Diagnostic Test Generation for Path Delay Faults in a Scan Circuit

by

Zeshi Luo

A thesis submitted to the Graduate Faculty of
Auburn University
in partial fulfillment of the
requirements for the Degree of
Master of Science

Auburn, Alabama
August 1st, 2015

Keywords: ATPG, path delay fault, scan circuit,exclusive test, fault diagnosis

Copyright 2015 by Zeshi Luo

Approved by

Vishwani Agrawal, James J. Danaher Professor of Electrical and Computer Engineering
Adit Singh, James B. Davis Professor of Electrical and Computer Engineering
Victor Nelson, Professor of Electrical and Computer Engineering

Abstract

With the increase of density, speed and test time of large VLSI circuits, manufacturers are eager to find efficient ways to bring up yields. Often, VLSI testing only tells if a circuit is faulty but is unable to locate faults. Diagnosis helps find locations of faults so that problems with the design and manufacturing process can be analyzed.

By adding a few logic gates and only two flip-flops to the netlist model, we are able to generate distinguishing tests for path delay faults with the same tools as are used for detecting a stuck-at fault. As a result, the capability of automatic test pattern generation (ATPG) tool is improved for diagnosis of path delay faults. Our proposed diagnosis method improves the capability to pinpoint the cause of failure by narrowing down the list of suspected fault candidates.

The proposed ATPG procedure generates tests to distinguish between path delay fault pairs, i.e., two faults are required to have different output responses. Test pattern generated from the model is shown to distinguish between any pair of path delay faults. In order to evaluate the improvement in tests we use a previously proposed Diagnostic Coverage (DC) metric.

We apply our diagnosis method to several ISCAS'89 benchmark circuits. Experimental results show the improvement of DC. The proposed diagnosis system may also be used for other fault models if behavior of faults can be mapped onto a stuck-at fault. It may also enhance the ability of conventional ATPG tools to significantly improve DC without increasing their complexity.

Acknowledgments

All human achievements, trivial or momentous, are invariably indebted to contributions from associates and peers. My work is completed with help from a lot of people. I realized at every step that whatever I have achieved was not possible without the amazing people I have in my life as mentors, friends and family.

First, I express my sincere gratitude to my academic advisor Dr. Vishwani Agrawal. Being a fantastic mentor, he always shows me the right direction and gives me great suggestions. He guided me with encouragement and patience.

I would like to thank Dr. Adit Singh. His wonderful courses impressed me a lot and also helped me do my thesis work. He has always been kind and helpful. I also thank Dr. Victor Nelson for agreeing to be on my thesis committee. His course guided me on how to correctly use simulation tools and how to analyze experimental results.

Also, my thank is owed to my friends Bei Zhang, Xiaolu Shi, Baohu Li, Chao Han for years of company and help.

Last, but not the least, my undying gratitude goes to my caring parents.

I dedicate this work to all those who have encouraged and inspired me.

Table of Contents

Abstract ii

Acknowledgments iii

List of Figures viii

List of Tables xi

1 Introduction 1

1.1 Motivation 2

1.2 Problem Statement 4

1.3 Contribution 4

1.4 Thesis Organization 5

2 Very-Large-Scale Integration (VLSI) Testing Technology 6

2.1 Concepts of Testing 6

2.2 Types of Testing 7

2.3 Types of Fault Models 7

2.3.1 Stuck-at Fault 7

2.3.2 Transition Delay Fault 8

2.3.3 Path Delay Fault 8

2.3.4 Other Kinds of Fault Models 8

2.4 Background and Algorithms of Automatic Test Pattern Generator (ATPG) 9

2.4.1 Failures of ATPG to Generate Test 10

2.4.2 D-Algorithm 10

2.5 Fault Simulation 13

2.6 Fault Equivalence [22] 13

2.7 Fault Collapsing [23] 14

2.8 Scan Design for Test 14
2.8.1 Scan Design 14
2.8.2 Scan Design Rules 16
2.8.3 Process of Scan Test 16
2.8.4 Summary of Scan Design 17
2.9 Launch on Capture (LOC) and Launch on Shift (LOS) 18
2.9.1 Introduction to LOC 18
2.9.2 Introduction to LOS 18
2.9.3 LOC vs LOS 19
3 Background and Overview of Fault Diagnosis 20
3.1 Approach of Fault Diagnosis 20
3.2 Combinational Fault Diagnosis Methods 21
3.2.1 Fault Table 22
3.2.2 Fault Dictionary 22
3.2.3 Reduction in Size of Diagnostic Data 23
3.3 Generating Tests to Distinguish Faults 24
3.4 Redundant and ATPG Untestable Faults 25
3.4.1 Redundant Fault (RE) 25
3.4.2 ATPG Untestable Fault (AU) 25
4 Generating Test for Delay Faults Using Stuck-at Fault Tools 27
4.1 Background 27
4.2 Three Major Phases in Path Delay Fault Testing 27
4.2.1 Initialization Sequence 27
4.2.2 Path Activation Sequence 27
4.2.3 Propagation Sequence 28
4.3 Test Generation Model 28
4.3.1 Test Specified for a Single Stuck-at Fault 28

4.3.2 Test Model for Falling Transition at Destination Flip-flop . . . 29
5 Background and Overview of Exclusive Test . . . 32
5.1 Background . . . 32
5.1.1 XOR Gate . . . 32
5.1.2 Exclusive Test for a Pair of Faults . . . 32
5.2 Boolean Analysis of New Exclusive Test Algorithm . . . 34
5.3 Diagnostic Metric . . . 36
5.3.1 Fault Coverage (FC) . . . 36
5.3.2 Diagnostic Resolution (DR) . . . 36
5.3.3 Diagnostic Coverage (DC) . . . 37
5.3.4 Other Kinds of Diagnostic Metrics . . . 39
5.4 Multiple Output Circuits . . . 39
6 Diagnostic Test Generation for Path Delay Faults . . . 40
6.1 Setting the Goal . . . 40
6.2 Path Delay Fault . . . 41
6.2.1 Robust Path-delay Test . . . 42
6.2.2 Non-Robust Path-delay Test . . . 43
6.2.3 Functional Detection Test . . . 45
6.3 Modeling a Path Delay Fault . . . 46
6.4 ATPG Model of Path Delay Faults . . . 47
6.4.1 Scan Circuit Test . . . 49
6.5 Scan-Based At-Speed Test Generation . . . 50
6.6 Detection Test Phase . . . 50
6.6.1 Fullscan Circuit . . . 52
6.6.2 Construction of Diagnostic Dictionary . . . 52
6.6.3 The Need for Generating Exclusive Test . . . 53
6.7 Exclusive Test Phase . . . 54

6.7.1 Path Definition file 55
6.7.2 Construction of Exclusive Test Model 56
6.7.3 Analysis of ATPG Untestable (AU) Fault 59
6.7.4 Diagnostic Test Pattern Generation 61
7 Experiments Setup and Analysis of Results 62
7.1 Experimental Setup 62
7.1.1 Construction of AND Gate of Test Model 62
7.1.2 Construction of Test Model 63
7.1.3 Construction of Final ATPG Test Model 64
7.2 Results and Analysis 64
8 Conclusions, Developments and Future Work 70
8.1 Conclusion 70
8.2 Developments 71
8.3 Future Work 73
8.3.1 Application of Test Model to Non-Scan Circuit 73
8.3.2 ATPG Tools to Reduce ATPG untestable (AU) Faults 73
8.3.3 Diagnosis of Real Defects 73
8.3.4 Overlap Component of Two Paths Blocked by Other Signal . . 74
Bibliography 75

List of Figures

2.1 A Singualar Cover example. 11

2.2 Fault simulation flowchart. 14

2.3 A D flip-flop [13]. 15

2.4 A scan flip-flop [13]. 15

2.5 Scan design [13]. 16

2.6 Partial scan design. 17

2.7 Process of LOC. 18

2.8 Process of LOS. 19

3.1 Principle of failures back-tracing algorithm. 21

3.2 Process of cause-effect algorithm. 21

3.3 Example of different conditions of diagnosis. 24

3.4 Example of redundant fault in circuitry. 26

4.1 Test generation model for a falling transition [31]. 30

4.2 Operation of FSM state diagram [31]. 31

4.3 Test generation model for a rising transition at destination flip-flop [31]. 31

5.1 Exclusive test for two faults. 33

5.2 Exclusive test after simplification. 34

5.3 A CUT for exclusive test. 35

5.4 Relationship between DR and DE. 38

6.1 Fast and slow transitions on a path [13]. 43

6.2 Robust detection example [30]. 43

6.3 Non-robust detection test [13]. 44

6.4 Non-robust detection example [30]. 44

6.5 Functional detection example [30]. 45

6.6 A scan circuit example of path a-d-e. 46

6.7 Model of a slow-to-fall fault in path a-d-e. 47

6.8 Model of a slow-to-rise fault in path a-d-e. 48

6.9 An ATPG model: test that detect a s-a-0 fault distinguish a pair of slow-to-fall faults. 49

6.10 An ATPG model: test that detect a s-a-0 fault distinguish a pair of slow-to-rise faults. 50

6.11 ATPG test model: distinguish two slow-to-fall faults. 51

6.12 ATPG test model: distinguish two slow-to-rise faults. 52

6.13 Detection test flow in detection phase. 53

6.14 Flowchart of automatic exclusive test generation system. 54

6.15 An example of path definition file. 55

6.16 Position of AND gate in test model. 56

6.17 An example a path delay fault model. 56

6.18 An example an AND gate for Table 6.2. 58

6.19 An example an AND gate for Table 6.3. 58

6.20 An example of AU faults for stuck-at fault. 59

7.1 Test flows for a pair of path delay faults. 63

7.2 An example of a pair of slow-to-fall faults. 63

7.3 An example of AND gate. 65

7.4 An example of AND gate in test model. 66

7.5 An example of AND gate in test model. 67

7.6 An example of circuit after inserting test model to s27. 68

8.1 Calculation of path scores. 72

List of Tables

2.1 An example Singular Cover. 11

2.2 An example of D-cube. 12

3.1 An example of fault table. 22

3.2 An example of fault table. 23

5.1 Truth table of XOR gate. 33

5.2 Modified dictionary. 37

6.1 Diagnostic coverage after detection test phase 54

6.2 Signal requirement for signal e to be 1. 57

6.3 Signal requirement for signal e to be 0. 57

6.4 Signal requirement for path a-c-f-h-k. 60

6.5 Signal requirement for path b-e-g-h-k. 60

7.1 Signal requirement for k to be 1 in path g-k. 64

7.2 Signal requirement for k to be 0 in path g-k. 64

7.3 Comparison between DC of detection test and exclusive test. 67

7.4 Comparison between detection test and exclusive test. 69

Chapter 1

Introduction

The first semiconductor chips held few transistors each. Subsequent advances added more transistors to the chip which enable it to do more work. Current technology has moved quickly. There are millions or even billions of gates in today's microprocessor. However, many integrated circuits are not perfect. They may contain fabrication defects on manufacture.

Sometimes, initial die yield may be just 20% for high end circuits. Therefore, we need to carefully test the circuits and identify the locations and nature of faults in order to raise yield.

With the increases in die size, transistor density and process time of the chips, manufacturers need to find ways to improve the yields. After testing circuits, the defective circuits will be found. Fault diagnosis [1] is then applied on a malfunctioning circuit. It narrows down the range of the suspect area. This helps researchers locate the defects in a device and identify defects in manufacturing. The costs of time and equipment for the diagnostic work are so large that sometimes they dramatically increase the price of the product. However, fault diagnosis is a necessary step for the industry. Therefore, smart diagnosis algorithms need to be applied.

The purpose of fault diagnosis is to find the cause of defects in a manufactured chip. A good diagnosis system should efficiently help scientists quickly and accurately find the location of a defect. It is possible that certain single-location defects may behave as multiple faults. Defects of this kind will affect several fault locations or affect several branches.

Diagnosis plays an important role in improving yield. Physical defects in circuits are modeled with different fault models. In this thesis, we are dealing with stuck-at faults and path delay faults. Our work consists of detection test and exclusive test. The DC is about 20% after detection test. And then We will generate an exclusive test for distinguishing a pair of path delay faults. A test is found if the output response of two faults is different. We use Diagnostic Coverage (DC) to measure the quality of our work. Experimental results show the improvement of DC after implementing our exclusive test.

1.1 Motivation

In 1947, John Bardeen [2] and Walter Brattain [3] at Bell labs performed a series of experiments and found that output power could be greater than the input power after two gold points contacts were used on a crystal of germanium. In the following months, they got more knowledge of semiconductors. In 1956, Shockley, Bardeen and Brattain were awarded the Nobel Prize in Physics for the discovery of the transistor effect.

The transistor is one of the most famous inventions of the 20th century. Researchers use transistors to design different logic devices. People have created circuits by combining millions of transistors into a single chip. This made the device extremely small. As the function, transistor density and complexity increased, Gordon Moore made analysis of the number of components for each circuit in the previous years and he made a prediction that has been regarded as the Moore's law [4]. Over the history of computing hardware, the number of transistors in a dense integrated circuit has doubled approximately every two years. That observation is accurate. While scientists are trying to combine more components into a circuit, some circuits are unable to function as well as they are expected to. The reason is that the transistors and the wires that connect them are not absolutely safe and stable. Sometimes, they may be

unexpectedly open or short. Testing the circuits verifies the correctness of a circuit design. Faults in a circuit may be of any type and occur at any place. Several fault models are categorized as: stuck-at fault, transition delay fault, stuck-open fault, path delay fault and so on. Fault models are built in order to analyze them in different ways. Test patterns are generated for detecting them. Moreover, the number of test patterns is minimized by using one pattern to detect several equivalent faults.

However, even if scientists know exactly which kind of fault causes a malfunction in the circuit, they are still unable to correct the fault. If some similar problems occur many times, scientists began to look for the cause of the problem. They analyze the failed devices in order to identify the location of the fault. That work could be applied to the manufacturing process to improve the final results.

The standard that evaluates fault diagnosis is the accuracy with which faults can be located. This can be called a diagnostic metric. Fault equivalence is an indispensable concept in digital circuit world, especially in testability, test generation and logic analysis. Functionally equivalent faults are undistinguished. If none of the input test vectors can distinguish two faults at primary outputs, those two faults are functionally equivalent.

Fault equivalence methods are broadly categorized as structural and functional. Structural equivalence methods determine the extent to which two nodes are physically connected to each other. These methods are fast but not very complete. Functional fault equivalence methods will identify more classes, but they are more expensive and there are difficult to develop algorithms to solve this problem under various conditions. This is because functional fault equivalence identification in combinational circuits is co-NP complete [5].

1.2 Problem Statement

In this thesis, we propose a diagnostic method to improve diagnosis of path delay faults of a scan circuit. Our tests consist of detection test and exclusive test. After the detection test of path delay faults, the diagnostic coverage (DC) is generally found to be less than 20%, which may be considered unsatisfactory. In order to improve DC, we insert test modeling logic into the original circuit netlist. The modeling logic does not affect the hardware as it is not implemented. It enables us to generate a test for a stuck-at fault at the control signal of a multiplexer in the inserted logic. This test is an exclusive test that determines whether a targeted pair of path delay faults can be distinguished.

1.3 Contribution

In the field of fault diagnosis, it is crucial to enhance the fault resolution. The *resolution* normally refers to a group of faults that cannot be distinguished from each other. If more faults can be distinguished, there will be many more small groups, leading to higher resolution. As a result, the diagnostic coverage (DC) can increase. *Diagnostic coverage* (DC) is the ratio of the number of undistinguished fault groups to the total number of faults.

A test pattern generation tool for a single stuck-at fault is used in order to distinguish path delay faults. The traditional detection test is accomplished by generating test patterns for path delay faults and using fault simulation to detect which faults contain the same signatures. The DC in this traditional method is low. Moreover, if two paths end in the same output with same transition, these two path delay faults will be not distinguished. In the benchmark circuits, there are so many similar cases.

By adding a few gates and several flip-flops, we can generate a test for a stuck-at fault in order to distinguish a pair of path delay faults. Whether a test for the stuck-at

fault is found indicates whether or not the two path delay faults can be distinguished. If a test is found, the DC of path delay faults along the scan circuits will improve, because new groups will be constituted after generating test on the test model. The test model collects the path information of the target path. Actually this model we insert into the original circuit is based on the fault type and signal requirements of paths related to the target path.

Fault simulation shows that the test pattern generated for a single stuck-at fault in the distinguishing model can only detect one of the two path delay faults. Fault simulation also convinces us that the pattern is able to activate only one path delay fault if these two faults are distinguished. If there is no test pattern found then the pair of two faults are analyzed for the cause, as shown in the latter part of the thesis.

1.4 Thesis Organization

This thesis is broadly divided into seven chapters. The organization of chapters is as follows:

Chapter 2 introduces the background and gives an overview of VLSI testing to the reader.

Chapter 3 explains applications and algorithms for fault diagnosis.

Chapter 4 discusses how to generate delay test by using stuck-at fault testing tools.

Chapter 5 shows the theoretical proof of construction of our exclusive test on a pair of path delay faults.

Chapter 6 explains the method of constructing our test model.

Chapter 7 gives experimental results and their analysis.

Chapter 8 concludes the thesis work, outlining recent developments and suggestions for future research.

Chapter 2

Very-Large-Scale Integration (VLSI) Testing Technology

While more and more transistors are combined into a single circuit, more faults may be found in the circuit. VLSI testing is now playing an important role in the digital world.

Circuits will be tested to detect if they function well. Many integrated circuits are not perfect. Fabrications defects are created in the manufacture process. Therefore, testing chips before selling them to costumers is really necessary.

2.1 Concepts of Testing

Students are sometimes required to take exams after they learn from the teacher's lecture and textbook. In the exam, students will answer some questions. Then the exam paper is graded by the teacher. If the answer is wrong, teacher will mark it as a wrong answer. Whether the answer is right or wrong is based on the text book or the notes.

Testing is similar to answering questions from a tester. The result of circuit is like the student's answer while questions are similar to test patterns. Every circuit will be assigned test patterns to see what result it produces. If the result at the output is different from the right answer, it will be marked as wrong. Moreover, it is necessary to record what the expected response of good circuit should be before performing test on circuits.

2.2 Types of Testing

Testing types can be broadly categorized into three parts: verification testing [6], manufacturing testing [7] and acceptance testing [8]. Verification testing is ferociously expensive. It comprises electron beam testing, repeated functional tests and so on. Manufacturing test is to determine whether the manufactured chip meets the standard. It tests each device on the chip. Acceptance testing is to guarantee the quality of purchased parts. In this thesis, our focus is an manufacturing test.

2.3 Types of Fault Models

Fault models [9] are necessary for a test methodology. Approximation of defects in the circuit could be analyzed by the fault models. In the field of VLSI testing,a fault model identifies targets for testing and makes analysis possible. We will introduce several fault models in this section.

2.3.1 Stuck-at Fault

Stuck-at fault is one of the most common faults in VLSI testing field. Individual signals and pins are assumed to be stuck at logical 1 or 0 [12]. This defect causes the line to be permanently stuck at one value. A wire that connects to a transistor can cause this fault when it is broken. In order to target a stuck-at 0 fault, test patterns could be applied in order to get a value 1 at the target point. If the result is not 1 but 0, the fault can be detected since the value is different from the expected response of a good circuit. To test a stuck-at 0 fault, test pattern will be set to make 1 at the target point.

2.3.2 Transition Delay Fault

It is assumed that in the fault-free circuit all gates have some nominal delays and that the delay of a single gate has changed [13]. Transition delay fault is either a slow-to-rise or slow-to-fall fault. Fault list contains 2N faults for a circuit with N nodes. Unlike a stuck-at fault, a test transition fault requires two vectors that will cause a rise or fall transition at the node.

2.3.3 Path Delay Fault

A delay defect in a circuit is assumed to cause the cumulative delay of a combinational path to exceed some specified duration [13]. Paths can start from the primary input or flip-flop's input to primary output or flip flop's output. The specified duration is usually the duration of clock period. And sometimes it may also be the vector period. A path may contain several gates and wires. The propagation delay of interconnect or the switching delay of a device is able to cause path delay. Two vectors are needed to detect path delay faults. Moreover, detecting a path delay fault is more complicated than for a transition delay fault. Not only the on-path signal needs to be considered but the off path signal also requires careful consideration in order to activate a path delay fault. Three types of path delay faults need to be considered. We will discuss them in the latter part of this thesis.

2.3.4 Other Kinds of Fault Models

A stuck-open [10] fault sets an unexpected high-impedance state at the output of a gate. A test sequence is applied to the output, VDD and GND independently. A stuck-open fault also requires two vectors. A bridging fault [11] consists of two connected signals that should not be. This kind of fault shorts the circuit between lines or cells. Since a bridging fault is a bidirectional fault, it means that line a affects

line b, while b also affects a. There are various types of bridging faults: wire-Or, wire-And and so on. The two signal lines are connected. For wire-or fault, the line which has a value of 1 will determine the final result, no matter what the state of the other line is. It acts like an OR gate that gets a 1 at the output if the value of 1 appears at the input. Similarly, for wire-AND, if one line is set to be 0, the other line will be forced to be 0. Other kinds of faults are IDDQ [14] faults, stuck-off faults and so on.

2.4 Background and Algorithms of Automatic Test Pattern Generator (ATPG)

ATPG is an electronic technology applied to find particular test patterns for a digital circuit. These test patterns are able to find a difference between the behavior of a good circuit and a faulty circuit. ATPG helps people save time to find test patterns. In timing test, the process of the ATPG targeting a particular fault consists of two phases: fault activation and fault propagation. In the fault activation phase, it sets a value at the fault site which is opposite of the value from fault model. In the fault propagation phase, the test pattern sensitizes the target path to make sure that the resulting value of signal move towards the end of path. If the target fault is a stuck-at 0 fault, ATPG will just generate a test pattern to establish a 1 value that is the opposite value of the 0 at the target point. Actually, the second test pattern of the timing test is similar to targeting a stuck-at fault. Path delay fault consists of slow-to-rise and slow-to-fall fault.

The total amount of detected defects and number of test patterns are used to measure the effectiveness of ATPG. The former one is related to the test quality while the latter one indicates test application time.

2.4.1 Failures of ATPG to Generate Test

Unfortunately, ATPG sometimes may fail to generate test patterns for a particular fault. First of all, the fault may be a redundant fault, which is undetectable by the ATPG. Because the circuit is designed in a way that output will never change. Secondly, it is also due to the algorithm of the ATPG itself, since ATPG is a NP-complete problem [15]. There will be cases where patterns exist, but ATPG gives up since it will take an incredibly long time to find them. NP-complete problems are in NP, the set of all decision problems whose solutions can be verified in polynomial time [16]; NP may be equivalently defined as the set of decision problems that can be solved in polynomial time on a non-deterministic Turing machine [17]. Even if there exists a test pattern that could detect a fault, the ATPG might give up because it costs quite a long time to find that pattern.

2.4.2 D-Algorithm

Nowadays, there are a lot of algorithms for ATPG. The D algorithm is a very famous ATPG algorithm. We will show how it works in this section.

Overview of Singular Cover

Singular cover shows the necessary prime implicants of the Karnaugh map [18] with the minimal set of assignments of logic signal. The gates in the circuit are shown in a table with their inputs and outputs. There are three conditions for each input: 'X','0','1'. As a result, each gate has three lines to show different conditions. An example is shown in Figure 2.1.

We can easily find the corresponding three cases for each gate, it is shown in Table 2.1.

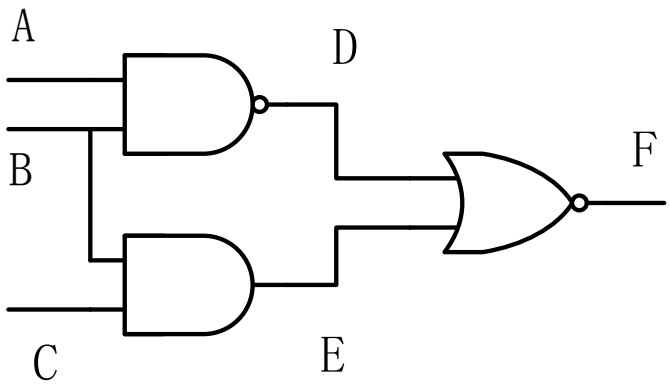

Figure 2.1: A Singualar Cover example.

Table 2.1: An example Singular Cover.

Gate	Input0	Input1	Output	Gate	Input0	Input1	Output
AND	B	C	E	NOR	D	E	F
1	0	X	0	1	1	X	0
2	X	0	0	2	X	1	0
3	1	1	1	3	0	0	1

D-Cube

D-cube is a collapsed truth table entry that can be used to characterize an arbitrary logic block [13]. We use Roth's 5-value algebra [19]. It can either change all of $\overline{D}$'s to D's or D's to $\overline{D}$'s . It is shown in Table 2.2

D-intersection

The definition of D-intersection is defined as the set of circumstances under which different cube labels for different logic gates can coexist in the circuit [13]. In other words, a specific signal value has already been assigned to one cube, the other cubes must assign same signal value or unknown value. The equation set for this example is Equation 2.1

Table 2.2: An example of D-cube.

input0	input1	output
B	C	E
D	1	D
1	D	D
D	D	D
$\overline{D}$	$\overline{D}$	$\overline{D}$
1	$\overline{D}$	$\overline{D}$
$\overline{D}$	1	$\overline{D}$

$$\begin{aligned} 0 \bigcap 0 = X \bigcap 0 = 0 \bigcap X = 0; \\ 1 \bigcap 1 = X \bigcap 1 = 1 \bigcap X = 1; \end{aligned} \tag{2.1}$$

D-contains

If the set of A cube vertices is a superset of the B cube vertices, cube A D-contains cube B.

Primitive D-cubes of failure

There are four items that could be modeled by Primitive D-cubes of failure. They are as following:

- stuck-at 1 fault.
- stuck-at 0 fault.
- bridging fault.
- arbitrary change in logic gate function.

For instance, the primitive D-cube of failure of a NOR gate stuck at 0 is "0 0 D". Because in the good circuit, both of the input must be set to 0 in order to make

a 1 at the output of the gate. However, the fault circuit will cause the output to be 0. Primitive D-cubes of failure are different from the propagation D-cubes. Because primitive D-cubes model a failure at the gate. However, the propagation D-cubes model a situation which propagates fault effects through gate.

Implication Procedure

Implication procedure could be categorized as three steps

- step1: Application of Primitive D-cubes of failure to model the fault.
- step2: propagation of fault effect to the output with appropriate propagation D-cubes(also called D-drive procedure).
- step3: justification internal circuit signals with singular cover cubes.

2.5 Fault Simulation

The purpose of fault simulation is to guide the test pattern generation process, measure effectiveness of test patterns and generate fault dictionaries. Fault simulation needs three components: fault list, test set and design model. Given these components, fault simulation will determine fault coverage [20] and set of undetected faults [21]. In the VLSI testing world, there are a lot of fault simulation algorithms, such as serial, parallel, deductive and concurrent fault simulation. Figure 2.2 shows the flowchart of fault simulation.

2.6 Fault Equivalence [22]

If all of the tests that detect fault1 can also detect fault2, these two faults are equivalent. In other words, the corresponding functions of the two faults are same. This concept can also help us distinguish a pair of faults. If a test is found that could

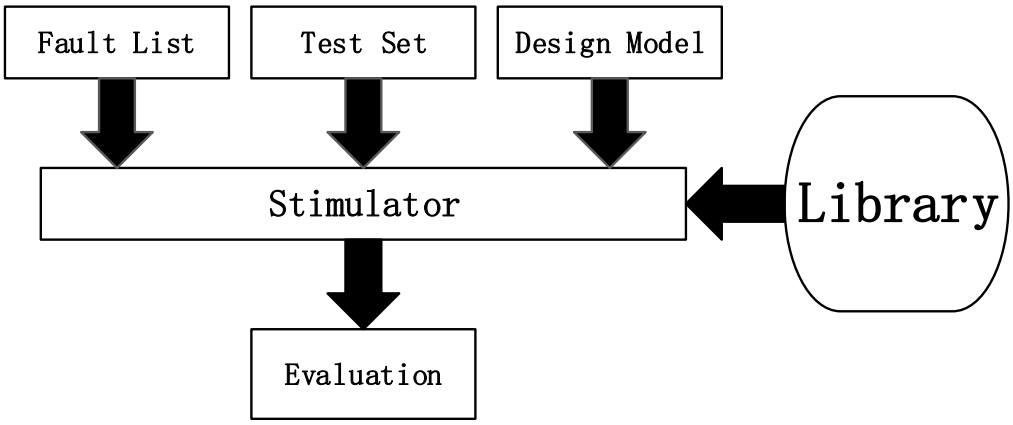

Figure 2.2: Fault simulation flowchart.

be able to detect one of the two fault but not the other, these two faults will not be equivalent.

2.7 Fault Collapsing [23]

If two faults are equivalent, any fault from a set of equivalent faults can actually represent the whole set. In this case, most of equivalent faults can be removed. The process of removing equivalent faults from the entire set of faults is called fault collapsing.

2.8 Scan Design for Test

The application of scan design to hardware test was published in the 1973 paper by Williams and Angell of Stanford University [24]. Many companies like IBM, NEC and others have broadly implemented the concept since then.

2.8.1 Scan Design

Scan design is the most popular structured Design for Testability(DFT) approach. Adding a test mode to the circuit enables all flip-flops to form one or more shift registers. Also, all flip-flops can be set to any state by just shifting logic states

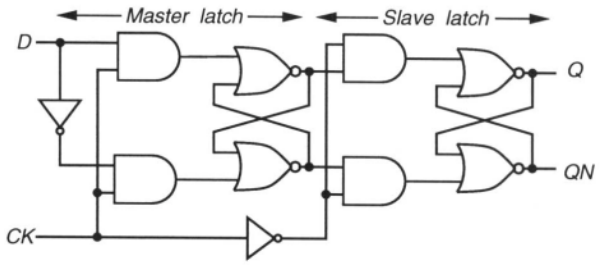

Figure 2.3: A D flip-flop [13].

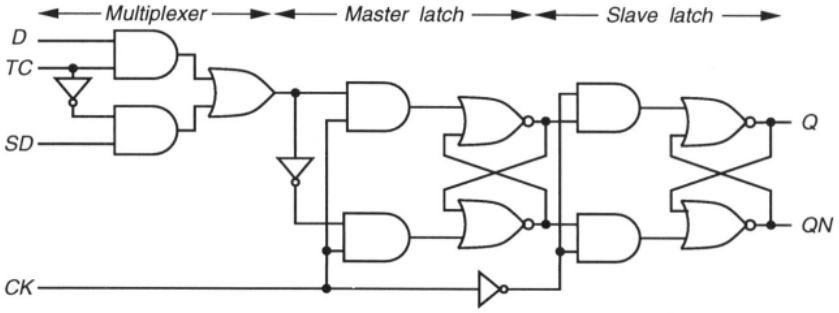

Figure 2.4: A scan flip-flop [13].

through scan chain. Observing the states of the flip-flops is quite convenient. This can be done by shifting the states of shift register. The time for observing could be the total amount time of the flip-flops of the longest scan register. A D flip-flop is shown in Figure 2.3 .

After adding a multiplexer and two new signals, scan data and test control, the D flip-flop becomes the scan flip-flop shown in Figure 2.4. Test control signal is similar to a switch that either propagates data or scan data into the D flip-flop.

In Figure 2.5, all of the D flip-flops have been replaced by the scan flip-flops. All of the test control (TC) signals are connected to a single TC signal. As a result, this TC signal controls all of the scan flip-flops. A scan chain is then built by firstly setting one of scan flip flop's scan input as a new primary input SCANIN. Then the scan chain connects the output of each scan flip-flop(SFF) to the Scan input of the next SFF. At the end of this chain, the output of the SFF is defined as SCANOUT.

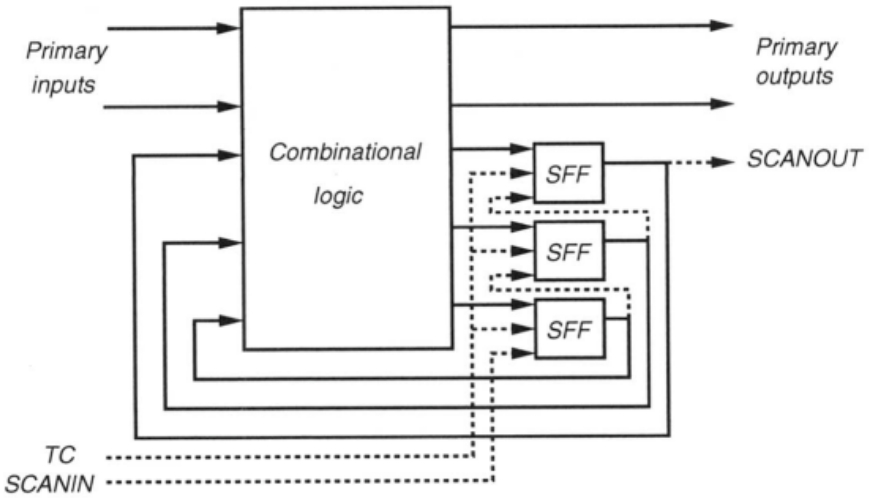

Figure 2.5: Scan design [13].

The scan circuit starts from the output of a SFF and end at the data input of SFF. This circuit go through combinational logic.

2.8.2 Scan Design Rules

- Rule1: The D-type master-slave flip-flop is the only one that could be used.
- Rule2: There should be at least one primary pin available.
- Rule3: The primary inputs are required to control all of flip-flop clocks.
- Rule4: Data inputs of flip-flops should not be fed by clocks.

2.8.3 Process of Scan Test

Firstly, the scan enable signal will be activated so that a series of test patterns are shifted through the scan chain. Secondly, the Scan Enable (SE) signal is disabled before the test patterns for primary inputs have been applied to the circuit. Finally, functional clock signals are pulsed to test the circuit and capture the combinational circuit outputs. And we shift results out to verify correct capture values. Different test patterns could be shifted through SFFs when the SE is enabled.

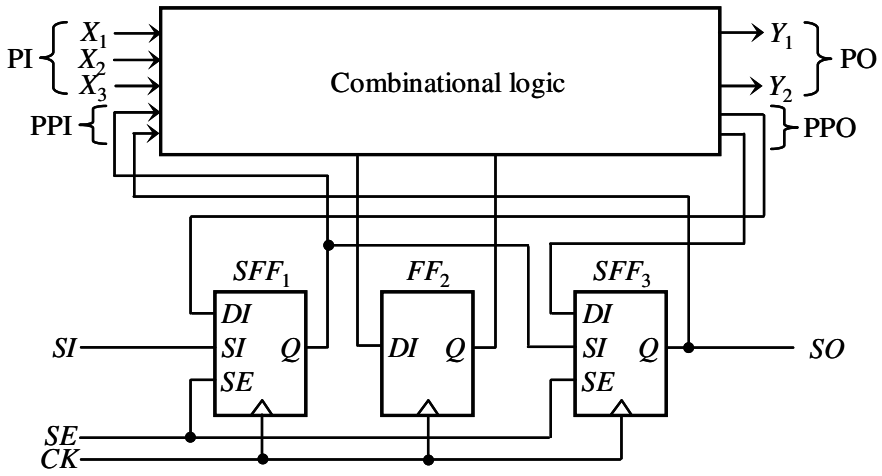

Figure 2.6: Partial scan design.

2.8.4 Summary of Scan Design

Scan design is regarded as a milestone in the industry. Before generating tests, design automation tools can insert scan logic into a circuit with D flip-flops. This method is quite simple and efficient. Nowadays, more scan methods have been developed. Figure 2.6 shows a partial scan design. Partial scan method only transforms a subset of D flip-flops in the circuit into scan flip-flops. Multiple scan chains reduce time to load and unload by inserting multiple scan chains in parallel instead of using long scan chain. In fault diagnosis, scan design helps get high fault coverage. Both of the scan overheads of area and performance are only about 5%. However, there are also some disadvantages of scan design. When this methodology is applied to a large circuit, it may take an incredibly long test time to test the circuit. Moreover, test data is also quite large. In a word, it's not a fast test.

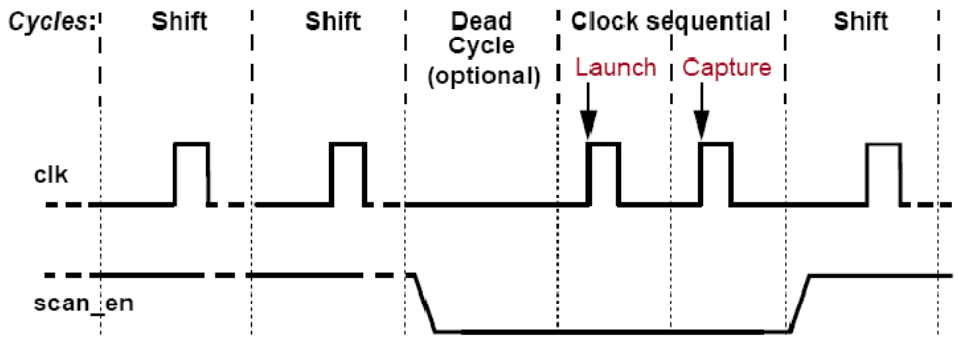

Figure 2.7: Process of LOC.

2.9 Launch on Capture (LOC) and Launch on Shift (LOS)

2.9.1 Introduction to LOC

Two vectors V1 and V2 are used to perform delay fault testing. Figure 2.7 shows the LOC waveform. There are five steps to implement LOC. (1) The circuit is initialized to be 1 which sets the circuit to scan mode. The first test vector is shifted into the scan chains with a slow scan clock. Values are also set on primary inputs. (2) The second vector is obtained by the response of first vector. (3) SCANEN is set to be 0 to set the circuit to functional mode. Second,the primary input vector is applied, and the circuit is clocked to launch the second vector. (4) The functional clock is applied to the circuit, with responses captured in scan flip-flops. (5) Set SE=1 and scan out the captured results as well as scan in the next vector.

2.9.2 Introduction to LOS

The difference of LOS is that the second vector is obtained by shifting one bit from first vector. Also, at step3, we hold SCANEN=1 for one cycle in order to clock the circuit in scan mode for one clock period while new primary inputs are applied. Figure 2.8 shows the process of LOS.

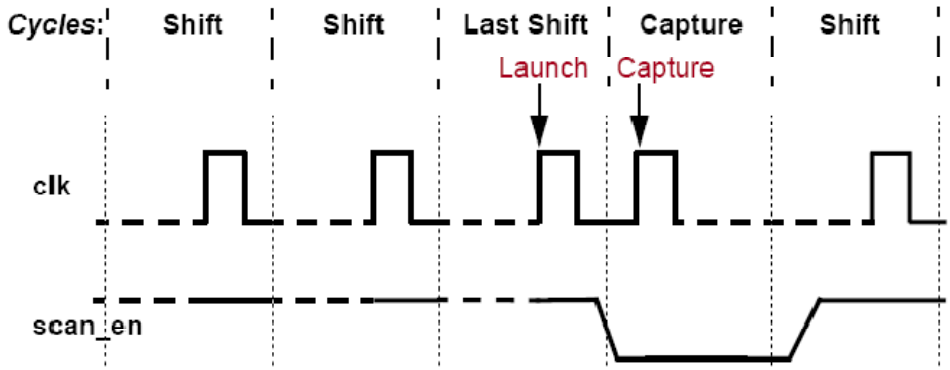

Figure 2.8: Process of LOS.

2.9.3 LOC vs LOS

LOC is different from LOS. In industry, LOC is more widely used that LOS, because sometimes using LOC cost less than LOC. The advantages and disadvantages of them are as following:

1. In LOS, the last shift happens at the fast clock speed. The entire design will become active that makes average power in the launch cycle very high.

2. In LOS, fast test generation methodologies for combinational circuits can be applied without many modifications. Scanned flip-flops are considered primary inputs in the ATPG for combinational circuits, so the new constraints on these "primary inputs" must be added to the existing ATPG.

3. In LOS, some redundant faults would be detected. SCANEN signals must operate at full speed. A large number of the sensitizable paths under the launch-on-shift constraints are sequential false paths that are not sensitizable in functional mode.

4. In LOS, switching the SCANEN signal during a short time period also costs a lot of time. Since SCANEN signal is broadly placed in the circuit.

5. In LOC, SCANEN signal is not required to operate at full speed. Sensitizable paths under the launch-on-capture constraints are also sensitized in functional mode.

Chapter 3

Background and Overview of Fault Diagnosis

With the development of VLSI testing, failed chips can be detected more easily than before. Moreover, scientists also try to find ways to identify the locations of these faults in order to increase the yield. If the behavior of a unit under test (UUT) [25] is different from the expected behavior, this UUT fails. Diagnosis helps scientists locate the physical fault in the model of a UUT.

3.1 Approach of Fault Diagnosis

Over the years, a lot of diagnosis algorithms are applied into the industry work. Two main types of diagnosis algorithms are circuit partitioning (Effect-cause diagnosis) [26] and model-based diagnosis (cause-effect diagnosis) [27]. The effect-cause diagnosis identifies possibly-faulty portions of a circuit, especially logic block interconnects. It's based on observed behaviors and expected (good-circuit) functions. Figure 3.1 shows the principle of back tracing failures, a method of effect-cause algorithm. It separates known-good portions of circuit from likely areas of failure. It's similar to picking up suspects from passengers in the airport. Intersection of multiple cones is highly suspect. This algorithm is simple and popular, but it sometimes fails in giving indication of a defect mechanism. Another algorithm compares behaviors to fault simulations with assumed fault models. Fault signatures [28] generated by a simulator can be used to predict the presence of different faults. It predicts what may happen when the circuit is not good. Figure 3.2 shows the process of it. However, wrong directions could be given by some unmodeled defects.

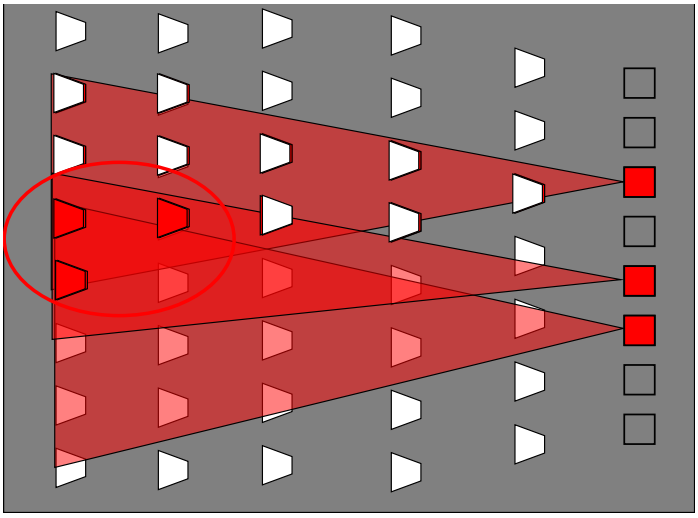

Figure 3.1: Principle of failures back-tracing algorithm.

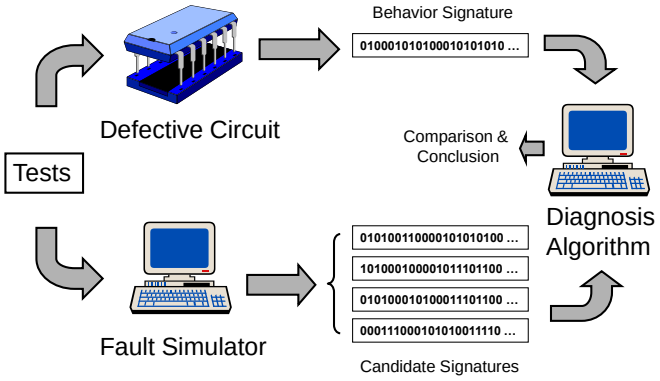

Figure 3.2: Process of cause-effect algorithm.

3.2 Combinational Fault Diagnosis Methods

Most of the work of combinational fault diagnosis will be done before testing. Fault simulation will be used to determine a response to a given test. A database will be constructed in this process to keep a record of responses. This database can be defined as a fault dictionary. If faults need to be located, one tries to match the actual results of a test with one of the previously computed expected results stored in the database. The results are the response that represents the response of faults to each test pattern.

Table 3.1: An example of fault table.

	F_1	F_2	F_3	F_4	F_5	F_6	F_7	E_1	E_2	E_3
T1	0	1	1	0	0	0	0	0	0	1
T2	1	0	0	1	0	0	0	0	1	0
T3	1	1	0	1	0	1	0	0	1	0
T4	0	1	0	0	1	0	0	1	0	1
T5	0	0	1	0	1	1	0	1	0	1
T6	0	0	1	0	0	1	1	0	0	0

3.2.1 Fault Table

A fault table is a matrix of test patterns and faults as shown in Table 3.1. The column represents faults while rows indicate whether each test pattern can detect the fault. If the test pattern can detect the fault, it will be 1 in the table. Otherwise, it will be 0. The test results of E matches a subset of column vectors $\{F_i, F_j, F_k\}$ in the fault table. This result corresponds to where a group of indistinguishable faults $\{F_i, F_j, F_k\}$ has been located. In the example the results of three test experiments E_1, E_2, E_3 are demonstrated. E1 corresponds to a case where a single fault is located, since E_1 only matches the F_5. E_2 matches both F_1 and F_4. E_2 corresponds to the a case where a subset of two indistinguishable faults is located. E_3 shows no match in the fault table indicating no faults can be located.

3.2.2 Fault Dictionary

A fault dictionary keeps the fault signatures as fault tables in order to be able to quickly detect the relationship between actual responses and expect results when there appears a fault. A fault table is actually a matrix where columns represent faults and rows represent tests. The test result is 1 when the actual result is not the same as the expected response, and it will be 0 otherwise. A fault dictionary consists

Table 3.2: An example of fault table.

Faults	Test1 Signature	Test2 Signature	Test3 Signature	index
F1	1	1	1	1
F2	0	1	1	2
F3	0	1	1	2
F4	1	0	0	3

of the same data as a fault table, with the difference that faults and expected results of test experiments are reorganized and represented in a more compressed form.

3.2.3 Reduction in Size of Diagnostic Data

A full response dictionary stores responses to each test vector. Millions or even billions of fault signatures are required to be included in the dictionary. As a result, the dictionary may be extremely large. Fortunately, compression techniques solve this kind of problem. Detected faults in fault simulation are removed from sets of simulated faults. Since faults detected by the same test patterns will produce same signature, these faults can be assigned to the same group. These faults are called equivalent faults.

In order to further reduce the size of a dictionary, another approach is a pass-fail dictionary [29]. As the name suggests, a pass-dictionary only keeps the data of pass or fail status of a fault for all applied vectors.

Table 3.2 shows a pass-fail dictionary. A 1 in the table indicates that fault failed this test while 0 is an indication of passing the test. F1 fails in all test vectors, thus given a signature of 111. The index will be 1 for it. Different index indicates different conditions of how the fault corresponds to the three tests. The index increases by 1 if the signature is unique. The same index will be assigned to same faults.

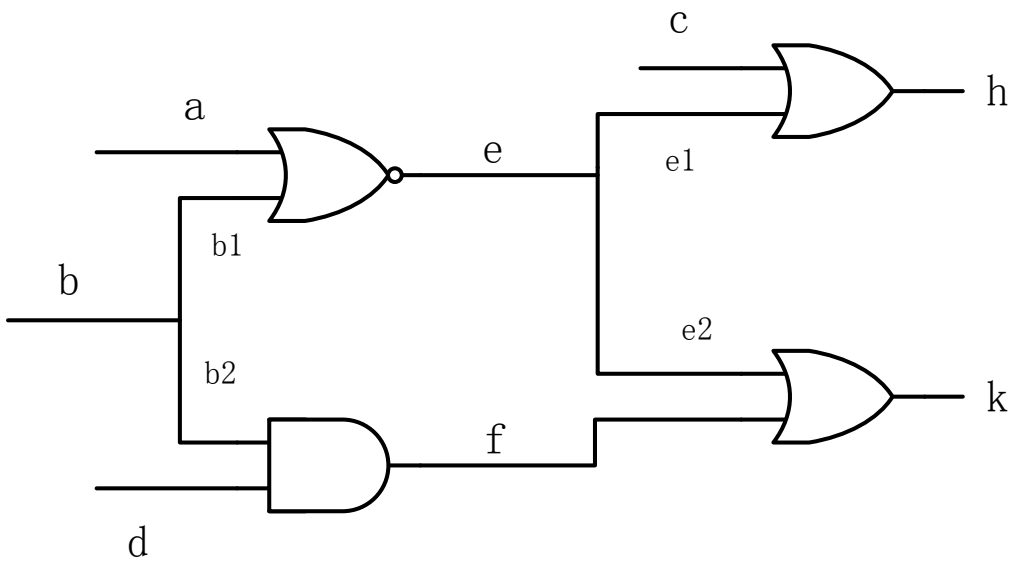

Figure 3.3: Example of different conditions of diagnosis.

3.3 Generating Tests to Distinguish Faults

Distinguishing equivalent faults with test pattern T can improve the fault resolution. If a pair of faults needs to be distinguished, there should be a test that can detect only one of these faults.

- Case1: F1 and F2 haven't influence on same set of outputs. A test should be generated for F1 will be using only circuit feeding the output of, or for F2 using only the circuit feeding the outputs of F2.

- Case2: F1 and F2 have influence on same groups of outputs. A test should generate F1 without activating F2. This idea will be used in this thesis.

Figure 3.3 shows an example of different conditions of diagnosis. Several cases are as follows:

1. There are two faults in the circuit. F1: b1 is stuck-at 0. F2: d is stuck-at 1. F1 can influence both outputs h and k. But F2 can only influence output k. Test pattern is 0010, can activate F1, and it will influence both outputs. As for F2, only

an output k can be detected. If both of h and k are wrong, then it is due to the presence of F1. F2 will be present if only k is wrong.

2. There are two faults in the circuit. F1: b2 is stuck-at 0. F2: e2 is stuck-at 1. These two faults will both influence the same output. But there exist a test pattern 0100 which only activates F2.

3. There are two faults in circuit. F1: b2 is stuck-at 0. F2: e2 is stuck-at 1. Test patterns, 0110, activates F1, and F2 is not activated, since d=0 blocks the AND gate.

4. There are two faults in circuit. F1: b1 is stuck-at 1. F2: b2 is stuck-at 1. Test pattern 1001 activates both of these faults to propagate to the same OR gate. However, the faults produce different values at the inputs of the gate, hence they are distinguished. If the output k is 0, it will be F1. Otherwise, if the output k is 1, there will two possible cases. One is F2, another is that neither F1 and F2 are present.

3.4 Redundant and ATPG Untestable Faults

3.4.1 Redundant Fault (RE)

The redundant fault class includes faults that the test generator considers undetectable. After the test pattern generator exhausts all patterns, it performs a special analysis to verify that the fault is undetectable under any conditions [30]. Figure 3.4 shows an example of a redundant fault. If D is a s-a-0 fault, output G will be stuck at 0, whatever values are applied to A,B,C.

3.4.2 ATPG Untestable Fault (AU)

The ATPG untestable fault class includes all faults for which the test generator is unable to find a pattern to create a test, and yet cannot prove the fault redundant. Testable faults become ATPG untestable faults because of constraints, or limitations, placed on the ATPG tool (such as a pin constraint or an insufficient sequential

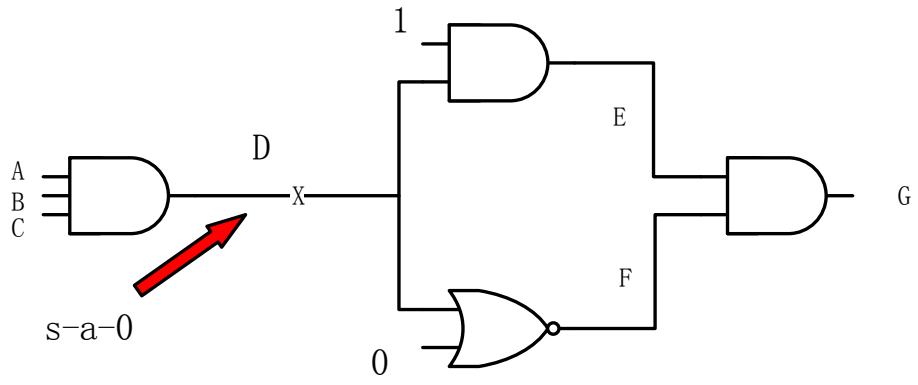

Figure 3.4: Example of redundant fault in circuitry.

depth). These faults may possibly be detectable, if you remove some constraint, or change some limitation on the test generator (such as removing a pin constraint or changing the sequential depth). You cannot detect more of them by increasing the test generator abort limit [30].

Chapter 4

Generating Test for Delay Faults Using Stuck-at Fault Tools

4.1 Background

A lot of work on delay testing can only be applied to scan circuits. A paper: Generating Test for Delay Faults in Nonscan Circuits [31] shows how to implement their proposed method in nonscan circuits. This proposed model augments the netlist of a circuit with a logic block in which testing a single stuck-at fault is equivalent to testing a path delay fault. This makes generating a test for path delay faults easier.

4.2 Three Major Phases in Path Delay Fault Testing

Initialization vectors, path activation vectors and propagation vectors are three kind of test vectors that enable a test to activate the path delay fault and propagate fault effects to primary outputs. We can observe the results at the output.

4.2.1 Initialization Sequence

In this phase, the initialization sequence vectors are V_0, $V_1 \cdots V_i$. Activating a clear signal will bring the flip-flops to the 0 state. If the clear signal is not applied to a flip-flop, then the flip-flop will go to a known state. At the end of the initialization sequence, all flip-flops are set in states required by the path activation vectors.

4.2.2 Path Activation Sequence

In this step, two consecutive vectors will be applied to the circuit. We denote these two vectors as (V_{i+1}, V_{i+2}) whose states are (S_{i+1}, S_{i+2}). Signals of both U0(X0)

and U1(X1) are specified only for V_{i+2}. Signals S0(00) and S1(11) indicate steady value for both vectors without static hazard. R(01) and F(10) are hazard free transition. XX is don't care.

V_{i+2} should arrive at the flip-flops before the application of clock period. For example, if a falling transition occurs at the destination flip-flop, the right value should be 0. And a path delay fault will be detected if the value captured in the flip-flop is 1. D indicates that an expected value of 0 in good circuit and 1 in faulty circuit. This can be indicated as second state.

4.2.3 Propagation Sequence

The main purpose of a third vector V_{i+3} is to propagate the fault effect to the primary output. This is similar to the D-algorithm, which also needs to propagate the fault effect to the output to be observed.

4.3 Test Generation Model

The Verilog netlist is modified in order to generate a test for a single stuck-at fault that can detect a path delay fault.

4.3.1 Test Specified for a Single Stuck-at Fault

1. Initialization vectors can precede path activation vectors if necessary. The single Stuck-at fault is required to be activated only after two vectors have been applied to combinational logic.

2. After the activation of the stuck-at fault, fault effect in the form of $\overline{D}$ or D will be injected to the destination flip-flop. The stuck-at fault must not influence the circuit before the activation of second vector.

3. After the flip-flops have captured the fault effect, the stuck-at fault should allow fault-free circuit function during propagation of the error to a primary output.

4.3.2 Test Model for Falling Transition at Destination Flip-flop

Figure 4.1 shows the test generation for a slow-to-fall path delay fault. The path a-c-e is the target path that lies between the two flip-flops FFS and FFD. The transition at e is a falling transition. A model within the dashed line is inserted into the circuit which makes e' the end point of the target path. AND1 and AND2 capture the signal requirement of the related path during V_{i+1} and V_{i+2}.The output of AND1 feeds the FF1 while the output of FF1 feeds AND2. FF1 begins at 0 state. After the second vector of activation, the result of AND1 will be propagated to an input of AND2. A stuck-at 1 fault is inserted at the output of AND2N. This stuck-at 1 fault will only be activated after two consecutive vectors of path activation are applied. The states of the output of AND2N could be defined as $\overline{D}$. This fault effect is required to be propagated to output of the path in order to observe i. The TERM gate (OR gate) and AND4N could be used to accomplish this task under the control of the Finite State Machine(FSM). The TERM gate is an OR gate since it is a falling transition.

1. Initialization phase: During this phase, the fault effect is not allowed to feed the FFD. At this time, the flip-flop FF2 stays at 1 state to ensure that. At the end of initialization, the FF2 will be cleared to a 0 state to guarantee that the output of AND4 gate remains before and after fault activation. Continuity can be provided through the TERM gate.

2. Activation phase: In the path activation phase, the FSM produces a 1 output to inject a $\overline{D}$ into FFD only when the path is activated and AND2 turns to 1. Subsequently, the FSM settles into a 1 state with a 0 output and remains in that state throughout the propagation phase. Figure 4.2 presents the FSM state diagram. As shown in Figure 4.1, the FSM is implemented with the single flip-flop FF2, which is clocked by Ck.

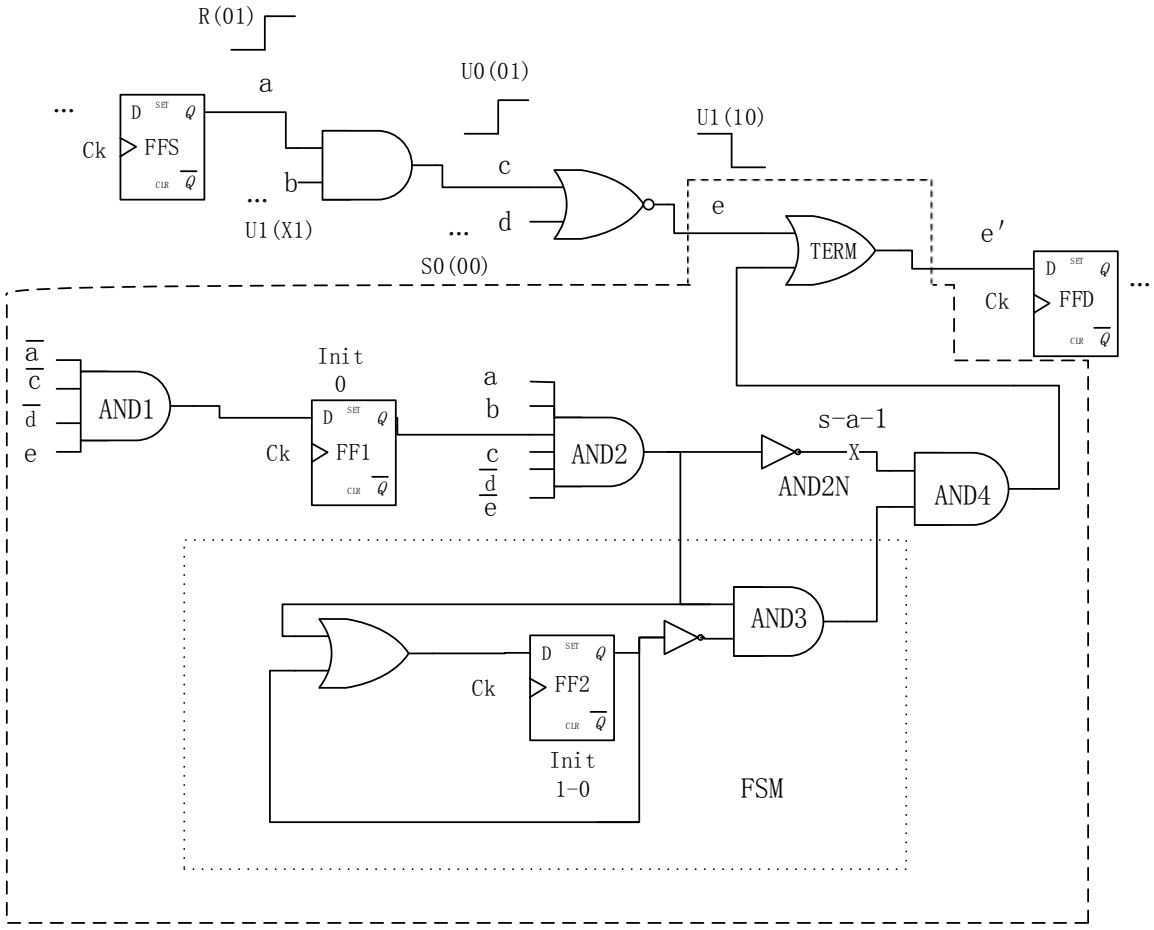

Figure 4.1: Test generation model for a falling transition [31].

When there is a rising transition at the destination flip-flop, the TERM gate will be a AND gate. Figure 4.3 shows the test generation for that.

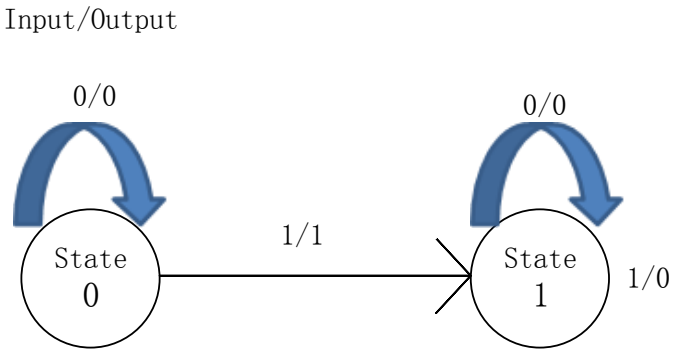

Figure 4.2: Operation of FSM state diagram [31].

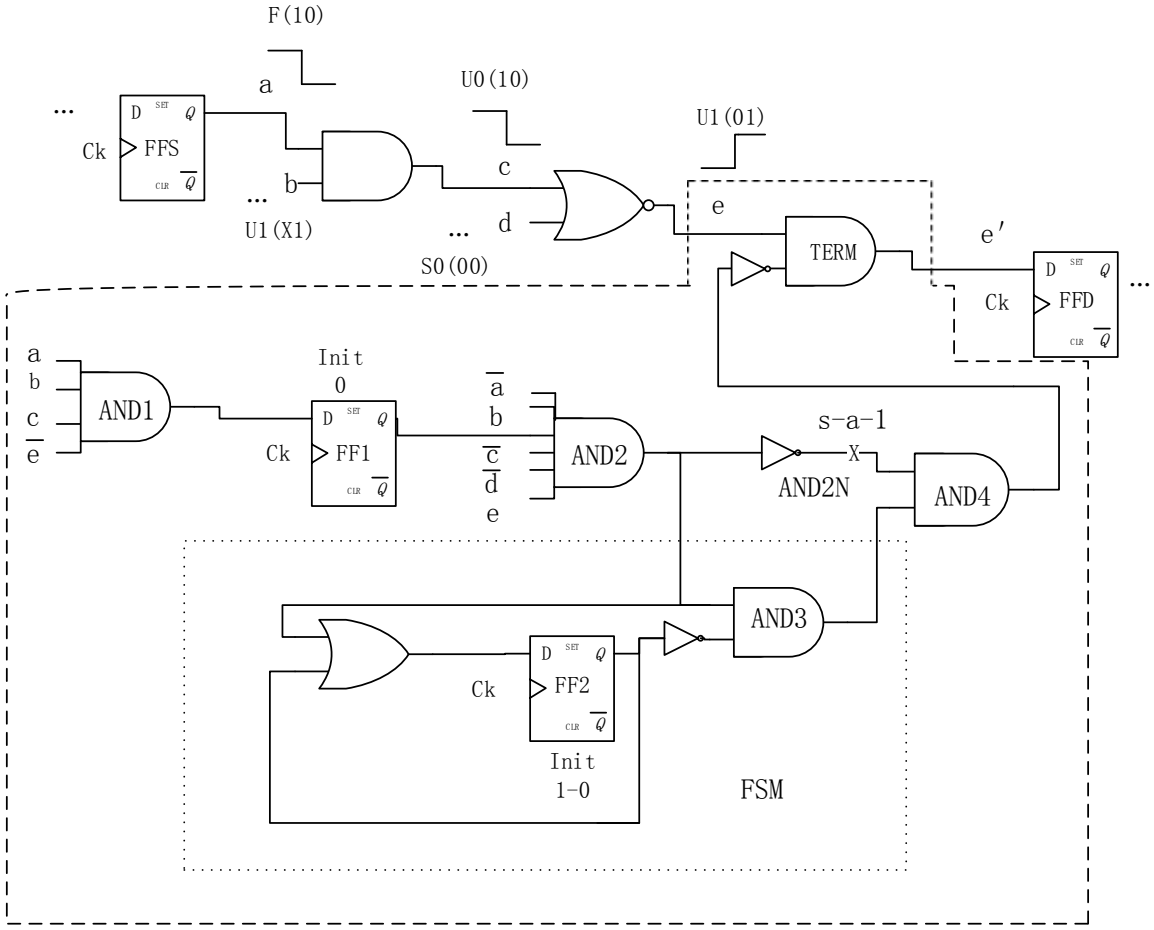

Figure 4.3: Test generation model for a rising transition at destination flip-flop [31].

Chapter 5

Background and Overview of Exclusive Test

In this chapter, exclusive test [33] is shown how to be applied to fault diagnosis. The purpose of diagnosis is to generate test vectors targeting pairs of faults. The different responses of the output can be used to distinguish faults, which increases the resolution of diagnosis. Distinguishing pairs of faults also reduces the size of the fault candidate list [32].

5.1 Background

5.1.1 XOR Gate

The XOR gate (sometimes EOR gate, or EXOR gate and pronounced as Exclusive OR gate) is a digital logic gate that implements an "exclusive or"; that is, a true output results if one, and only one, of the inputs to the gate is true. Table 5.1 shows the truth table of an XOR gate. The XOR gate can be used in half adder circuit. Moreover, as the name suggest, the XOR gate plays an important role in the exclusive test.

5.1.2 Exclusive Test for a Pair of Faults

An Exclusive test is to detect only one fault from a pair of targeted faults at a primary output. The object of exclusive test is a pair of fault set. A fault pair has two faults, F1 and F2. An exclusive test must detect one and only one of the two faults. There is circuit C0 which is fault free. C1 and C2 are same circuit which has F1 and F2, respectively. For clarity, we will only consider single output functions for

Table 5.1: Truth table of XOR gate.

Input0 A	Input1 B	Output A XOR B
0	0	0
0	1	1
1	0	1
1	1	0

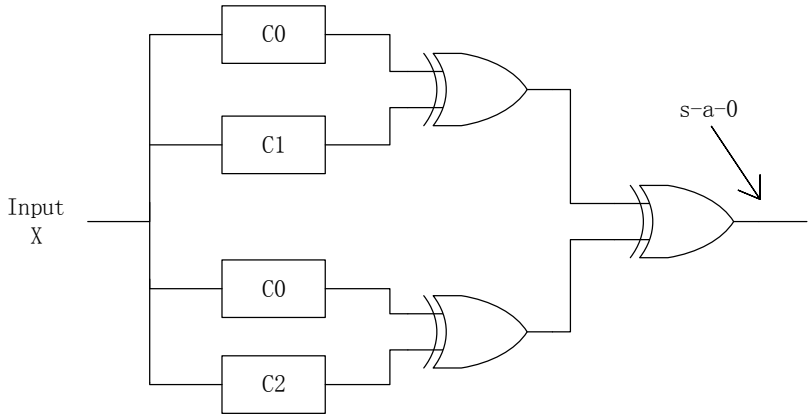

Figure 5.1: Exclusive test for two faults.

now. We show an example of a multiple output circuit at the end of this section. Figure 5.1 consists of three XOR gates and several circuits. In order to detect a stuck-at 0 fault at the output of the circuit, the input vector should generate a 1 at the output. This test is an exclusive test for the fault pair (F1,F2). This Boolean satisfiability formulation of the exclusive test problem is shown in Equation 5.1,

$$(C_0 \oplus C_1) \oplus (C_0 \oplus C_2) = 1; \tag{5.1}$$

Equation 5.1 can simplify to Equation 5.2,

$$(C_1 \oplus C_2) = 1; \tag{5.2}$$

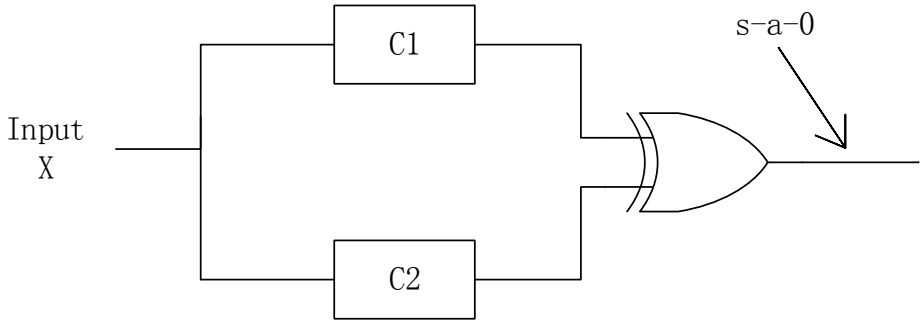

Figure 5.2: Exclusive test after simplification.

Equation 5.3 shows the test to detect the stuck-at 0 fault could distinguish the output of two circuits. This problem is also expressed as

$$(C_1 \oplus C_2) \oplus (C_0 \oplus C_0) = 1; \tag{5.3}$$

Equation 5.3 indicates that an exclusive test could be a test for a pair of faults in two copies of circuits. A different single fault is included in each copy of the circuit under test producing a single output through an Exclusive-OR gate. This problem could also be adapted to a single fault ATPG under an alternative approach. In an exclusive test, if no test exists for these faults, the two faults may be equivalent or redundant. If these two faults are independent, there will be no vector that can detect both of them. Then, there exists a test that detects only one of the two faults.

5.2 Boolean Analysis of New Exclusive Test Algorithm

An exclusive test generation algorithm can simplify the DATPG to a single stuck-at fault problem. A new primary input will be inserted in the CUT. The stuck-at fault is inserted at a new added primary input pin. The existence of the test will be an exclusive test for the two faults under analysis.

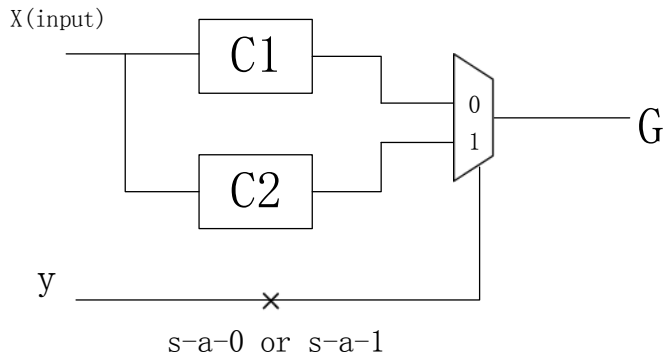

Figure 5.3: A CUT for exclusive test.

Boolean algebra is used for the analysis of the exclusive test. Figure 5.3 shows the single ATPG problem. The new added primary input is the control signal of the multiplexer.

$$A \oplus B = A\overline{B} + \overline{A}B; \tag{5.4}$$

Equation 5.4 shows the XOR function. Based on Figure 5.3, we could get Equation 5.5 which shows the function clearly implemented as Shannon's expansion [34] for G.

$$G(X, y) == \overline{y}C1 + yC2; \tag{5.5}$$

Detecting either a stuck-at 0 or stuck-at 1 fault on y, equation 5.6 shows the expression of this problem. This is same as equation 5.2. Thus we prove that a vector X that detects either stuck-at 0 or stuck-at 1 fault at y in the circuit G(X,y) of Figure 5.3 is also able to detect the stuck-at 0 fault in the circuit of Figure 5.2.

$$\frac{\partial G}{\partial y} = G(X, 0) \oplus G(X, 1) = C1 \oplus C2 = 1; \tag{5.6}$$

Equation 5.2 indicates the C1 is not equal to C2.

5.3 Diagnostic Metric

When we want to measure how long a desk is or the weight of a chair, we need a special unit for them. Fault diagnosis also needs various types of units to measure. In this section, some criterion of fault diagnosis will be shown.

5.3.1 Fault Coverage (FC)

Fault coverage is the percentage of faults detected from all faults that test pattern set tests, treating untestable faults the same as undetected faults [30]. 100% FC means all of the modeled faults are detected by test vectors FastScan calculates FC using the Equation 5.7 :

$$FC = \frac{Number\ of\ detected\ faults}{Total\ number\ of\ faults}. \tag{5.7}$$

5.3.2 Diagnostic Resolution (DR)

In fault diagnosis, diagnostic resolution measures the quality of a given test set. Equation 5.8 is the expression of DR.

$$DR = \frac{Total\ number\ of\ faults}{Number\ of\ syndromes(signatures)}. \tag{5.8}$$

From the equation we can see that DR gives us the average of faults per group. In the detection test period, which is before the exclusive test phase, each fault counts once since the equivalent fault class is unknown. After the exclusive test, the total number of faults is reduced for the reason that more fault groups have already been built. A perfect DR of 1.0 indicates that all of the faults groups are identified which means each one fault could represent each equivalent fault class.

Table 5.2 shows a modified dictionary. For a full-response dictionary, there are four different signatures: 101010, 000101,001010,100000 from four faults. As a result:

Table 5.2: Modified dictionary.

Faults	Test1 signature	Test2 signature	Test3 signature	index
F1	1/10	1/10	1/10	1
F2	0/00	1/01	1/01	2
F3	0/00	1/10	1/10	2
F4	1/10	0/00	0/00	3

the DR is 4/4=1 which is also a perfect DR. But for a pass-fail dictionary, there are three unique signatures: (111,011,100) from 4 faults. So the DR=4/3=1.33. Two main kinds of fault sets are diagnosed fault sets and undiagnosed sets. Undiagnosed fault sets means that there are at least two faults that have same syndromes. During the diagnosis period, more pairs of faults are selected from undiagnosed faults sets to be diagnosed in order to achieve a satisfactory diagnosis resolution. Fault dictionary will be updated and faults regrouped when exclusive test simulates more faults. However, dictionary based diagnosis methods also have limitations. They always require substantial storage space.

5.3.3 Diagnostic Coverage (DC)

For a set of test vectors, a fault group is such that each fault in the group is distinguishable from all other faults in every other fault group. Faults in a same group hold the same signature while faults from different groups have different signatures. If there is a new test that only detects a part of faults in a group then this group will be portioned into two groups. One of the two new groups contains the faults that can be detected by the new test vectors. The other group consists of the rest of the faults that cannot be detected by the test.

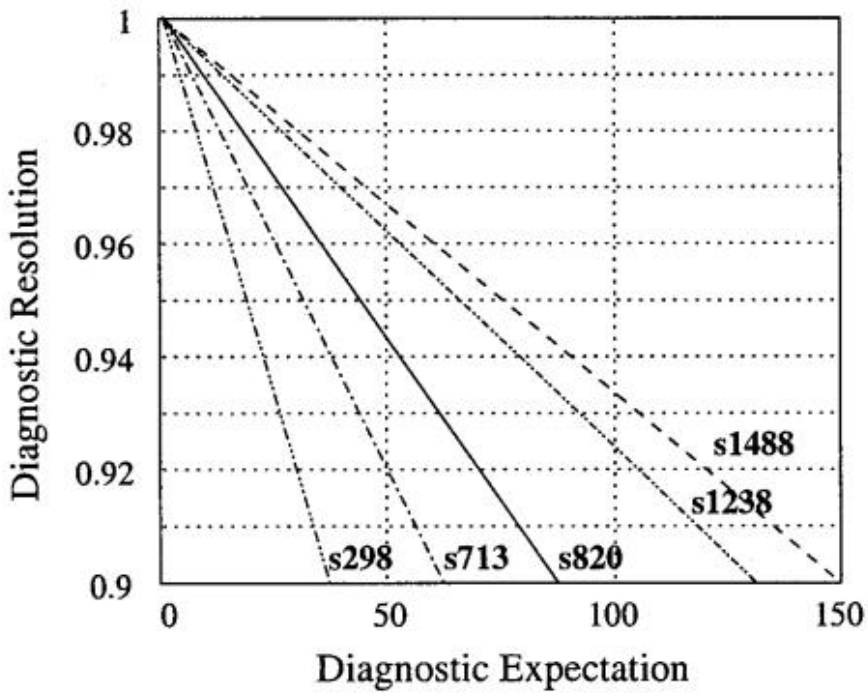

Figure 5.4: Relationship between DR and DE.

If there are enough test vectors that distinguish between each fault pair, the number of fault groups will be equal to the total number of faults. In other word, each fault group has only one fault.

The original group is defined as G_0 before fault diagnosis. As there are more and more tests generated for distinguishing the faults, new fault groups are constructed as some new detected faults leave the original groups. G_1, $G_2 \cdots G_n$ are new group names. If each fault group has only one fault, $n = N$ (total number of faults). The diagnostic coverage is defined as Equation 5.9

$$DC = \frac{Total\ number\ of\ detected\ fault\ groups}{Total\ number\ of\ faults}; \tag{5.9}$$

DC =1 means that each group has only one fault which is also a perfect diagnosis. It is easy to see that DC is actually the reciprocal of DR which we previously defined.

5.3.4 Other Kinds of Diagnostic Metrics

There are also some other kinds of diagnostic metrics. In a given circuit, the diagnostic power is the fraction of faults that are fully distinguished [35].

Another diagnostic metric is diagnostic expectation (DE) [36]. The DE is the expected size of a fault's indistinguishable class resulting from diagnosis if the probability of each fault is assumed to be the same. Figure 5.4 shows the relationship between DR and DE in several types of benchmark circuits. As what we can see from the figure, DR of larger circuits decreases as DE increases. As a result, in order to keep high diagnostic expectation of larger circuits, a high diagnostic resolution is required.

5.4 Multiple Output Circuits

Exclusive test can be also applied to multiple output circuits. Some of the circuits are not required to modify the circuit in order to have only one single output. Each pair of outputs could be added a XOR gate to construct the exclusive test model. The advantage of multiple output circuit models is that more outputs for observing and propagating the fault effects can be used. Moreover, if two faults are detected on two different outputs of a multiple output circuit, these two faults can be distinguished and diagnosed. Similarly, if a pair of faults is established to be equivalent in a multiple output circuit, the pair of these faults can also be found equivalent in single output circuit. A multiple output circuit model can be regarded as the extension of single output model.

Chapter 6

Diagnostic Test Generation for Path Delay Faults

As we have discussed before, many failed circuits of modern VLSI chips have relationships to timing issues. Sometimes, if the clock period is so short or there is delay along a path, it may result in a violation of setup time. The violation of long path constraint and short path constraint can also cause time-related problems. Diagnosis of a timing related problem helps improve the yields of chips and product quality.

Some types of delay fault models have been introduced previously. The path fault detection can be done by generating test patterns for a single stuck-at fault. Moreover, path delay fault also requires considering the off-path signal, which increases the complexity of the model.

We propose a model that can diagnose path delay faults. The algorithm of the model may also be further extended to be applied to other fault models. With this diagnosis system, we can also distinguish between various types of faults. This greatly extends the range of application of this method.

6.1 Setting the Goal

The work of this chapter enhances the path delay fault diagnosis ability with existing tools. We focus on using the existing techniques to implement our model. The basic tool we use is the ATPG simulation and test pattern generation for detecting a single stuck-at fault.

Launch-off-capture (LOC) and launch-off-shift (LOS) are important ways of conduction of scan test. These will be discussed in detail in the latter part. In scan testing, the first vectors are shifted through the scan chain to the scan flip-flop. Then the second vector may be produced by clocking the circuit in the normal mode (launch-off-capture test) or in the scan mode (launch-off-shift or LOS test), following which the response is captured in the scan register in the normal mode and scanned out in the scan mode. As we previously discussed, SCANEN switches the circuit mode between scan mode and normal mode. However, the SCANEN may cause problems while using it in LOS. As a result, LOC is more widely used than LOS today.

The Diagnostic Coverage metric will be used in this thesis to evaluate the effectiveness of exclusive test. The new modeling technique is quite efficient and easy to be implemented. Our work contains two parts: detection test phase and exclusive test phase.

6.2 Path Delay Fault

In this chapter, path delay fault will be discussed in detail. The delay defect in the circuit is assumed to cause the cumulative delay of a combinational path to exceed some specified duration [13]. The specified duration can be the duration of a clock period or the vector period. Propagation delay is how long a signal event will take in order to traverse the path.

The total number of the path delay faults is twice the number of physical paths in the circuit since each path may have slow-to-rise or slow-to-fall faults. Path delay faults are more complicated than transition delay fault. There are several types of path delay fault tests.

6.2.1 Robust Path-delay Test

A robust path delay test guarantees to produce an incorrect value at the destination if the delay of the path under test exceeds a specified time interval (or clock period), irrespective of the delay distribution in the circuit [13]. When the gating inputs used to sensitize the path are stable from the time of the launch event to the time of the capture event, the robust detection can be used. Robust detection keeps the gating of the path constant during fault detection. Therefore, it will not affect the path timing. Because it avoids any possible reconvergent timing effects, it is the most desirable type of detection and for that reason is the approach FastScan tries first.

Fast and slow transition on a path is shown in Figure 6.1. The path delay fault test requires a vector pair (V1, V2) to detect the fault at the output. The initial value (0) is steady-state output of V1 and final value (1) is output of V2. Fast transition means that the transitions propagating through paths whose delays are smaller than clock period. Slow transitions indicate that propagating time through paths with delays is greater than clock period.

In order to measure the delay of a path the following properties are required:

1. Transition from the initial value to final value should be a real event. Because a real event can exist without help from others.

2. This controlling event doesn't allow other events to appear before it occurs. As a result, the output value will remain the initial value until the controlling event occurs at the output.

Figure 6.2 shows an example of the robust detection. Robust detection occurs when the gating inputs used to sensitize the path stable from the time of the launch event to the time of the capture event. The off-path of the target circuit is able to sensitize the target path in both initial state and after transition state.

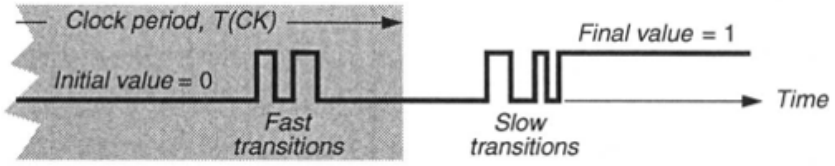

Figure 6.1: Fast and slow transitions on a path [13].

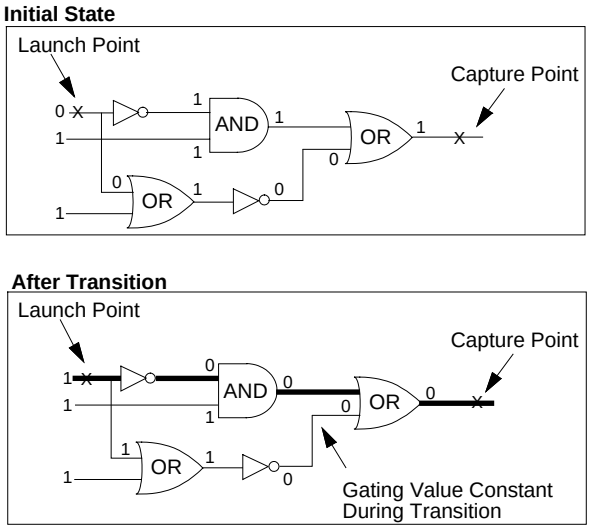

Figure 6.2: Robust detection example [30].

6.2.2 Non-Robust Path-delay Test

Non-Robust Path-delay Test detects a path-delay fault without the presence of other path-delay faults. The path-delay fault for which a non-robust test exists is defined as single-testable path-delay fault [37].

After applying a pair of vectors which cause a transition at the input of a path, we can measure the output value after a period (usually the clock period.) The expected output value should be uniquely controlled by the transition propagating through the path.

Referring to Figure 6.4, the path B, E, G, J and K is our target path. So signals B, E, G, J and K can be called on-path signals. Off-path signals represent the signals

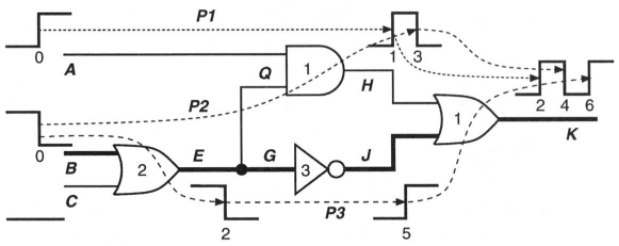

Figure 6.3: Non-robust detection test [13].

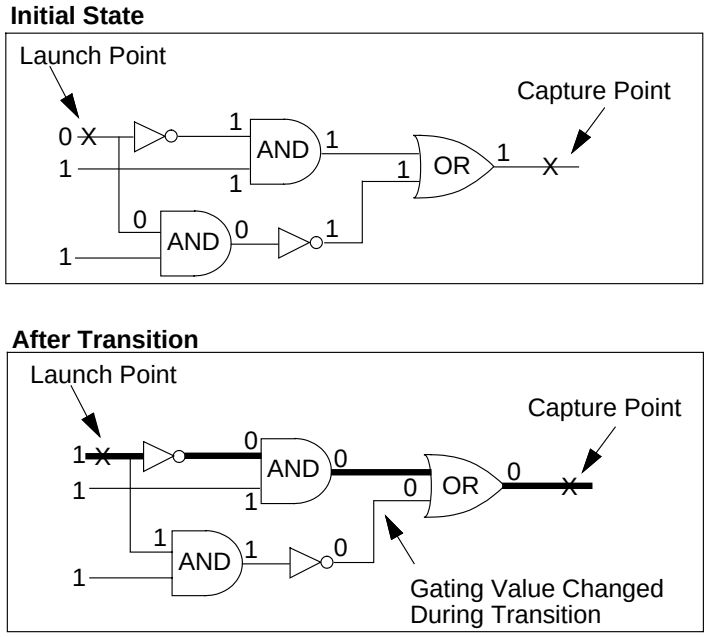

Figure 6.4: Non-robust detection example [30].

that are not on the target path. The vector pair (V1,V2) = (010,000) produces a falling transition at B to test the fault at P3. After the application of second vector V2, all of the off-path input signals should be non-controlling values. This condition is defined as static sensitization. For example, 0 will be fed into each OR or NOR gate while 1 is fed to each AND or NAND gate. In figure 6.3, transitions occur at P1 and P3, but only P3 is static sensitization. Therefore, non-robust test is only achieved for the fault in P3.

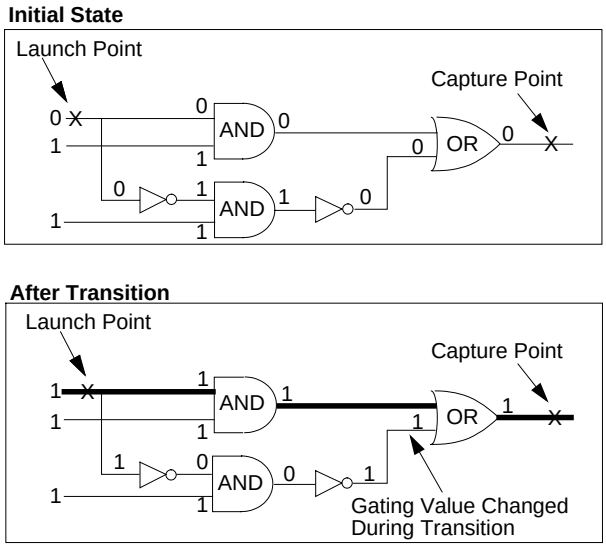

Figure 6.5: Functional detection example [30].

Figure 6.4 shows the non-robust detection test in detail. After the second vector, the off-path signal should all be non-controlling.The gating value on the OR gate changed during the 0 to 1 transition placed at the launch point. Thus, the proper gating value was only at the OR gate at the capture event.

6.2.3 Functional Detection Test

Functional detection test has fewer requirements than non-robust test or robust test. Functional detection test further releases the requirements on the gate inputs used to sensitize the path. The input gate signals of the path do not have to be stable as in robust detection, nor does it have to be sensitizing at the capture event, as required by non-robust detection. Functional detection requires only that the gating inputs not block propagation of a transition along the path. FastScan places faults detected by functional detection in the det_functional (DF) fault class.

Figure 6.5 gives an example of functional detection for a rising transition. The off path signal of the OR gate is neither stable, nor sensitizing the at the time of capture

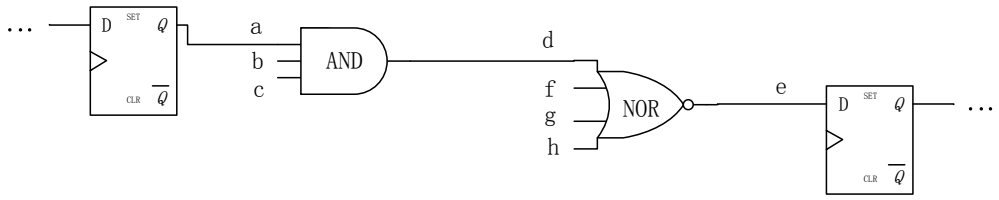

Figure 6.6: A scan circuit example of path a-d-e.

event. However, the path input transition still propagates to the path output, because the on-path signal of the OR gate during capture cycle is controlling value.

6.3 Modeling a Path Delay Fault

We need to use the synchronous model for path delay faults. Figure 6.6 shows a path of a scan circuit. Figure 6.7 shows a device of modeling a slow-to-fall path delay fault on the path a-d-e with a synchronous sequential circuit. The logic that is within the dotted line does not belong to the original circuit. We insert this model in order to model a slow-to-fall path delay fault. Therefore, e' becomes the new path output. Any fault that detects a s-a-0 fault at signal OUT1 will detect a slow-to-fall fault at path a-d-e.

The flip-flop FF1 which is placed between AND2 and AND3 is initialized be the 0 value. The signal requirement of the input of these gates is based on the target path signal requirement which we discussed previously. The first vector is required to set e as 1. Input of AND2 will be set based on signal requirements. Second vector will force the e to be 0 in order to make a falling transition. The signal requirement of second vector is placed at the input of AND3. Because the FF1 is initialized to 0, which is fed to AND3, the signal Out1 will be 0, propagating the value of signal e to e' in the first phase. Since signal e is 1 at the same time, it makes e' as 1.

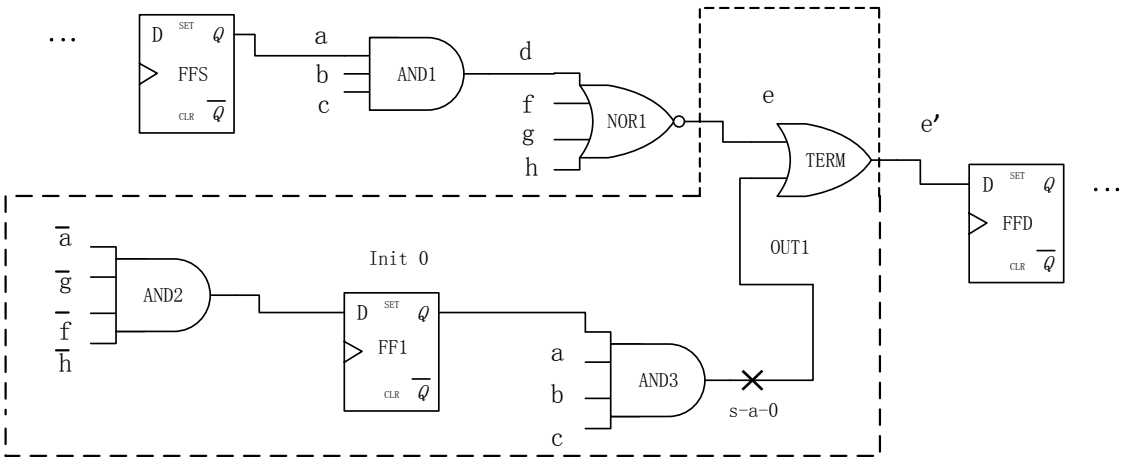

Figure 6.7: Model of a slow-to-fall fault in path a-d-e.

At this time, the output of AND2 comes at the input of FF1. Since signal a, b, c in the gate AND3 propagates the value to Out1, they will propagate 1 to the TERM gate, and e' will be forced to 1 since it is an output of a OR gate. This process models a slow-to-fall path delay fault.

Figure 6.8 shows the architecture of modeling a slow-to-rise path delay fault. The difference is the gate is the AND gate placed after e. And any fault that detects a s-a-0 fault at signal OUT1 will detect a slow-to-rise fault at path a-d-e.

6.4 ATPG Model of Path Delay Faults

ATPG model is actually a Verilog netlist of the circuit under test (CUT). Some logics are inserted into the netlist in order to model the path delay fault.

The exclusive test for a pair of path delay faults should be a single output model. In this exclusive test, only one of these faults will be activated. Suppose the model is a multi-output model. It will be quite possible to activate both of the two faults at the same time which makes the test inefficient. As a result, a single output model is our choice.

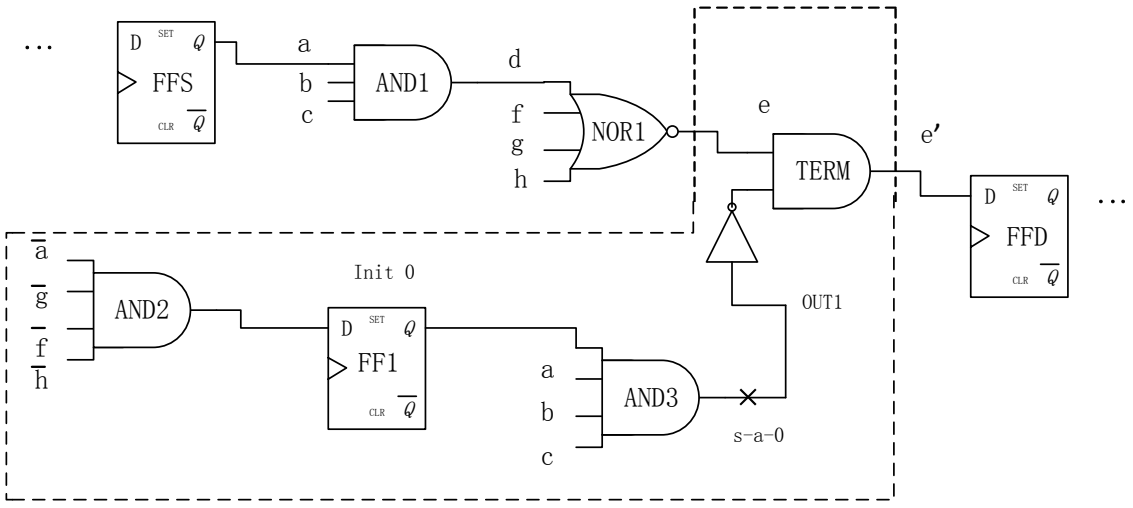

Figure 6.8: Model of a slow-to-rise fault in path a-d-e.

The ATPG model in Figure 6.9 presents the conventional Boolean formulation. The model we construct can model the two path delay faults. One is the slow-to-fall fault on path a-d-e while the other one is on path m-k-h-e with the same transition. The models of delay faults on path a-d-e and path m-k-h-e are placed in parallel. Note that a 1 output from the XOR gate cannot be obtained by a single vector. In order to detect a stuck-at 0 fault at the output of XOR gate, the value in Out1 and Out2 should be different. Both of the OR gate Term1 and Term2 has an input that is connected to e in the original path. The other input of these two OR gates is connected to the former AND gate. In other words, if the In03 is different from In06, a stuck-at 0 fault at the output of XOR gate will be detected.

The XOR gate in Figure 6.9 can be replaced by a multiplexer. The control signal of the multiplexer is the new adding primary input pin. Any test that could detects a stuck-at 0 or stuck-at 1 fault in the control signal will be the test that only activates the one of these two faults in the path.

For example, the vector pair (011,111) is applied to a, b, c, and creates a falling transition at e. After the first vector, both Out1 and Out2 will be 1 since the e is 1 which is a controlling value for the two TERM gates(OR gates). The pair of

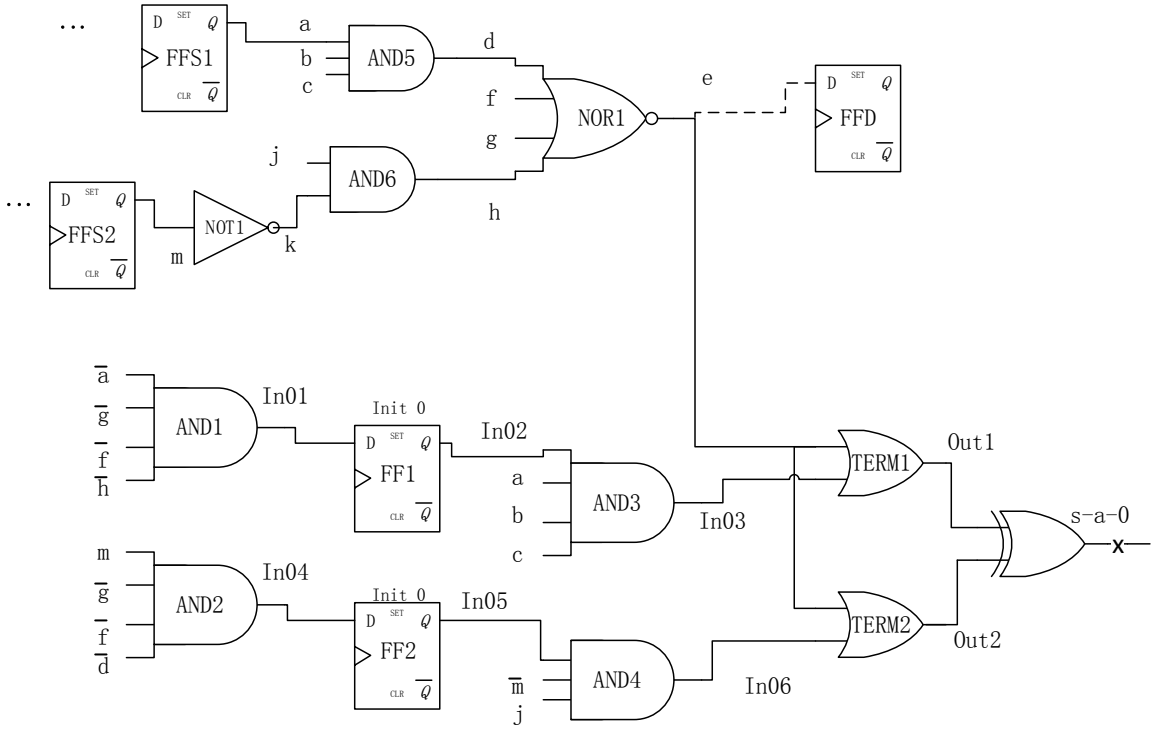

Figure 6.9: An ATPG model: test that detect a s-a-0 fault distinguish a pair of slow-to-fall faults.

vectors creates a transition at the input of a-d-e. But no transition is created at path m-k-h-e. Therefore, value of Out1 is 1 while the value on Out2 is 0 after applying the second vector. This condition makes a difference at Out1 and Out 2 that enables ATPG to generate a test pattern to detect the stuck-at 0 fault at output of XOR gate. Figure 6.10 shows an ATPG model to detect a slow-to-rise fault.

6.4.1 Scan Circuit Test

For a scanned sequential circuit under test(CUT), ATPG will generate two-vector tests. The vectors are generated by a scan program of ATPG to accommodate the modeling flip-flop. Either a Launch-off-capture sequence or Launch-off-shift sequence can be applied to generate the second vector.

Figure 6.11 shows the model for distinguishing two slow-to-fall path delay faults. Out3 will feed the input of the destination flip-flop. Similarly, the model for dis-

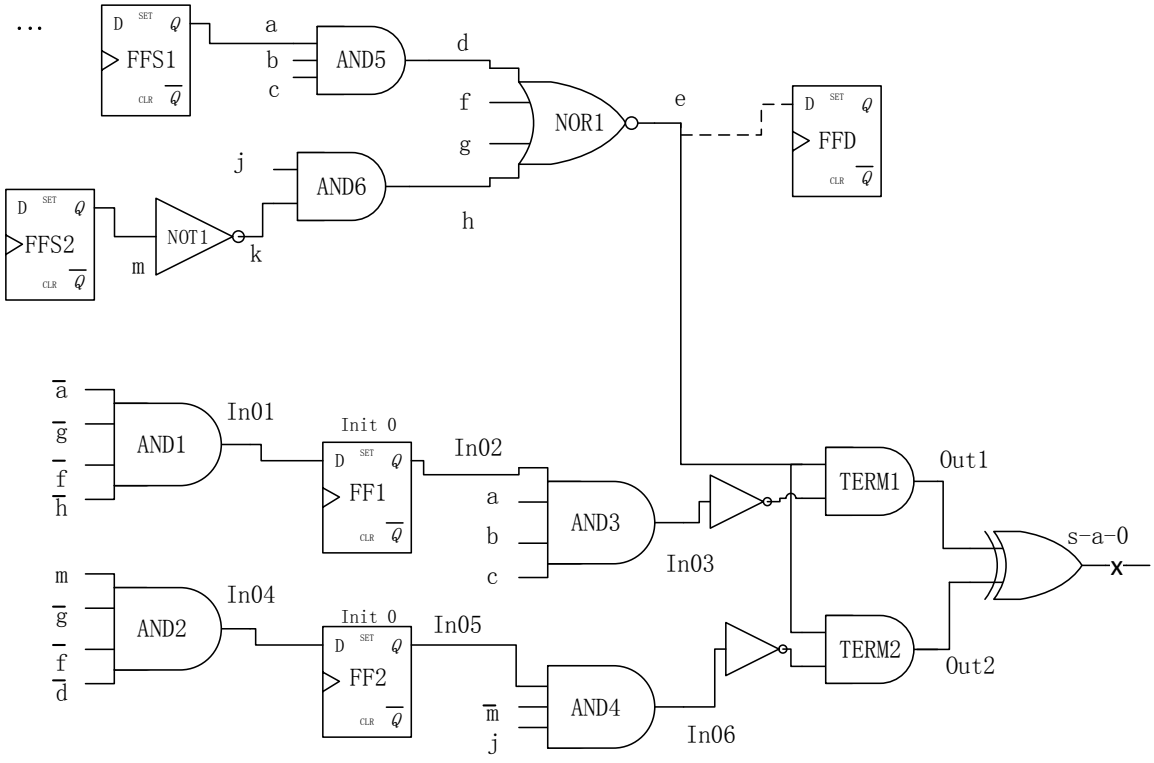

Figure 6.10: An ATPG model: test that detect a s-a-0 fault distinguish a pair of slow-to-rise faults.

tinguishing two slow-to-rise path delay faults can also be constructed, as shown in Figure 6.12.

6.5 Scan-Based At-Speed Test Generation

The Mentor Graphics Fastscan ATPG tools can generate test patterns for path delay faults. Actually, the test is in LOC form which consists of a scan-in sequence and primary input vectors. Our work consists of two parts: detection test and exclusive test.

6.6 Detection Test Phase

During the detection phase, ATPG tools are used to diagnosis path delay faults. Diagnostic coverage (DC) will then be calculated. Whether to generate exclusive

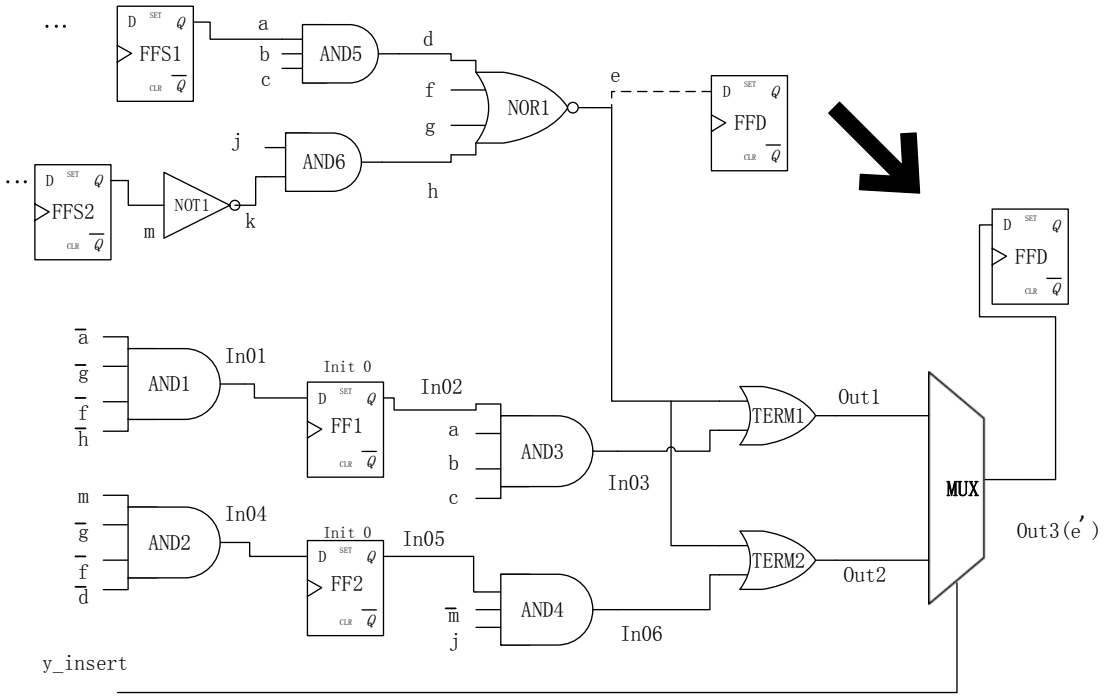

Figure 6.11: ATPG test model: distinguish two slow-to-fall faults.

tests on undistinguished fault groups depends on the DC. Figure 6.13 shows the test flow of the detection phase.

Test patterns generated by ATPG will try to detect the path delay faults in the circuit. DFTadvisor [38] is used to full-scan a benchmark circuit with a single scan chain. Then test patterns will try to activate the path delay faults with fault simulator. Meanwhile, each fault will store one or several patterns(signatures) that could activate the fault. A diagnostic dictionary is constructed after diagnostic fault simulation. The dictionary is a pass/fail dictionary which is a compact dictionary. It only stores pass/fail information for each fault corresponding to test patterns. Fault simulation consists of four steps. First, we need to find detected faults with input test vectors. And then we will group faults with same signature. Thirdly, diagnostic coverage is calculated. Finally, the test will go through step 1 to step 3 until no vectors left.

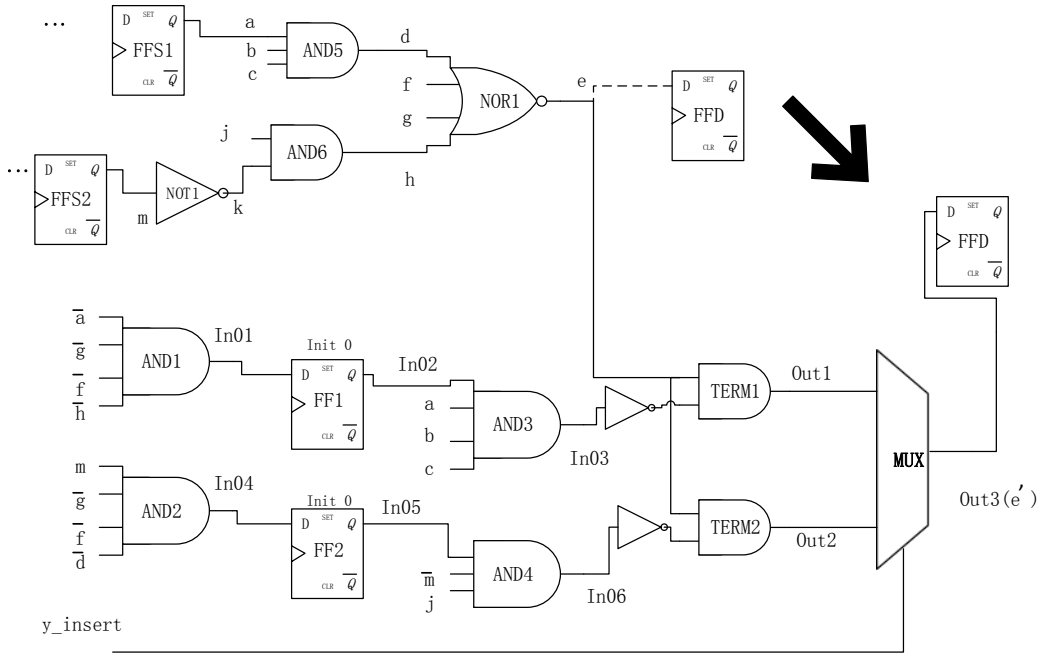

Figure 6.12: ATPG test model: distinguish two slow-to-rise faults.

6.6.1 Fullscan Circuit

The benchmark circuit is full-scanned by DFTadvisor. All D flip-flops have a multiplexer added at the input to make them scan flip-flops.

6.6.2 Construction of Diagnostic Dictionary

Given a test pattern, good machine simulation can predict the logic values in a good circuit. It uses the test pattern to activate the path delay faults in the fault lists. Fault diagnosis is performed by observing the failures on test to the expected response to every test (signature) in the dictionary. The cost of physical defect localization depends on the size of the fault set, size of tests and capability of the dictionary. Faults activated by same set of test patterns will be assigned into the same group.

Table 6.1 shows the total group numbers and the diagnostic coverage. If two outputs of the paths are not located at same place, the two faults from these paths

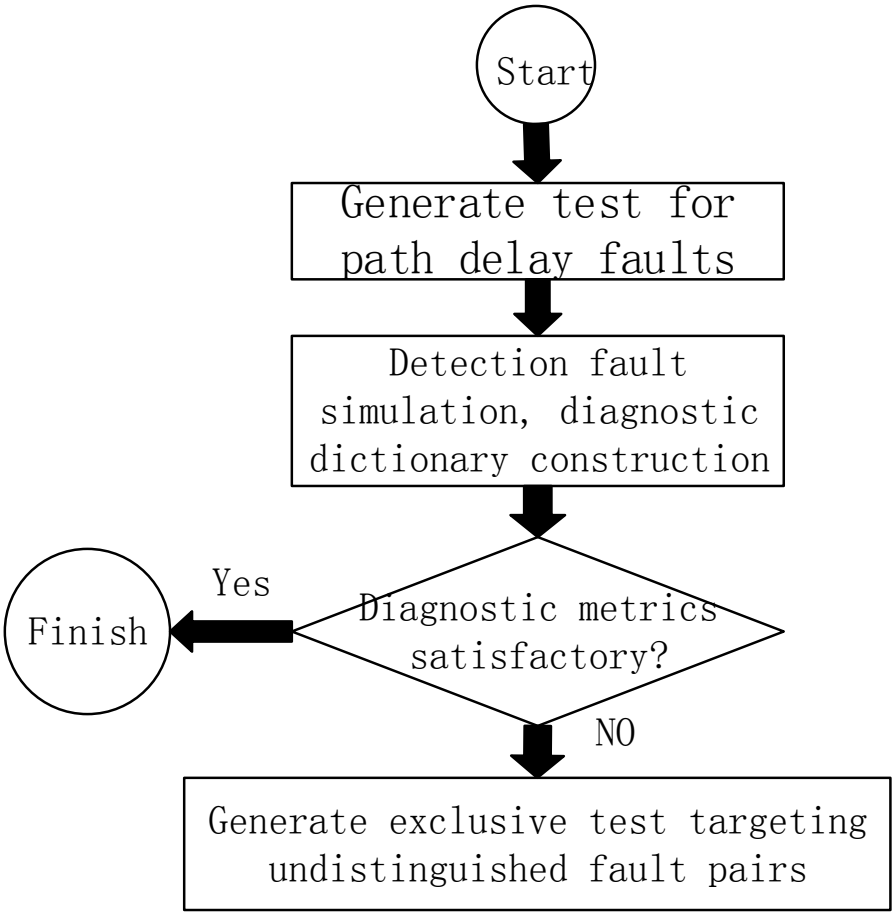

Figure 6.13: Detection test flow in detection phase.

can be distinguished. Moreover, diagnostic dictionary also reveals that different transitions of the path delay faults in the same output can also be distinguished. As a result, the total number of groups will be twice the total numbers of outputs of the scan circuit path. The total number of outputs of the scan circuit is equal to the number of flip-flops (number of output of scan circuit).

6.6.3 The Need for Generating Exclusive Test

The diagnostic coverage (DC) of path delay faults in these circuits shown in Table 6.1 is usually less than 20%. All of them are in need of generating exclusive

Table 6.1: Diagnostic coverage after detection test phase

Circuit	Diagnostic coverage	Total flip-flops	Total groups
s298	16.87%	14	28
s382	13.38%	21	42
s420	18.10%	16	32

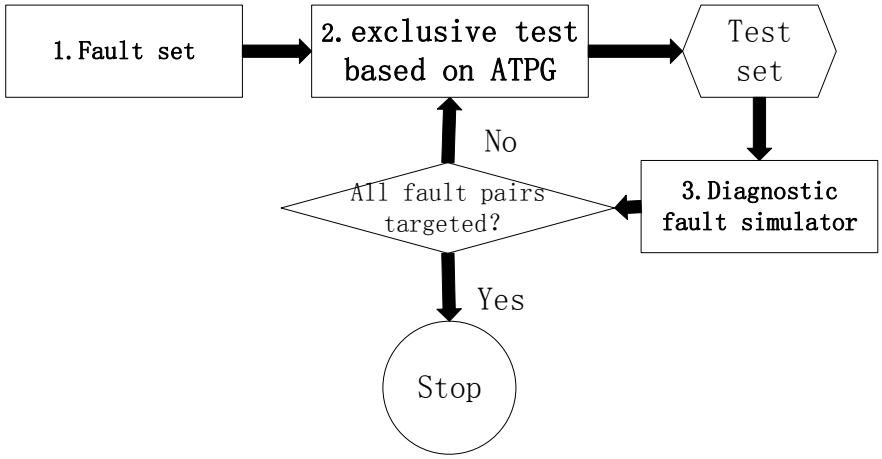

Figure 6.14: Flowchart of automatic exclusive test generation system.

tests. Exclusive test on them will pick out some faults in a group to constitute new groups. Therefore, new groups increase the DC.

6.7 Exclusive Test Phase

The automatic exclusive test generation system will generate exclusive tests for a given circuit to improve diagnostic coverage (DC). The system is used for our test generation and the flowchart is shown in Figure 6.14 .

The whole system is implemented in Perl programming language [39] and consists of several functional blocks. Block 1 represents the fault sets that are constituted by conventional detection ATPG system in detection phase. Exclusive test generation is performed on Block 2. Block 3 is a diagnostic fault simulator.

```
PATH "path0" =
    PIN /I$6/Q + ;
    PIN /I$35/B0 + ;
    PIN /I$35/C0 + ;
    PIN /I$1/I$650/IN + ;
    PIN /I$1/I$650/OUT - ;
    PIN /I$1/I$951/I$1/IN - ;
    PIN /I$1/I$951/I$1/OUT + ;
    PIN /A_EQ_B + ;
END ;
```

Figure 6.15: An example of path definition file.

After fault simulation of a detection test, faults are partitioned into several groups based on different signatures. Exclusive test is then performed on the undistinguished fault pairs. There will be two possible cases. One case is that we can find a test. The two faults will then be assigned to two different groups. The other case is that no test is found. The reason for that will be discussed later in the section.

6.7.1 Path Definition file

ATPG helps diagnose the path delay faults. A path definition file [30] describes the paths that we want in the test set. For each path, we must specify two things. One is the path name that identifies a path. The other is path definition which is the topology of the path from launch to capture point as defined by an ordered list of pin pathnames. Each path must be unique.

Figure 6.15 shows the definition of a path. PIN identifies a pin in the path by its full pin pathname. Pin statements are required to be ordered from launch points to capture points. A "+"or "-" indicates the inversion with respect to the launch point. A "+" indicates no inversion while a "-" indicates inversion. The path definition file shows the start point and end point of a target path. Any gate that is on the target path is shown in the definition file.

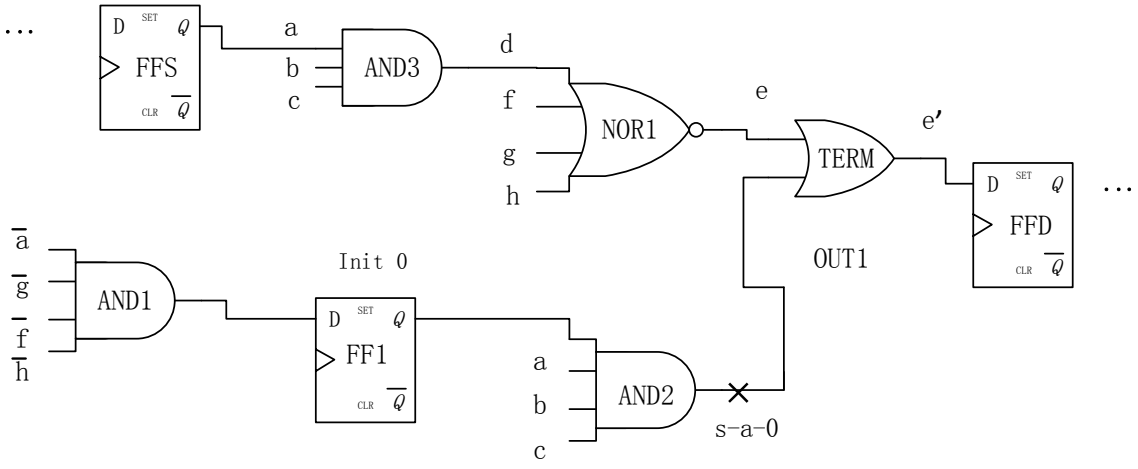

Figure 6.16: Position of AND gate in test model.

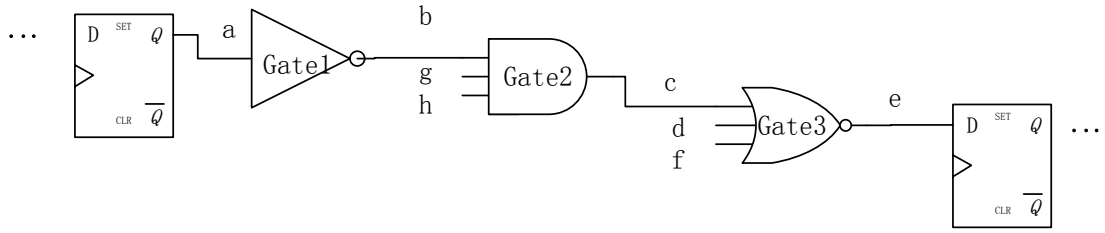

Figure 6.17: An example a path delay fault model.

6.7.2 Construction of Exclusive Test Model

Signal Requirement for Modeling a Path Delay Fault

The exclusive test model is shown in Figure 6.11. The input signals requirement of the model depends on the transition at the output of the path and the gate type.

In this section, we will show the methods for preparing input values (which are signal requirements) for the AND gate of our test model. Figure 6.16 shows the position of the AND gate(AND1 and AND2).

The path definition file shows the types of gates and which input the target path goes through. Our purpose is to propagate the transition of an input to the output of the path. The path definition file help us find the configuration of each gate's input signal.

Table 6.2: Signal requirement for signal e to be 1.

Signal requirement	Value
e	1
c,d,f	0
b	0
a	1

Table 6.3: Signal requirement for signal e to be 0.

Signal requirement	Value
e	0
c	1
g,b,h	1
a	0

Figure 6.17 shows a path that starts from a and ends at e. Either a slow-to-fall or slow-to-rise path delay fault needs a transition at the output which requires a 1 or 0. If e is required to be 1, then we can go back through the path to find signal requirements of each gate. Since Gate3 is a NOR gate, signals d, c, f are all required to be 0 in order to make e as 1. Thus the signal requirement for Gate3 has been set. And then we still move backward along the target path. It's easy to find that c is set to be 0. Since Gate2 is a AND gate whose output is 0, only the signal b needs to be 0. Because b is an on-path signal and 0 is controlling value for an AND gate, the rest of the input signals in Gate2 can be ignored. We will go back through the path until the start point of the path. Similarly, if e is required to be 0, the signal requirement can also be found. Therefore, the signal requirement of a slow-to-fall path delay fault at path a-b-c-e has been constructed as Table 6.2 and Table 6.3. This method is similar to the D-algorithm.

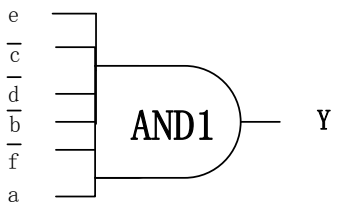

Figure 6.18: An example an AND gate for Table 6.2.

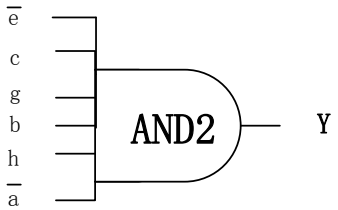

Figure 6.19: An example an AND gate for Table 6.3.

If the signal is required to be 1, the signal will be directly connected to the input of an AND gate in the exclusive test model. Otherwise, an inverter gate is required to be placed, and the AND gate can be constructed in Figure 6.18 and Figure 6.19.

This method ensures that the minimum requirement for modeling a path delay fault is obtained. It helps decrease the complexity of building a AND gate based on the information of gates.

Modification of Original Verilog Netlist

In Figure 6.11, the control signal y_insert is added to the primary input of the original circuit. A stuck-at 0 or stuck-at 1 fault will be inserted on the y_insert signal. The output of the multiplexer is connected to the input of the scan flip-flop. Moreover, the e signal, which is the output of original circuit, will be connected to two OR gates of the test model instead of being connected to the input of scan flip-flop.

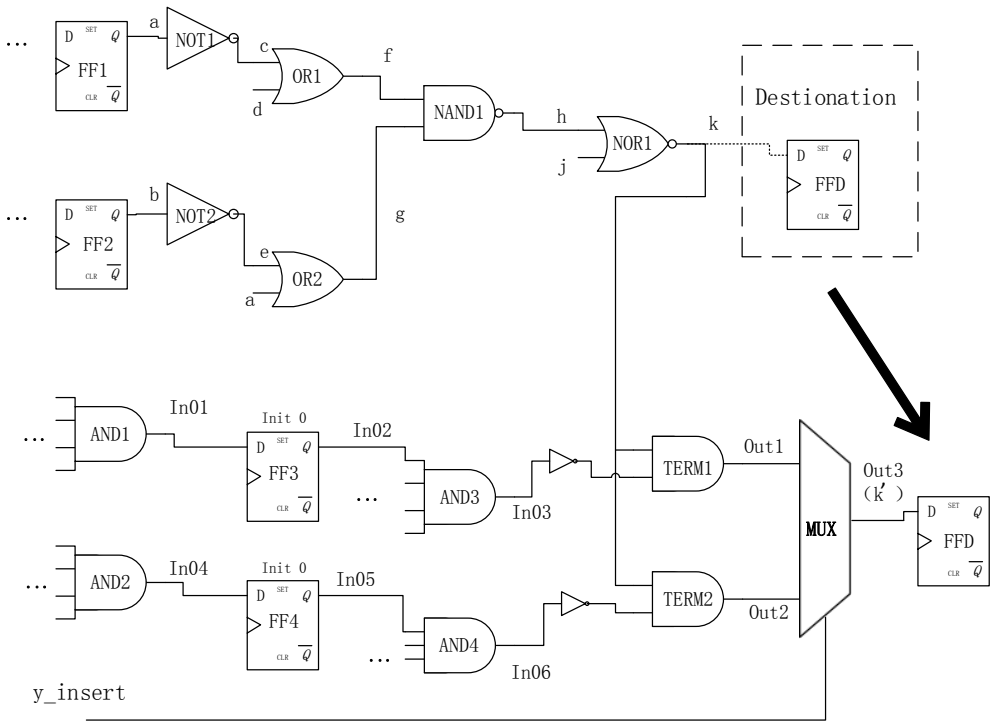

Figure 6.20: An example of AU faults for stuck-at fault.

6.7.3 Analysis of ATPG Untestable (AU) Fault

Analysis is performed when there is no test for a stuck-at fault at y_insert signal. Simulation results reveal that the fault at y_insert will be an AU fault. Figure 6.20 shows an example of generating an exclusive test for two path delay faults of the circuit. Path k connects the TERM1 and TERM2. Signal requirements are revealed in Figure 6.20.

Paths a-c-f-h-k and b-e-g-h-k are our two target paths. Path activation requires two vectors. Since it is a rising transition, the first vector needs to feed the input of FFD to be 0. And the second vector is able to ensure the input of FFD to be 1. The signal requirements through the target path are based on the kind of gates and transitions.

Table 6.4: Signal requirement for path a-c-f-h-k.

First vector	Value	Second vector	Value
k	0	k	1
h	1	j,h	0
f	0	f,g	1
c,d	0	c	1
a	1	a	0

Table 6.5: Signal requirement for path b-e-g-h-k.

First vector	Value	Second vector	Value
k	0	k	1
h	1	j,h	0
f	0	f,g	1
a,e	0	e	1
b	1	b	0

Table 6.4 and Table 6.5 show the signal requirements for the two paths. At the second vector, both of the signal requirements of the path could feed the input of FFD to be 1 without conflicts. The test pattern for a stuck-at fault on y_insert can not be found since the second vector can force both Out1 and Out2 to be 1.

Signal a of the first path is required to be 1 in order to make the destination flip-flop as 0. But at the same time, the signal a of second path needs to be 0 which has a conflict with the first path signal requirement. The two paths may be multiply testable [40], sometimes also referred to as functionally testable paths. Neither of the paths is testable alone. But they are testable when both paths are faulty. Thus, if a failure is observed then both paths should be considered faulty. They can be considered as equivalent, though not so in the real sense.

6.7.4 Diagnostic Test Pattern Generation

The previous sections provide the method for modeling a path delay fault as a single stuck-at fault. The main benefit of this model is that the conventional diagnostic tools for single stuck-at faults can be used to diagnose path delay faults.

1. Diagnostic coverage (DC) of tests provides a quantitative measure for their ability to distinguish any two faults.

2. Detection test determines the DC of the path delay faults. Implementation exclusive test on those path delay faults can generate more test patterns if necessary to enhance DC.

3. Signal requirements of modeling a path delay fault are based on the fault types and the related gate types through the path. Path definition files show the detailed information about the target path.

Chapter 7

Experiments Setup and Analysis of Results

Previous sections show the importance of fault diagnosis and how to construct a test model of exclusive test. In the first section of this chapter, we will briefly describe our experimental setup. In the second section, results of some circuits will be shown and evaluated.

7.1 Experimental Setup

In order to evaluate the test model, we will show the schematic views of test model. We use DFTadvisor to build the final the test model with a Verilog netlist, and the test model will be used to generate tests in Fastscan. Figure 7.1 shows the process of generating a test for a pair of faults

7.1.1 Construction of AND Gate of Test Model

A pair of slow-to-fall faults is shown in Figure 7.2. According to the method we previously discussed, we can obtain signal requirements for testing the two path delay faults.

Table 7.1 and Table 7.2 show the signal requirements, which will be expressed by an equation. For example, Table 7.1 will indicate the function as equation 7.1. Figure 7.3 shows the structure model in gate level. We can see these three signals y, g, h.

$$y = (\sim g)\&(\sim h). \tag{7.1}$$

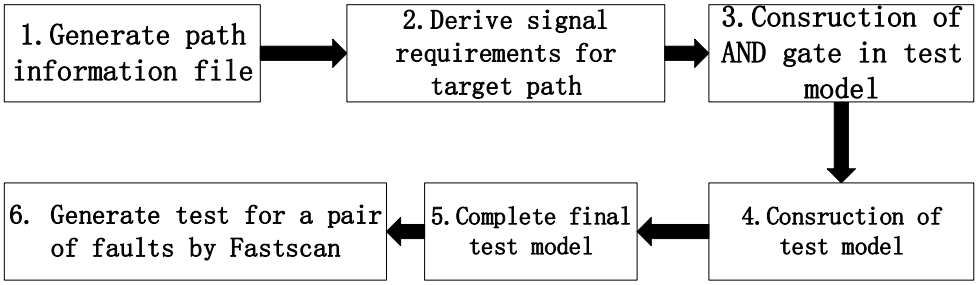

Figure 7.1: Test flows for a pair of path delay faults.

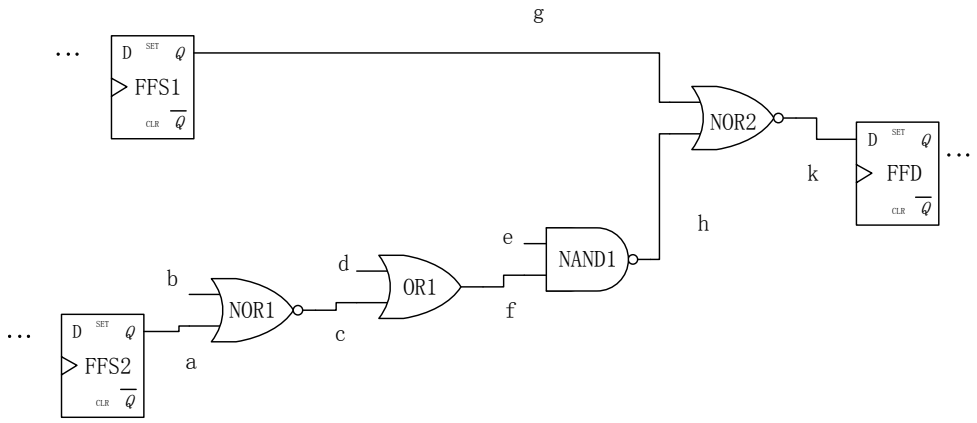

Figure 7.2: An example of a pair of slow-to-fall faults.

DFTadvisor can implement the equation by generating a structure model of it, as shown in Figure 7.4(signal y is the output of AND gate). Gate vin1 is AND gate. Gate vin2 and Gate vin4 are NOT gate.

7.1.2 Construction of Test Model

A test model of benchmark circuit s27 is shown in Figure 7.5. The output of the multiplexer will feed the destination flip-flop. Our test model will then be placed

Table 7.1: Signal requirement for k to be 1 in path g-k.

Signal	Value
k	1
g,h	0

Table 7.2: Signal requirement for k to be 0 in path g-k.

Signal	Value
k	0
h	1

between the output of last component of the path and the input of destination flip-flop. Therefore, signal z is connected to the output of the last component in the target path while outs will feed the input of the destination flip-flop.

7.1.3 Construction of Final ATPG Test Model

A y_insert(control signal of the multiplexer in test model) is added to the primary input of the circuit in order to generate a test for a stuck-at fault at y as shown in Figure 7.6. The test pattern should distinguish the pair of faults. Figure 7.6 shows the final test model and Fastscan will then try to generate a test for the fault in the circuit.

7.2 Results and Analysis

Results are shown in Table 7.3 and Table 7.4. Fastscan is used to generate tests for path delay faults. The target pair of path delay faults may be distinguished by generating a test for a single stuck-at fault. All normal flip-flops of the original circuit are scanned except the modeling flip-flop (MFF) of the test model. The initial state of

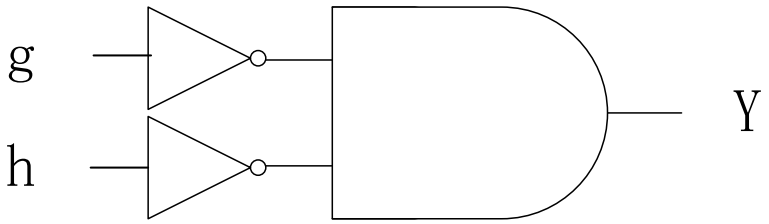

Figure 7.3: An example of AND gate.

unscanned MFF cannot detect fault by the first vector. Table 7.3 shows a comparison between DC of detection test and exclusive test.

The detection test has a DC shown in column 2. It is about 20% after the detection test. By applying our method to the undistinguished fault pairs, the DC greatly increased. As is shown in the third column, the method could raise DC of s27 circuit to 100%. This was the perfect case that each group has only one fault. We can see that the DC greatly increases after using our test model. It also raises the DC of benchmark circuit s298 from 23.9% to nearly 97%.

More details are shown in Table 7.4. The total faults of each circuit is shown in column 2. The fault coverage (FC) of path delay fault can be found at the third column. FC is less than 100%. The reason is that redundant or untestable fault is not identified and there were aborted ATPG runs.

The main purpose of our exclusive test is to generate tests that can distinguish a pair of faults with in the same fault group(constituted by detection test). Test that can distinguish a pair of faults will partition the groups. And then these two faults will be placed in separate groups. As a result, the number of groups increases in exclusive test. Column 4 of Table 7.4 shows the total number of groups after detection test while column 6 gives the number after exclusive test. It reveals that the number of total groups has changed a lot. The Diagnostic resolution (DR) decreases as the number of groups increases, and we can see the difference of column 5 and column 7. CPU time for each circuit is also listed at the last column, which is acceptable.

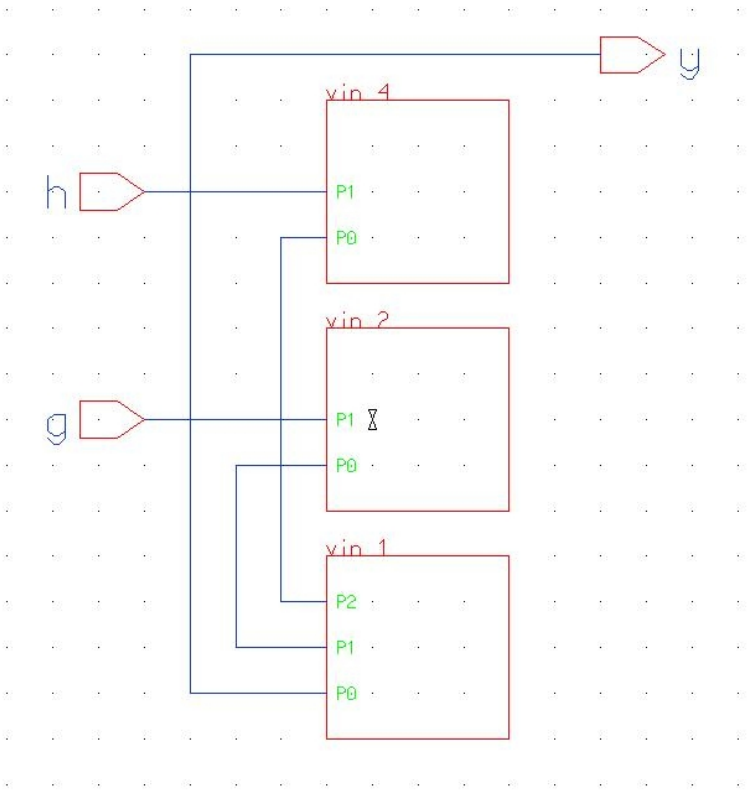

Figure 7.4: An example of AND gate in test model.

During the process of our test, if the exclusive test is not to be found, we enhance the ATPG abort limit which makes ATPG run longer to try to find a test. However, sometimes the test is still not to be found. Then we analyze the two faults in the circuit manually. As we previously described, the output response of two faults is same after we apply the second vector.

Generally speaking, this method can be also applied to other kinds of faults if their fault behavior can be mapped to stuck-at fault. The traditional ATPG tools to diagnose a stuck-at fault can be used for diagnosing these kinds of faults. By adding a few logics such as primitive gates, multiplexer and model flip-flops to the original circuit(only modified for ATPG), a test can be implemented by conventional ATPG tools. It greatly improve the diagnostic coverage of faults with little cost. This method is very easy to implement on diagnosing faults.

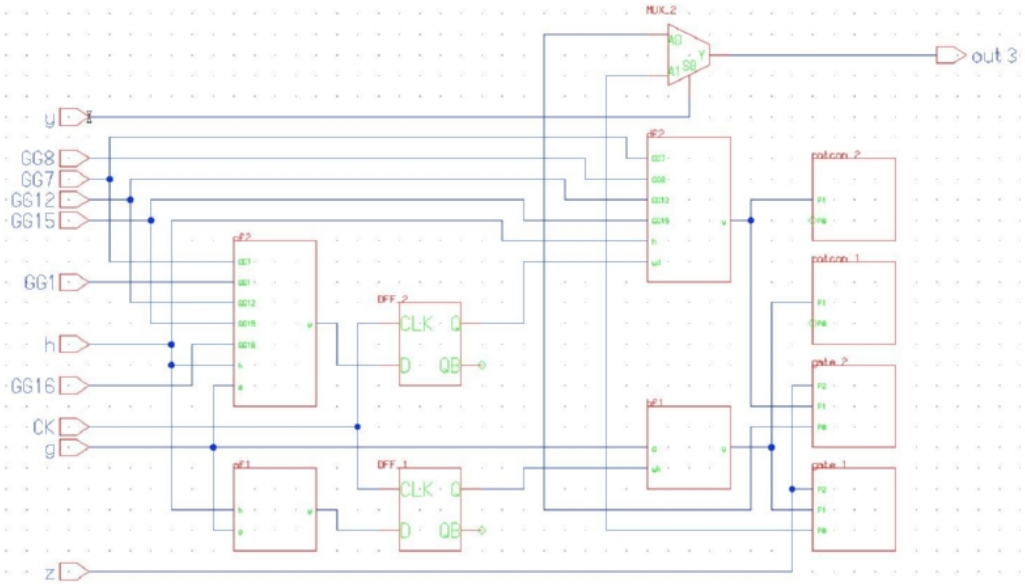

Figure 7.5: An example of AND gate in test model.

Table 7.3: Comparison between DC of detection test and exclusive test.

Circuit Name	DC Detection test %	DC Exclusive test %
s27	50.0	100.0
s298	23.9	96.5
s382	13.3	61.1
s400	12.0	78.8
s420	18.1	51.7
s444	19.53	80.0
s526	34.15	82.9
s838	10.4	31.2
s1423	5.3	62.9

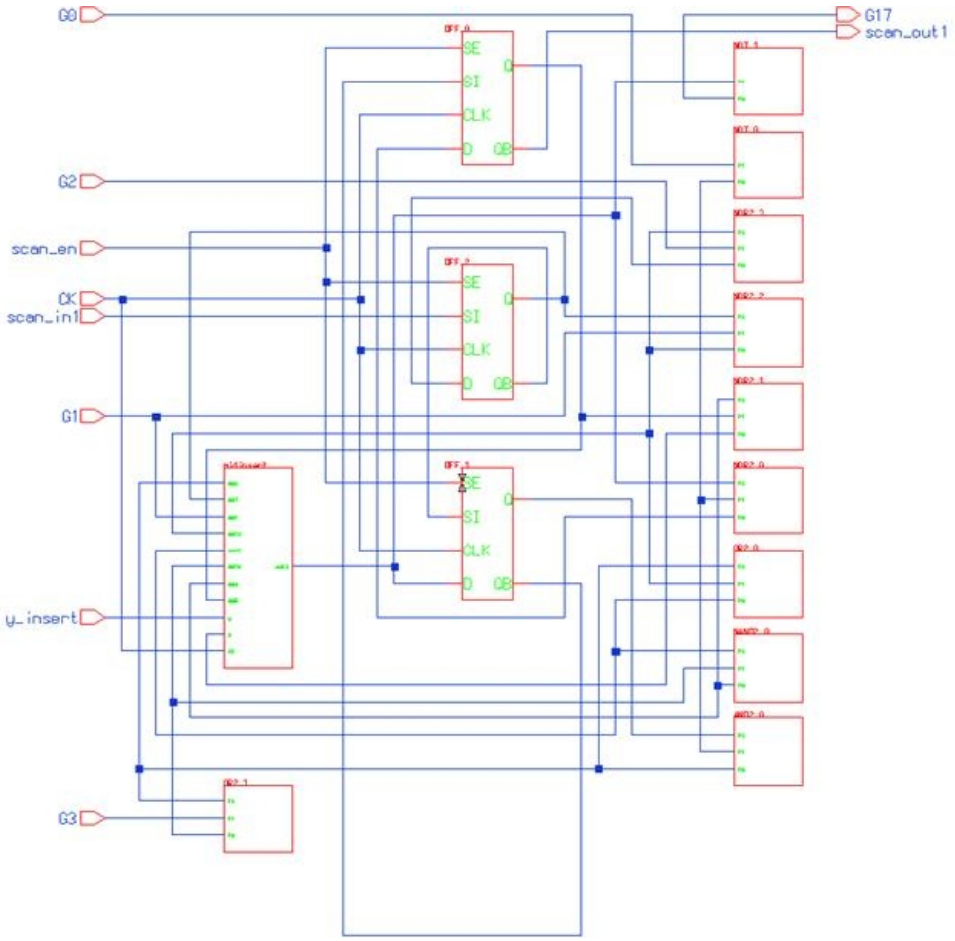

Figure 7.6: An example of circuit after inserting test model to s27.

Table 7.4: Comparison between detection test and exclusive test.

Circuit Name	Total Faults	FC %	Detection Test		Exclusive Test		
			Total Groups	DR	Total Groups	DR	CPU time (sec)
s27	4	100.00	2	2.00	4	1.00	5
s298	117	70.48	28	2.16	113	1.03	612
s382	314	74.41	42	7.47	192	1.63	4899
s400	350	72.61	42	8.33	274	1.27	943
s420	176	64.71	32	5.50	91	1.93	857
s444	215	31.99	42	5.11	172	1.25	260
s526	123	36.18	42	2.92	102	1.21	501
s838	612	57.95	64	9.56	421	3.20	4821
s1423	2759	44.77	148	18.64	1737	1.58	17897

Chapter 8

Conclusions, Developments and Future Work

From our experimental results and what we previously discussed, we learned that our proposed method is capable of successfully improving the diagnostic coverage metric (DC) by generating exclusive tests. The main purpose of diagnosis is to identify the location and nature of faults. However, some real defects cannot be mapped to fault models such as stuck-at faults. This chapter discusses the conclusions, recent developments an further work that needs to be done.

8.1 Conclusion

A proposed diagnosis method for path delay faults is applied to generate tests to distinguish a pair of faults. Our exclusive test on a pair of faults will try to increase the diagnostic coverage (DC), which is the ratio of the number of fault groups to the number of total faults. We can distinguish more fault pairs to constitute new groups. This can help increase DC.

At the beginning of the test, there is only one group of faults. Then a detection test is performed on these faults. DC is always less than 20% after detection test. In order to improve DC, exclusive test will start. If a test is generated for a pair of faults, this means that the two faults can be distinguished by this test. Therefore, these two faults will no longer stay in the same group, and the number of groups is increased.

In Chapter 4, we introduce a very efficient method for generating delay test by using stuck-at fault tools. In Chapter 5, we provide the Boolean analysis to give a theoretical proof of constructing a test model for generating exclusive tests. Only one

of two faults can be activated if a test is found. If these two faults produce the same output response after applying the second vector, a test of the two path delay faults cannot be found. Analysis of the two faults is shown in Chapter 6, which indicates that the two faults may not be equivalent faults. Then we explain how to construct an ATPG test model for a pair of path delay faults in Chapter 6. The method augments the ability of conventional ATPG to generate exclusive test by adding a few logics to the original circuit. Chapter 6 discuss the steps to construct a test model based on the transition and path information. Results of the improvement of DC and other related metrics are presented in Chapter 7.

In field of exclusive test generation, effective and efficient tools can be used to identify equivalent faults in order to have an adequate exclusive test. When we want to identify equivalent faults, the backtrack limit and test running time are required to be increased dramatically. Sometimes, ATPG is still not able to find the exact conclusions.

Generally speaking, our diagnostic method improves the DC of path delay faults. The problem of diagnosis of a pair of path delay faults is modeled as a single stuck-at fault. Equipped with the proposed technique, it's easy to construct an ATPG model to generate exclusive tests for path delay faults. Also, our diagnostic method can further improve the work [41]. Moreover, any fault whose fault behavior can be mapped to stuck-at faults has the potential to generate exclusive test with similar methods.

8.2 Developments

At present, there exist some works for path delay fault diagnosis. The path Scoring [41] method utilizes a reasoning-based diagnosis technique and stuck-at fault diagnosis results to improve the diagnosis resolutions of delay fault diagnosis.

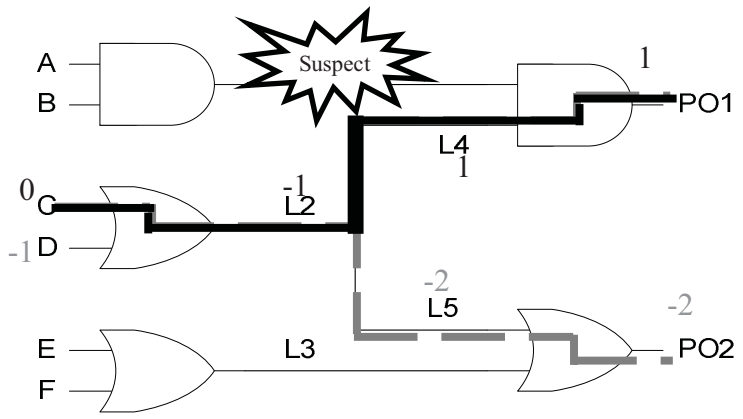

Figure 8.1: Calculation of path scores.

The diagnosis algorithm firstly extracts initial candidate faults by applying critical path tracing algorithm. Secondly, it calculates path scores with stuck-at fault diagnosis results. Finally, it evaluates candidate paths and reports path delay fault results.

Figure 8.1 shows how to calculate path scores and rank the suspect. If the primary output of the sensitized path is failed, the scores of all lines on the path are increased by 1. Similarly, if the primary output of the sensitized path is passed, the scores of all lines on the path are decreased by 1 [41]. Path 1: C-L2-L4-PO1 and Path 2: C-L2-L5-PO2. And we will calculate the total scores of all lines for each path. Therefore, the score of Path 1 is 1 while the score of Path 2 is -5. Since the score of path1 is higher than Path 2, it becomes the suspect. Finally, the ranked suspect lists are reported as the path delay fault diagnosis results.

8.3 Future Work

8.3.1 Application of Test Model to Non-Scan Circuit

Our work focused on application of our test model on scan circuits. Actually, the method [31] we previously used for detecting path delay faults with stuck-at faults tool also works in non-scan circuits. Therefore, if we make some changes to our method, it may be possible to be implemented in non-scan circuit.

8.3.2 ATPG Tools to Reduce ATPG untestable (AU) Faults

In this thesis, we use the path delay fault model to diagnose faults. However, the appearance of AU faults indicates that the ATPG has not determined whether there exists a test for detecting those faults. The ATPG tools may be sometimes neither capable of generating a test nor making sure that the fault is redundant. ATPG abort limit and test running time is greatly increased in order to detect more faults.

The presence of AU faults shows that the ATPG algorithms for detecting a fault still need improvement. Moreover, the diagnostic coverage of path delay fault may improve further if the ATPG calculation capability is enhanced.

8.3.3 Diagnosis of Real Defects

The main purpose of diagnosis is to find the nature and locations of real defects. The diagnosis method can use multiple fault models to activate and detect the multiple faults as much as possible. Our method uses traditional ATPG tools for stuck-at fault to diagnose path delay faults. If more fault models such as stuck-open faults, and bridge faults can be modeled as we did, more real defects can be detected. Net diagnosis technology [42] can diagnose a net fault, which often leads to a multiple fault scenario. The method can be used in many fault models. If the DC of more

fault models can be improved, net diagnosis technology and electrical test can narrow down the location of faults and find actual cause.

8.3.4 Overlap Component of Two Paths Blocked by Other Signal

In circuit, there may be an overlap component of the two target paths. This component may be blocked by the signal from another path(a path other than these two target paths). In this case, our test model will need more changes.

Bibliography

[1] "Fault diagnosis: Basic Concept.," http://www.pld.ttu.ee/diagnostika/theory/faultdiagnosis.html, accessed on 04/15/2015.

[2] "John Bardeen:Background, Inventor of Transistor.," http://en.wikipedia.org/wiki/John_Bardeen, accessed on 04/15/2015.

[3] "Walter Brattain: Background, Inventor of Transistor.," http://inventors.about.com/od/bstartinventors/p/Walter_Brattain.html, accessed on 04/17/2015.

[4] "Moore's Law: Background of Moore's Law.," http://www.mooreslaw.org , accessed on 05/17/2015.

[5] "Co-NP-complete.," http://en.wikipedia.org/wiki/Co-NP-complete, accessed on April 25,2015.

[6] "Main purpose of verification test.," http://www.csee.umbc.edu/ cpatel2/links/418/lectures/ chap1_lect00_testintro.pdf, accessed on 04/15/2015.

[7] Neil H. E. Weste and David Money Harris, "CMOS VLSI Design:A Circuits and Systems Perspective", Pearson Education, Boston, March, 2010.

[8] "background of acceptance testing.," http://en.wikipedia.org/wiki /Acceptance_testing, accessed on 05/15/2015.

[9] "Intro of fault model: Wikipedia.," http://en.wikipedia.org/wiki/ Fault_model, accessed on 05/20/2015.

[10] "Intro of stuck-open fault.," http://www.ece.unm.edu/ jimp/vlsi_test/slides/html /faults2.html, accessed on 04/29/2015.

[11] M. Renovell, P. HUC, and Y. Bertrand, "CMOS Bridging Fault Modeling", in *VLSf Test Symp.*, 1994, pp. 392–397.

[12] "Intro of stuck-at fault: Wikipedia.," http://en.wikipedia.org/wiki/Stuck-at_fault, accessed on 04/18/2015.

[13] M. L. Bushnell and V. D. Agrawal, "Essentials of Electronic Testing for Digital, Memory & Mixed-Signal VLSI Circuits", Boston, Springer, 2000.

[14] "Intro of IDDQ test: Wikipedia.," http://en.wikipedia.org/wiki/ Iddq_testing, accessed on 04/23/2015.

[15] "Intro of NP-complete problems: Wikipedia.," http://en.wikipedia.org/wiki/ NP-complete#NP-complete_problems, accessed on 04/14/2015.

[16] "Intro of Polynomial Time: mathworld.," http://mathworld.wolfram.com /PolynomialTime.html, accessed on 04/28/2015.

[17] "Intro of Turing machine: Stanford Encyclopedia.," http://plato.stanford.edu/entries/turing-machine/, accessed on 04/05/2015.

[18] "Definition and Examples of Karnaugh Maps: examples, definitions.," http://www.facstaff.bucknell.edu/mastascu/elessonshtml/Logic/Logic3.html, accessed on 04/15/2015.

[19] Laung-Terng Wang, Yao-Wen Chang, "Electronic Design Automation: Synthesis, Verification, and Test (Systems on Silicon)", Morgan Kaufmann, March 12, 2009.

[20] Laung-Terng Wang, Cheng-Wen Wu, and Xiaoqing Wen, "VLSI Test Principles and Architectures: Design for Testability", Morgan Kaufmann, July, 2006.

[21] "Description of undetected fault.," http://www.eng.auburn.edu/ strouce/class/elec6970/BISTc2.pdf, accessed on 05/18/2015.

[22] "Description of fault equivalence.," http://www.csee.umbc.edu/cpatel2/links/418/lectures/chap4_lect0_5_faults.pdf, accessed on 05/18/2015.

[23] "Description of fault collapsing.," http://www.ohio.edu/people/starzykj/network/Class/ee617/Handouts/dominance.pdf, accessed on 05/18/2015.

[24] M. J. Y. Willaims and J. B. Angell, "Enhancing Testability of Large-Scale Integrated Circuits via Test Points and Additional Logic", in *IEEE Trans. on Computers*, vol. C-22, no. 1, Jan 1973, pp. 46-60.

[25] "Description of UUT: examples, definitions.," http://www.vlsiencyclopedia.com/2012/12/testbench.html, accessed on 04/18/2015.

[26] Wang Z, Marek-Sadowska M, Tsai K-H and Rajski J, "Multiple fault diagnosis using n-detection tests", in *Proc. 21st international conference on computer design*, 2003, pp. 198-201.

[27] Gruning T, Mahlstedt U, Koopmeiners H, "DIATEST: a fast diagnostic test pattern generator for combinational circuits", in *Proc. IEEE international conf on computer-aided design*, 1991, pp. 194–197.

[28] Brian Chess, Tracy Larrabee, "Creating Small Fault Dictionaries", in *IEEE Trans Comput-Aided Des*, vol. 18, no. 3, March, 1999, pp. 346–356.

[29] "Description of pass-fail dictionary.," http://www.eng.auburn.edu/agrawvd/THESIS/ZHANG/dissertation_yu_2012.pdf, accessed on 04/18/2015.

[30] "Mentor Graphic Scan and ATPG Process Guide (DFTAdvisor, FastScan and FlexTest)", Mentor Graphics Corporation, August 2007.

[31] P. Agrawal, V. D. Agrawal, and S. C. Seth, "Generating Tests for Delay Faults in Nonscan Circuits", in *IEEE Design & Test of Computers*, vol. 10, March 1993, pp. 20–28.

[32] Mohammad Tehranipoor, Ke Peng, Krishnendu Chakrabarty,"Test and Diagnosis for Small-Delay Defects", Springer, 2011.

[33] V. D. Agrawal and K. K. Saluja, "Antitest, Exclusive Test, and Concurrent Test", *Unpublished Manuscript*, February 22, 2004, pp. 1–6.

[34] "Description of Shannon Expansion," http://ecse.bd.psu.edu/cse271/comprttb.pdf, accessed on 04/19/2015.

[35] P. Camurati, D. Medina, P. Prinetto, and M. S. Reorda, "A Diagnostic Test Pattern Generation Algorithm", in *Proc. of the IEEE Intl. Test Conf.*, 1990, pp. 52–58.

[36] P. G. Ryan, W. K. Fuchs, and I. Pomeranz, "Fault dictionary compression and equivalence class computation for sequential circuits", in *International Conf. on Computer-Aided Design*, Nov 1993, pp. 508–511.

[37] M. A. Gharaybeh, M. L. Bushnell, and V. D. Agrawal, "Classification and Test Generation for Path-Delay Faults Using Single Stuck-at Fault Tests", in *Journal of Electronic Testing: Theory and Applications*, vol. 11, no. 1, August 1997, pp. 55-67.

[38] "DFTAdvisor Reference Manual",Mentor Graphics Corporation, August 2007.

[39] "Description perl language.," https://www.perl.org/.

[40] Krstic, Angela and Kwang-Ting(Tim) Cheng,"Delay Fault Testing for VLSI Circuits", Springer, 1998.

[41] Y. Lim, J. Lee, and S. Kang, "Path Delay Fault Diagnosis Using Path Scoring", in *Proc. International SoC Design Conf.*, 2008, pp. 199–202.

[42] L. Zhao, "Net Diagnosis Using Stuck-at and Transition Fault Models", Master's thesis, Auburn University, ECE Department, December 2011.