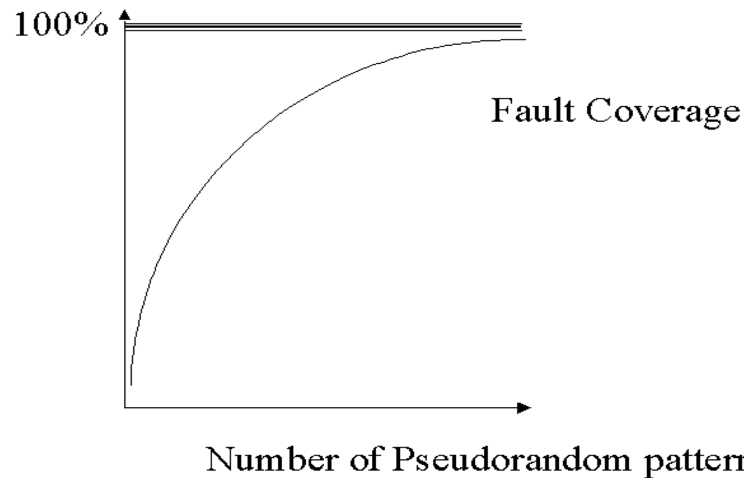


Fig. 2.5 Rate of change of coverage with number of patterns

The test patterns are generated and applied by an Automatic Test Pattern Generator (ATPG). The ATPG generates pseudo-random test patterns. In the starting of testing, random patterns are applied. The coverage increases rapidly initially. But as more and more faults get tested, the rate of fault detection decreases. So once the achieved coverage reaches around 60%, targeted testing is done. A fault is selected from the list of remaining faults and a test pattern is generated and applied to check for it. The advantage of this targeted testing is reduction is test patterns.

The ATPG uses complex algorithms for pattern generation and application to achieve the desired coverage. Some of them are:

i) D Algorithm
ii) Path-Oriented Decision Making (PODEM)
iii) Fan-Out Oriented (FAN Algorithm)
iv) Pseudorandom test generation
v) Wavelet Automatic Spectral Pattern Generator (WASP)

Chapter 3

At-speed testing techniques

The transition fault testing is done through the at-speed testing as seen in earlier chapter. For testing these faults, two patterns, namely the Initialization pattern and the Launch pattern have to be applied and the response has to be captured at functional speed. Based on how these two patterns are applied, there are two methods of at-speed testing:

i) Launch On Shift (LOS)
ii) Launch On Capture (LOC)

Also both the methods can be combined in the same design to do Hybrid testing in which the scan chains can be selected to be tested in LOC or LOS mode.

3.1 Launch On Shift

The LOS is also known as skewed load technique.

In this, the initialization pattern is applied during initialization cycle (IC). The launch pattern to launch the transition is applied at the Launch Cycle (LC) which is the last shift cycle in this case. The capturing of the response happens at functional clock speed. So the difference between the rising edge of the LC and rising edge of the CC must be equal to that of the functional clock period. The Scan Enable (SE) is high for the IC and the LC but should be low for the CC. So the SE should make a 1 → 0 transition at functional clock speed as shown in below figure. The SE goes to all the scan flops and hence has to be routed separately if it is to make an at-speed transition. This increases the time and efforts required in routing. Apart from this disadvantage, the LOS has advantages too. The coverage obtained from LOS testing is always more than the LOC. Also the number of test patterns required is less in LOS than the LOC.

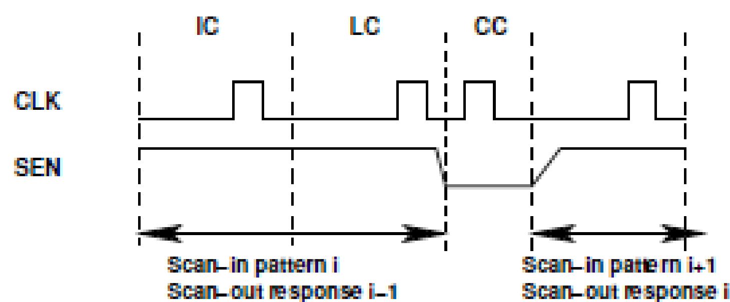


Fig. 3.1 Launch On Shift waveform

3.2 Launch On Capture

The LOC testing is also known as Broadside testing.

In this method, the launch cycle is not a part of the shifting operation as in LOS. The fault is activated or initialized during the initialization cycle. The scan enable then goes low and this makes the functional path active as seen in the below figure. So the patterns applied during the launch cycle for testing are the output of the combinational circuit at the fanin of the scan flop. Two at-speed clock pulses are applied, one for the launch and the other for the capture cycle. But here, unlike the LOS, there is no requirement of the SE to make a transition at functional clock speed because the SE is already low before the launch cycle. The LOC test gives less coverage than the LOS counterpart and also the patterns required as more than that of the LOS. But it does not have the necessity for an at-speed scan enable signal so the time and efforts for routing the SE are saved.

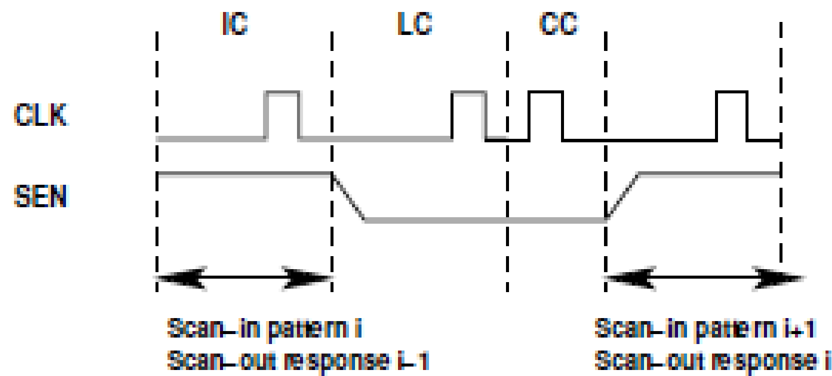


Fig. 3.2 Launch On Capture waveform

LOS	LOC
• Gives more coverage than the LOC technique	• Less coverage compared to the LOS
• Number of test patterns required are less	• More test patterns required than the LOS technique
• Needs an at-speed scan enable signal	• Does not require an at-speed scan enable signal
• More efforts required for scan enable routing	• No additional routing required for scan enable
• Also known as skewed load technique	• Also known as broadside technique

Table 3.1 Comparison between LOS and LOC

3.3 Hybrid test

The hybrid test is a combination of the Launch On Shift and the Launch On Capture techniques. Here, some of the scan chains work in the LOS mode while the rest are made to work in the LOC mode. The advantage here is that the timing for the at-speed scan enable signal needs to be only closed for the chains in the LOS mode. This reduces some efforts for the at-speed scan enable required for LOS testing.

The scan flops are added with an additional combinational logic to select the mode of testing i.e. LOS or LOC. To select the mode of testing, control signals are generated viz. Scan Test Select (STS). By combination of the STS1 and STS2, the mode of testing can be full LOS, full LOC, hybrid etc. A control cell is placed in each scan chain to select the mode that scan chain will be operating in.

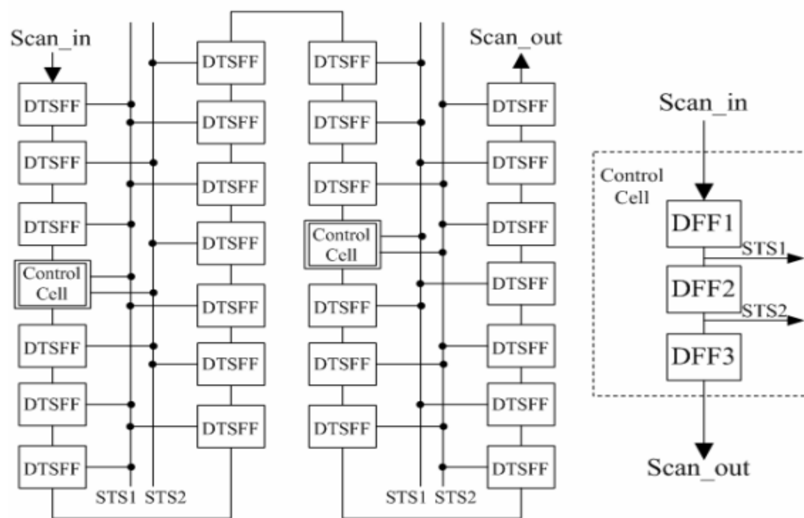


Fig. 3.3 Hybrid test architecture

Chapter 4

Scan flop correlation

The main objective of this project is to reduce the unobservable faults by improving the controllability and observability of the design. The combinational blocks common between two flops leads at-speed to coverage loss. The correlation data for all the flops is gathered and the algorithms and techniques for finding the numbers are explained in this chapter.

4.1 Effect of scan flop correlation on coverage

The scan-stitching and reordering done during the synthesis and place and route processes is based on many factors like clock-domain, power-domain, tool algorithm etc. The algorithms that govern these processes are mostly physical driven i.e. the stitching and reordering processes are done such that the wire length and hence the delay is minimum in the inserted scan chain. So a scan flip flop may most probably get stitched to an adjacent scan flop or a scan flop closest to it within the same clock and power domain to reduce the extra wire length added by scan stitching process. This is done to meet the timing requirement of the design which is very crucial. The insertion of scan chain and other DFT components cause an extra timing overhead. By this approach the timing of the design is met, but there are other factors which get affected, example the fault coverage.

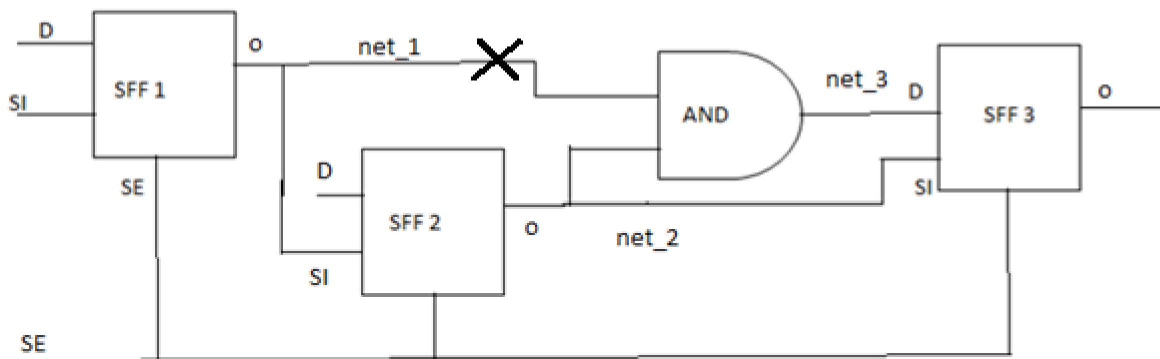


Fig. 4.1 Transition fault testing example

The transition test requires two test patterns to be applied for detecting one fault: the Initialization and the Launch pattern. Consider two adjacent scan flops in the same scan chain driving the same combinational cell as shown in figure below. Suppose a 0 → 1 transition fault is to be tested at net_1. The initialization pattern is 0 and launch pattern is 1. During both, the initialization and the launch pattern, the SE remains high and hence the scan path is active during this time. For the initialization cycle, the SFF1 must be first loaded with 0. So 0 will be applied to net_1. To launch the 0 → 1 transition fault, 1 must be applied at net_1 is the next clock cycle.

At this clock cycle, 0 is applied to net_2.Though the transition occurs at net_1, the second input to the AND gate is 0 and hence at the next clock cycle the fault does not propagate to the next level and hence becomes undetectable or redundant. Such scenarios can occur very frequently in larger designs where the number of combinational cells shared between two scan flops is very high (in hundreds). Due to such cases where the fault becomes untestable, the at-speed coverage reduces.

To solve this problem, the 2 SFFs must be stitched in different scan order. This will improve the controllability, which will lead to increased transition fault testing and hence help in increasing the at-speed coverage. The combinational cells common between the two SFFs after re-stitching should be 0 or at least less than the common combinational cells before re-stitching. So it is necessary to get the correlation between all the SFFs in the design for proper re-stitching.

4.2 Algorithm to get correlation array

A code was developed to get the correlation between all the SFFs of design using Synthesis tool commands and TCL scripting. The algorithm is as follows:

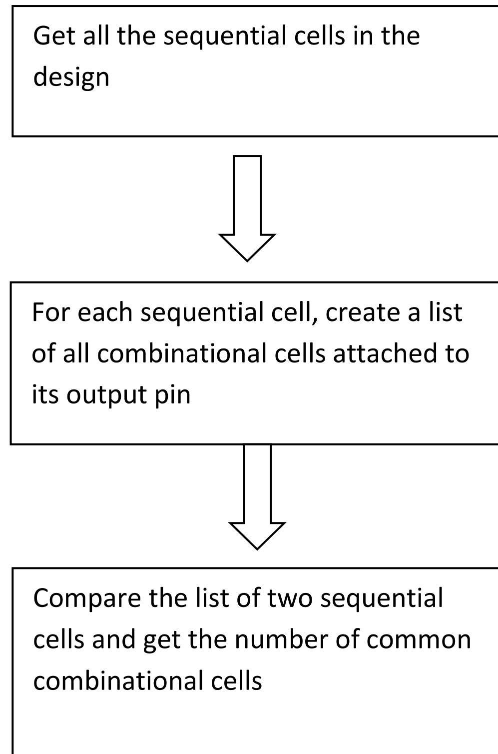


Fig. 4.2 Correlation algorithm flowchart

Initially for a design with 133722 combinational cells and 35069 sequential cells, the runtime of this code was more than 4 hours. To make the code runtime efficient, the approach of the algorithm was changed. Along with this, some techniques of reducing runtime were tried. Some of them are:

i) **TCL Multithreading:** TCL provides the ability of multithreading. The main concept is that there can be multiple tcl interpreters per thread, but one interpreter should only be used in one thread i.e. the thread that created it. So a main thread can run while other threads can run parallel in background performing other tasks. For enabling the multithreading feature, the thread package has to be downloaded and installed. The threads can even communicate with each other by passing data. Each thread will have its unique process id. This is how we can identify the threads and know if they are still running or have been completed. A thread created has to be destroyed after it gets over. It doesn't get destroyed by itself.

   Some of the TCL multhreading commands are:

   - **thread::create**: Creates a thread containing tcl interpreters.
   - **thread::id**: Returns the ID of the current thread.
   - **thread::release**: Destroys the thread with the given ID.

   This approach was only limited to the processing in tcl and not the tool processes. Hence it was not used as a better alternative was available.

ii) **Multiple tool shells:** By launching multiple tool shells at the same time, the process to be run can be highly parallelized. There is a direct runtime reduction by a factor of number of shells launched.

iii) **Using basic TCL commands:** Tcl commands were used instead of the tool commands wherever possible. The tcl commands take lesser time to be interpreted than the tool command to be implemented. This makes a great impact on the overall runtime when same things have to be done multiple times.

New approach: The problem with the first approach was that the number of combinational cells for each sequential cell was fairly high. So comparing the list of combinational cells for each sequential cell becomes very time consuming. So instead the approach used was to get all the combinational cells and make a list of sequential cells connected to its input pins. Then the correlation array was made using the lists. So instead of the exponential processing, the problem was changed to a larger linear problem. Also, multi-processing was done by opening multiple tool shells together and running the code parallel in all the shells.

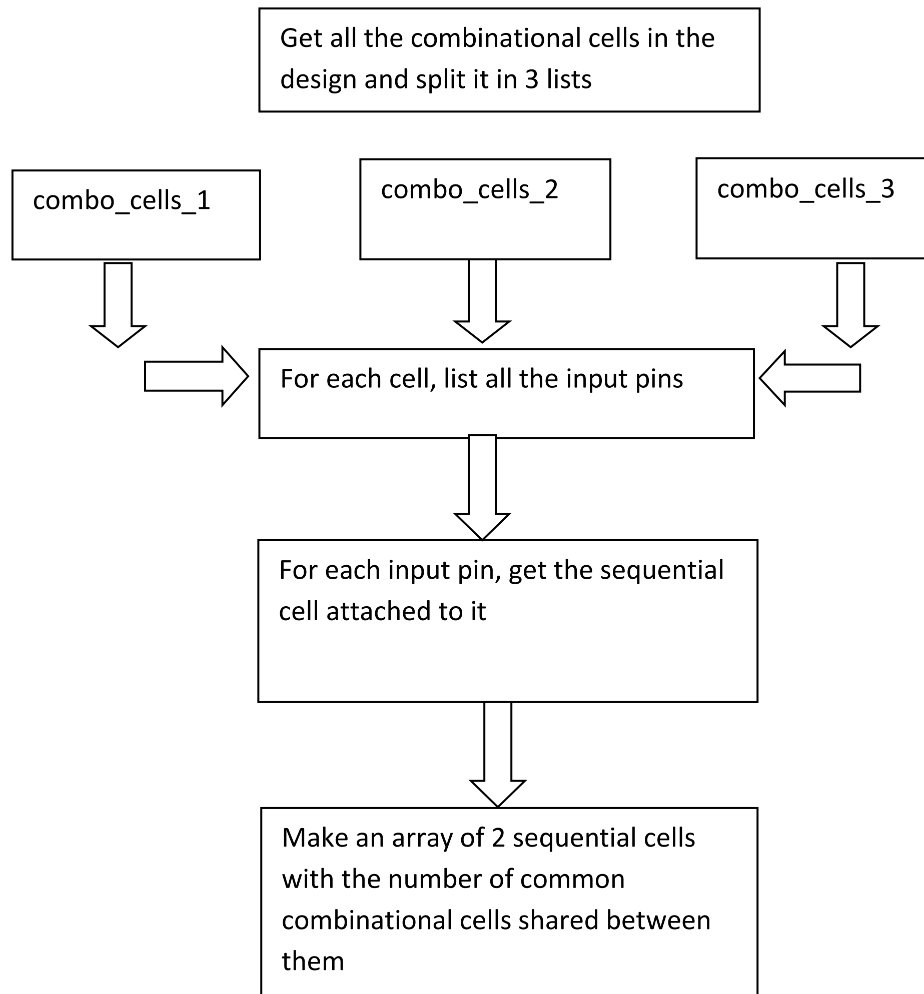


Fig. 4.3 Flowchart for other approach of correlation algorithm

The runtime of the code after applying all the techniques was more than 2.5 hours.

The earlier approach in this project was to stitch all the flops in only 1 chain. This makes the ATPG enablement easy. So for finding the correlation data, all the flops were considered. But later, the approach was changed to perform reordering on the existing solution given by the synthesis tool. So in the later approach, only the scan flops from the same scan chain were considered for the correlation data. This reduced the computation greatly and with it the runtime of script for finding the correlation between the scan flops. The runtime with this approach was about 35 minutes.

Chapter 5

Shadow flop technique

This technique adds extra scan flops before each scan flops in the design to improve the controllability of the scan flops. This is to reduce the dependency of the common combinational logic between two scan flops during the launch cycle. The technique was implemented in synthesis tool. The technique, its use and its implementation is explained in this chapter.

5.1 Technique explained

As discussed in the previous chapter, a major part of at-speed coverage is lost due to the correlation i.e. combinational block common between scan flip flops of same chain. The combination block sometimes makes it impossible to capture the output from a net where a pattern is launched. One such example is given in figure 4.1.

One solution to overcome this loss of coverage is by using the shadow flop technique. In this technique, an extra scan flop is inserted before each scan flop in the design. The newly inserted flops are called as shadow flops as they shadow the flops already present in the design. Their tput is only connected to the scan in pin of the next flop and not to any combinational logic. The connection of the original scan flops in the design and the shadow flops is done in such a way that the original flops in the design can be made controllable for test i.e. any test pattern can be loaded and a proper capture can happen without any hindrance from the combinational blocks.

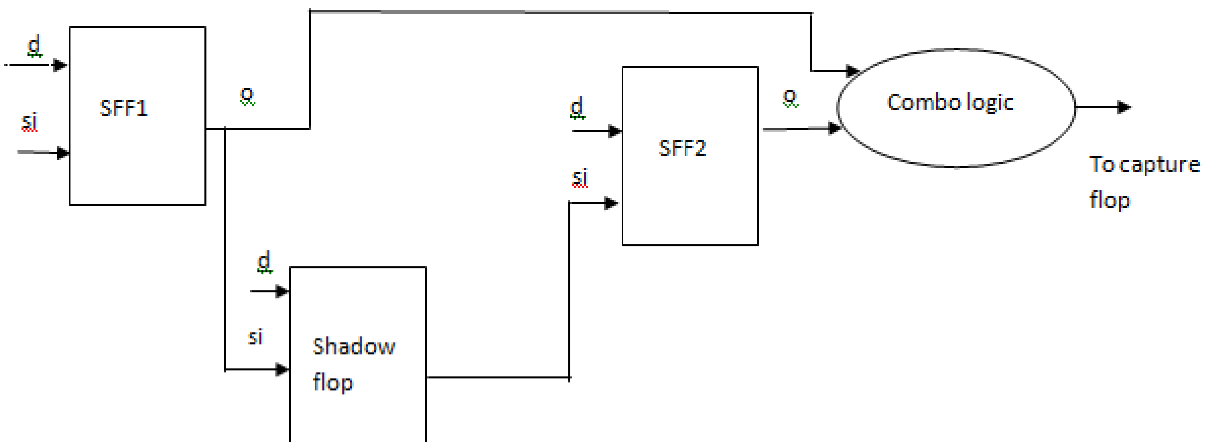


Fig 5.1 Shadow flop technique

Above figure shows shadow flop insertion for SFF_2.The output (o) of SFF_1 is connected to the ScanIn (si) pin of the shadow flop instead of ScanIn of SFF_2. Output of shadow flop goes to ScanIn of SFF_2. So the combinational block common between SFF_1 and SFF_2 can be controlled in a better way during testing by the insertion of the shadow flop.

This technique is limited to theory and experiment only and does not have practical usage in real time designs due to its huge added area and power requirements.

5.2 Implementation of Shadow flop technique

The implementation of the shadowing technique in the design was done during the synthesis process via a script. The following flowchart describes the implementation steps:

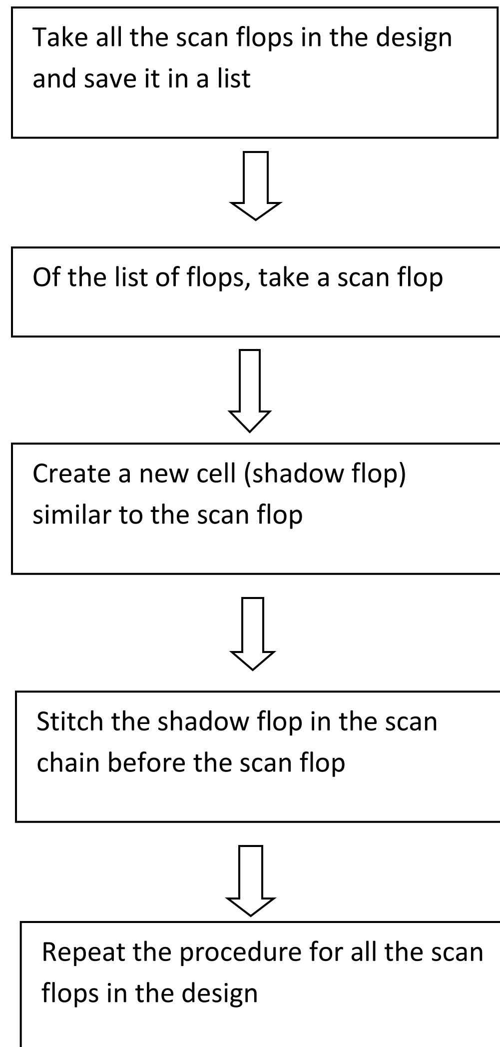


Fig 5.2 Shadow flop implementation flowchart

Chapter 6

Scan chain reordering

The scan chain insertion is done during the synthesis process. The normal d flip flops are converted to scan flip flops and stitched in a chain. The tool stitches the chain in a timing aware way i.e. it tries to reduce the wire length to improve the timing by reducing delays caused by the parasitic. This may affect the testing capability of the scan flops as seen in chapter 4 where the faults became unobservable. So the chains are stitched in a way to improve the coverage. The Travelling Salesman Problem (TSP) is the perfect fit for this. The implementation of the same is explained along with the theory and types of TSP algorithms.

6.1 Travelling Salesman Problem (TSP)

The Travelling Salesman Problem is a classic problem for optimization. It is a NP-complete problem and a hard optimization problem. The problem is based on a salesman who has to travel N cities. The order of travel is not important but each city should be travelled and the salesman should return to the city where he started at end of the trip. Each city is connected to other in various modes of transport and each mode has its own cost of travel. The cost can be anything from distance between the two cities or the actual cost of travel or a combination of different parameters. Basically it describes the difficulty of the travel. The main point of the solution is that the salesman should complete the trip in minimum cost. Each city is represented by a node and their connection as an edge.

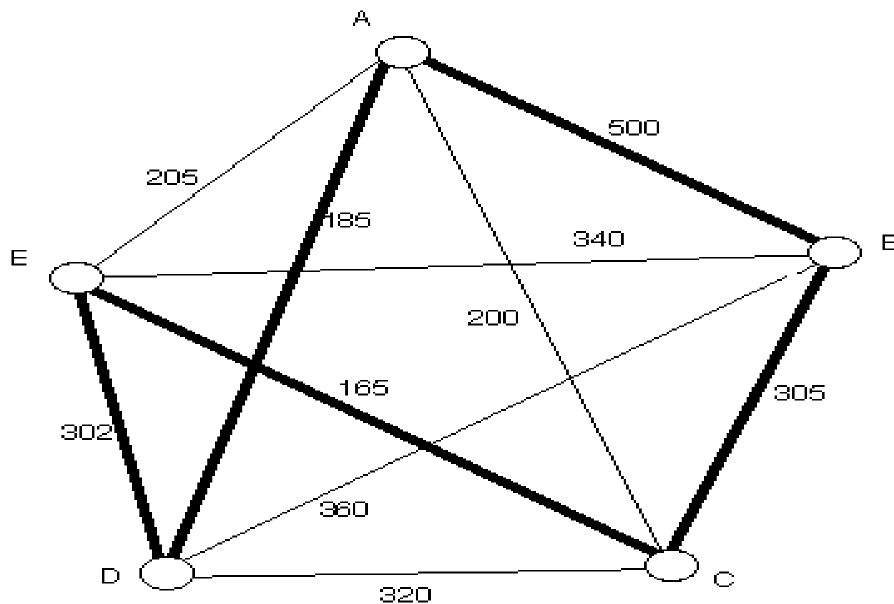


Fig 6.1 TSP example

In the above example, A B C D E can be considered as cities and the cost of travel between two cities is the number represented on the edge between the two cities. The path with the least cost is represented by darkened lines. If we consider the start of travel as A, then the order of travel can be either of the two:

A -> B -> C -> E -> D -> A or

A -> D -> E -> C -> B -> A

For both the paths, each city is travelled only once and the sum of the costs is the same i.e. 1437.

6.2 Types of TSP solutions

The problem statement defining the TSP is: For a given weighted graph G, the solution of TSP is the cycle covering all the vertices of G such that the cost of travelling the vertices is minimum. The TSP algorithm explained above can be used for reordering the scan chain. It is a natural fit for this reordering process as we are trying to reduce a cost function i.e. the correlation between the scan flops. It works in the following way:
Scan flops should be considered as cities and the cost function of travelling between the cities is the correlation between the scan flops that is explained in chapter 4. So once the stitching of the chains is done by the tool, we can reorder the chains on top of the existing scan order using the TSP algorithm.
The solution to TSP can be formed using both exact and heuristic approaches:
Exact solutions:

1. **Brute force approach:** This is the most basic approach for solving the problem. It works by trying the permutations possible and find the path with the least cost which will be the optimal solution for the problem. The biggest drawback for this approach is that it is not feasible for bigger problem sets. The runtime will be too high for larger data sets and mostly impractical for larger scenarios. On the other hand, it gives the best possible solution for smaller data sets.
2. **Branch and Bound algorithm (B&B):** The B&B algorithm systematically analyzes all possible solutions for the given problem but does not evaluate some possible solutions based on a condition. It makes use of lower and upper bounds value of the cost function or the quantity being optimized. The algorithm works by dividing the main problem into many sub problems. Each sub problem can have multiple solution. A solution of one sub problem can affect the solution of later sub problems and hence the overall solution. Each sub problem is evaluated and if its solution is found to be not meeting the bounding condition, it is not evaluated further. This approach saves the overall runtime of the algorithm.
   e.g. Suppose there is a problem to minimize a cost function. If the solution of a sub problem is less than the lower bound, that subset is not further evaluated.

Let S be some subset of solutions.
L(S) =a lower bound on the cost of any solution belonging to S
Let C=cost of the best solution found so far
If $C \leq L(S)$, there is no need to explore S because it does not contain any better solution.
If $C > L(S)$, then we need to explore S because it may contain a better solution.

Heuristic approach:

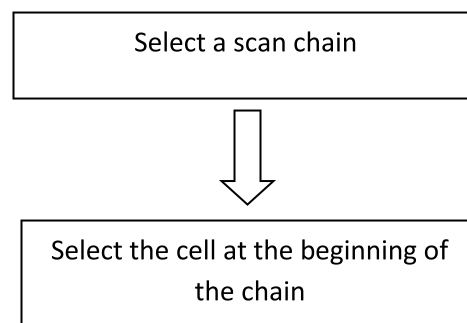
1. **Nearest Neighbor (NN):** The nearest neighbor or greedy algorithm works in a simple yet effective way. It forms the path by finding the next city which has the least cost function with the current one. It is a very quick way to find a solution for TSP having large number of vertices. The result won't be the optimal one but the runtime will be very less for large data sets. But the solution of the TSP by this approach will depend greatly on the start point. For different start point, the final solution will be different.
2. **K-opt heuristic:** This algorithm was proposed by Lin-Kernighan. It starts by deleting k mutually disjoint edges from the tour. The next step is to reassemble the fragments created into a path. The reassembling is a TSP on itself but with reduced data set. Each fragment can now be considered as a node and the path connecting all the fragments with least cost function can be found. Thus the problem is simplified to a great extent. The reassembling can also be carried out with brute force approach for best results. The quality and runtime of the solution depends on the number of edges deleted in the beginning i.e. k. Larger the value of k, better the solution but also larger the runtime and vice versa for lower value of k.

There any many other solution algorithm for solving TSP like:

i) **Christofide's algorithm for the TSP**
ii) **V-opt heuristic**
iii) **Ant colony solution**

6.3 Using TSP for Scan chain reordering

For this project, the Nearest Neighbor (NN) also called as the greedy algorithm is used. It is a very fast algorithm and gives a good solution though not optimal.
The algorithm works as shown in the following flowchart:



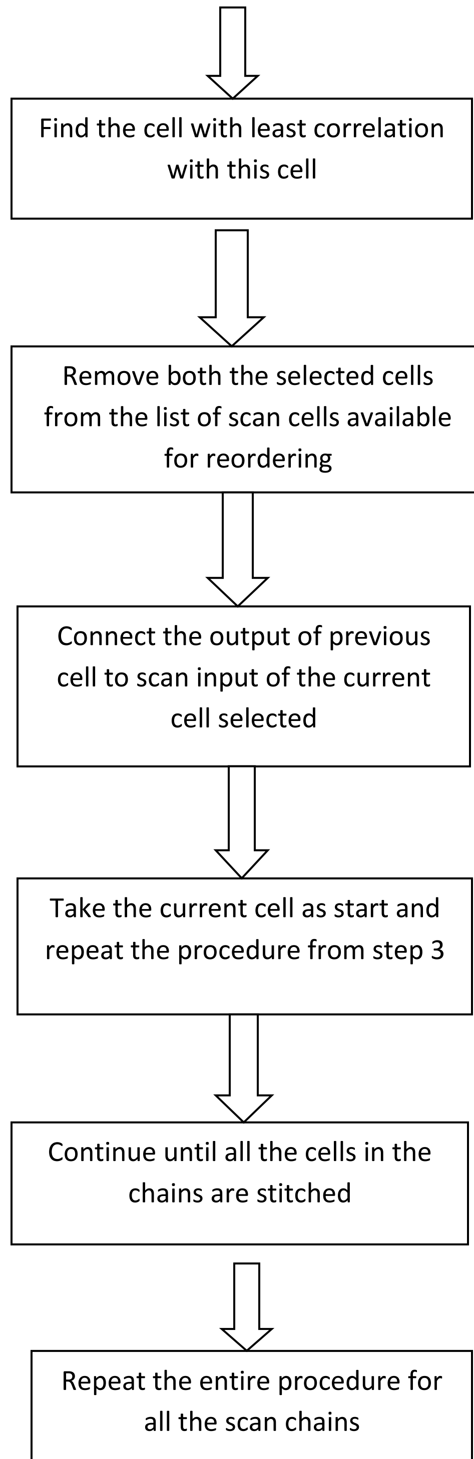


Fig 6.2 TSP implementation flowchart

Chapter 7

ATPG execution and results

7.1 ATPG execution

The ATPG was configured for at-speed testing in Launch on Shift (LOS) mode. This mode is for transition fault testing. The LOS mode requires a local scan enable signal which is generated from the global Scan Enable (SE) signal. This is done by pipeline scan enable cells which act as synchronizer cells and convert the global asynchronous SE n files were to local synchronous SE signal. This reduces the need for high speed SE signal for LOS testing. This procedure was done in synthesis step and a corresponding netlist was generated on which ATPG pattern generation was performed. Other ATPG inputs such as .do files and configuration files were generated accordingly.

7.2 Results

The details of the design on which experiments were performed are as follows:
Combinational Cell Count: 133722
Sequential Cell Count: 35069

Multiple experiments for improving at-speed coverage were performed on the design. The run configurations and their results are as follows:

Run 1: Normal design
Configurations: Normal at-speed LOS coverage was found for the design without making any changes in the design. The ATPG results are:

Fault Coverage	Test Coverage	No. of faults	No. of Test patterns	No. of unobservable faults
78.69	85.37	1249878	2597	8899

Table 7.1 ATPG results for normal design run

Run 2: Shadow flop technique
Configurations: In the second run, shadow flop technique was implemented. For each scan flop, a dedicated shadow flop was inserted before it in the scan path so as to improve the controllability on the scan flops. The number of flops effectively doubles and so this is not a practical approach. In total 27915 cells were added during the shadow flop technique. A new netlist was generated after applying the shadowing technique.
Initially, the coverage improvement was found to be > 2%. But those results were not accurate.

This is because a large portion of sequential logic is added by the shadow flops which got tested and improved the coverage number. The extra logic should not be considered for testing. So the new scan flops added were excluded from testing in ATPG or rather the faults of the new scan faults were excluded from fault list and results were generated.

Fault Coverage	Test Coverage	No. of faults	No. of Test patterns	No. of unobservable faults
78.93	85.61	1249846	2478	7176

Table 7.2 ATPG results for shadow flop technique

The improvement in coverage as compared to the normal run is not great. It can be seen that the number of faults are same in both normal and shadow flop run. This guarantees that the extra flops added are not considered for testing. A great amount of pattern reduction was observed i.e. around 5% by this technique. Also the number of unobservable faults reduced considerably. This was the motive of applying the shadow technique.

Run 3: Partial shadow flop technique

Configurations: In another attempt, rather than shadowing all the scan flops, only one type of scan flops were shadowed. The purpose was to reduce the extra flops added and check the coverage. 13343 extra scan cells were added which is less than half of the previous run. The results are as below:

Fault Coverage	Test Coverage	No. of faults	No. of Test patterns	No. of unobservable faults
78.91	85.59	1249846	2584	7480

Table 7.3 ATPG results for partial shadow flop technique

Is can be seen that the coverage is

There are no sources in the current document. almost the same for all shadowing and partial shadowing technique. But the number of patterns increased relative to the previous technique. So we can achieve the same coverage even with half the scan cells added.

Run 4: Shadowing based on correlation data

Configurations: In chapter 4, the correlation between the scan flops is explained and also how it is generated. Larger the correlation number, higher the chance of coverage loss for that scan flops. So for this run, shadow flops were added based on that numbers. The scan correlation data array was sorted in descending order of the correlation value. The highest 100 numbers were

taken and their corresponding scan flops were noted. Shadowing was performed only on that scan cells. The purpose was to target only the scan flops responsible for more coverage loss. It was found that only 18 flops were present in the worst 100 correlation numbers. So shadowing these 18flops can improve the testability for 100 worst coverage points. The coverage numbers obtained with a new netlist containing the extra 18shadow flops are follows:

Fault Coverage	Test Coverage	No. of faults	No. of Test patterns	No. of unobservable faults
78.71	85.39	1249846	2475	8857

Table 7.4 ATPG results for shadowing based on correlation data

There is not much improvement in results from the normal run. This is because a very small area is being targeted for coverage improvement. A larger set of flops can be targeted but would increase the area and power.

Chapter 8

Conclusion and Future work

The shadow flop technique gives the maximum at-speed coverage improvement possible. Though there is not much improvement in coverage numbers, the trend of coverage improvement is present. But it is unrealistic because it effectively double the sequential cells count in the design. A good amount of pattern reduction (around 5%) was observed by this technique. Also the number of unobserved faults decreased considerably (around 20%).
By using partial shadowing technique, the coverage numbers were almost equal to the full shadow flop technique, but the test patterns required to get that coverage were greater than the latter.

Future work

The ATPG results can be improved even more by configuring the ATPG in a better way. Currently for this project due to technical limitations, the ATPG was only configured for scan enable signal and basic LOS mode.
The coverage for scan chain reordering can be performed. Also a better TSP algorithm can be used for scan chain reordering.
It was found that the correlation for some scan flops was very high and that they were in the same scan chain. So that chain can be tested by the LOC mode and the rest of the chains can be tested in LOS mode. By testing the high correlation flops in LOC mode, there is no dependency for launching the fault on the previous flop as the fault is launched by the functional path and not the scan path. But this will be design specific solution as there is no guarantee that the high correlation flops will be in the same chain.

References

[1] N. Ahmed, C.P. Ravikumar, M. Tehranipoor and J. Plusquellic, “At-Speed Transition Fault Testing With Low Speed Scan Enable” in Proc. IEEE VLSI Test Symp. (VTS’05), pp. 42-47, 2005.

[2] J. Saxena et al., “Scan-Based Transition Fault Testing Implementation and Low Cost Test Challenges” Proc. Int’l Test Conf. (ITC 02), IEEE Press, 2002, pp. 1120-1129.

[3] DFT tool manual

[4] Synthesis tool manual

[5] IEEE Std 1149.1 (JTAG) Testability Primer

[6] www.anysilicon.com

[7] www.asic-world.com

[8] www.wikipedia.com