Diagnostic Test Generation for Transition Delay Faults Using a Two-Timeframe ATPG Model

by

Xiaolu Shi

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 1, 2015

Keywords: Detection test, Exclusive test, Two-timeframe, Combinational ATPG

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

Determining the location and cause of a defect in a faulty circuit plays a vital role in VLSI testing. These are critical factors in boosting product quality and reducing manufacturing costs.

An automatic test generation system is presented which combines detection test and diagnostic test. It generates a diagnostic test for a modeled transition fault pair by making a simple modification of netlist. Using a conventional ATPG, a test can be generated for a specific stuck-at fault in the netlist model. The test is an exclusive test for the pair of transition delay faults. If an exclusive test exists, the fault pair can be distinguished in original full-scan circuit. This is followed by diagnostic fault simulation to identify undistinguished fault pairs and to update the diagnostic coverage (DC), which is a measure of the capability tests to diagnose faults. A 100% DC means that each fault can be distinguished from all other faults. The exclusive test generation continues until an adequate goal for DC is achieved.

This thesis presents a two-timeframe ATPG model for a full-scan sequential circuit.It inserts a few logic gates to the original circuit netlist for analysis purpose only, then expands to a purely two-timeframe combinational logic ATPG model.

We have enhanced the ability of automatic test generation system to distinguish transition delay fault pairs. Thus, a transition delay fault pair can be either distinguished or proved equivalent, i.e., two faults have the exact same output responses at all nodes.

Compared to sequential logic, combinational logic is more effective for redundant fault identification. This property is exploited in our exclusive test generation system. The entire system was implemented using Python programming language and commercially available CAD programs. The proposed method is practical as the modeling, relatively simple and with improved DC. This is observed from experimental results on ISCAS'89 sequential benchmark circuits.

Acknowledgments

Among all the people who help me through this work, first and foremost, I owe a deep appreciation to my advisor Dr. Vishwani Agrawal. He is always supportive and encouraging. Throughout my thesis research, when I encountered tough, difficult and out of ideal circumstances, he showed extraordinary patience and provided constant guidance to my work. Without his remarkable dedication, I may never have completed my research this smoothly.

I would like to express my gratitude to Dr. Adit Singh and Dr. Victor Nelson to be my committee members and their enlightening courses. Moreover, they always guided and supported me during my research.

Among many colleagues and friends who helped me solve problems encountered in this work through their suggestions and build confidence are Chao, Zeshi, Jialin, Kunpeng, Haiqiao, Tenchi and many more. They have believed in my work and no one can imagine how strong this help is.

Last but not least, I wish to express my appreciation to my parents for their deep and unconditional love. They are always my strongest backup and steady inspiration.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

As the logic density during VLSI fabrication process advances rapidly, the defect related issues emerge increasingly. If a failing chip is not identified and eliminated, the manufacturing process will waste a large amount of money and effort. To reduce the number of faulty chips and obtain higher yield, it is necessary to make certain amount of investment on chip testing and diagnosis so that defective chips can be identified and defect mechanisms can be eliminated from the manufacturing process.

Various fault models have been devised to model real defects on chips. The single stuck-at fault model is referred to as the classical fault [15, 26]. Non-classical faults include other fault models such as transition delay fault, gate delay fault and bridging fault, which will be discussed in Chapter 2.

A vital aspect of guaranteeing high quality of chip fabrication is effective fault analysis, a process of detecting the failure, figuring out the specific reason of physical defect and eventually fixing the possible defect producing causes. As a first step in fault analysis, fault diagnosis assumes an important role in evaluating the failure location on chip and improving the yield. In addition, a better fault diagnosis algorithm as well as an organized test generation system have decisive effect on the diagnosis efficiency and achievement. Timing issue is always a main reason for high cost during VLSI testing. The use of fault dropping based diagnostic algorithm and combinational stuck-at fault automatic test pattern generation (ATPG) [1] in the diagnosis of timing faults is the main contribution of this thesis.

1.1 Problem Statement and Contribution

The goals achieved in this work are:

- Construct an effective two-timeframe combinational ATPG model
- Generate necessary test vectors to distinguish fault pairs
- Improve diagnostic coverage of transition delay faults

An original contribution of this work is in constructing a two-timeframe expansion combinational ATPG model for detecting transition faults in a full-scan circuit under test (CUT). The generated tests can distinguish targeted transition delay fault pairs. An automatic test generation system is developed using this combinational ATPG model and available combinational stuck-at-fault test generation and transition fault simulation.

For a transition delay fault pair, the ATPG system either generates an exclusive test which distinguishes the two faults (two faults have different output responses at any observable signal line) or proves them equivalent (two faults are always detected at same node). Two-timeframe ATPG model works in combinational ATPG mode which is easier for identifying redundant faults or generating test vectors. With more fault pairs successfully targeted, expectedly the diagnostic coverage (DC) of transition delay faults will be surely improved.

1.2 Organization of Thesis

Fundamentals of VLSI testing and fault diagnosis are presented in Chapter 2, including digital circuit type, various fault models, ATPG principle, fault simulation and design for testability (DFT) techniques. In Chapter 3, we introduce the basic background for diagnosis and preview contributions on diagnostic test generation system. Chapter 4 discusses the ATPG model construction for exclusive test generation under both launch off capture (LOC) and launch off shift (LOS), and the diagnostic test procedure is presented in detail. Experimental results of this work and comparison to previous results are shown in Chapter 5. Chapter 6 gives the conclusion from this research and discusses future work targeting the LOS test mode that was not implemented in the present work.

Chapter 2

Background and Overview of Fault Diagnosis

This chapter will discuss background of VLSI testing and fault diagnosis. It starts with introduction of digital circuit types which are classified as combinational circuit and sequential circuit. It will present several fault models which constructed to represent the real defects on chips. Then ATPG as important tool and fault simulation as crucial procedure in testing will be explained. Finally the chapter will describe DTF techniques for scan based test to generate tests as easily as for combinational logic.

2.1 VLSI Testing

In VlSI dectection test, we need to figure out whether a chip is good or faulty. A detection test procedure is shown in Figure 2.1. We apply test vectors to circuit under test (CUT) and get outputs, then compare the outputs with the expected output responses. If they are match, which means the CUT is a good circuit; if they are mismatch, that is to say, there exists defects on this chip.

2.2 Digital Circuit Type

In general, digital circuits can be divided into two categories, combinational circuits and sequential circuits. A combinational circuit, as shown in Figure 2.2, contains only combinational logic gate, and output responses can be obtained immediately after applying input vectors. It therefore is a time-independent circuit. Some fundamental combinational circuit examples includes adder, multiplier, multiplexer and encoder.

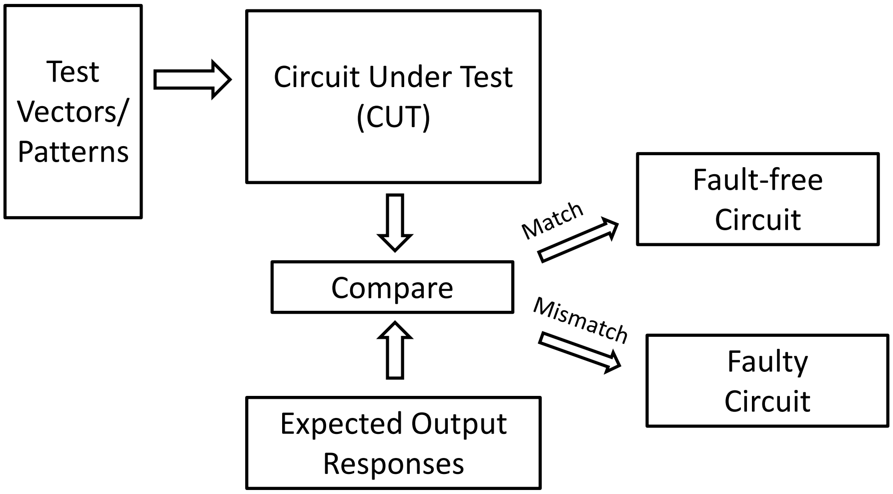


Figure 2.1: VLSI detection test procedure.

A Sequential circuit contains memory elements and clock signals. The output not only results from the input value, but also the stored value from previous states. A sequential circuit in which D filp-flops are memory elements is shown in Figure 2.3.

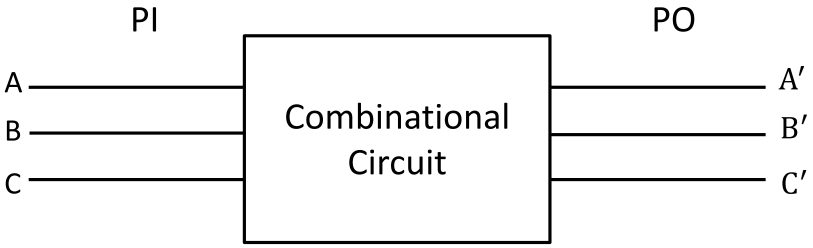


Figure 2.2: A combinational circuit.

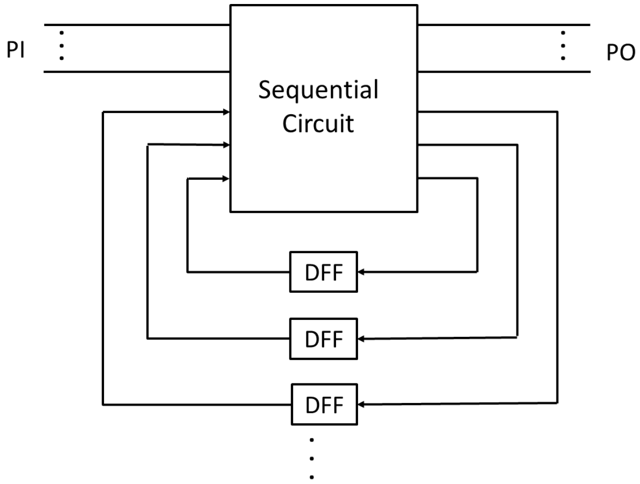


Figure 2.3: A sequential circuit.

2.3 Fault Model

During the testing and fault diagnosis processes, we need to represent real chip defects as corresponding logical faults, meaning construct a fault model which represents the characteristics of a target circuit. The popular fault models are divided into transistor level faults, gate-level faults, delay faults and so on.

2.3.1 Stuck-at Fault Model

The most widely used gate-level faults model is stuck-at fault model, which is a signal line stuck at a logic '0' or '1', referred to stuck-at 0 and stuck-at 1 respectively, and assuming the value cannot be changed. Hence it is a logic fault model without concerning timing issue. The stuck-at fault generally represents a physical short defect (a signal line short to GND shown as stuck-at 0, to VDD shown as stuck-at 1). For a circuit with 'N' signal lines, the number of possible stuck-at faults is '2N'.

As shown in Figure 2.4, the primary input value of the NAND gate is '11'; the output value is supposed to be '0'. However, because of the stuck-at 0 fault in input line 1, the actual input of this NAND gate is stuck at '01' and the output response

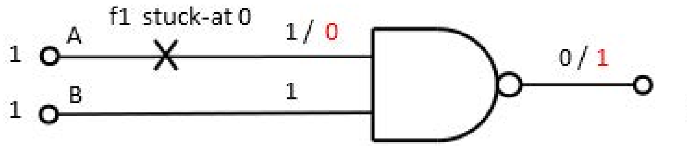


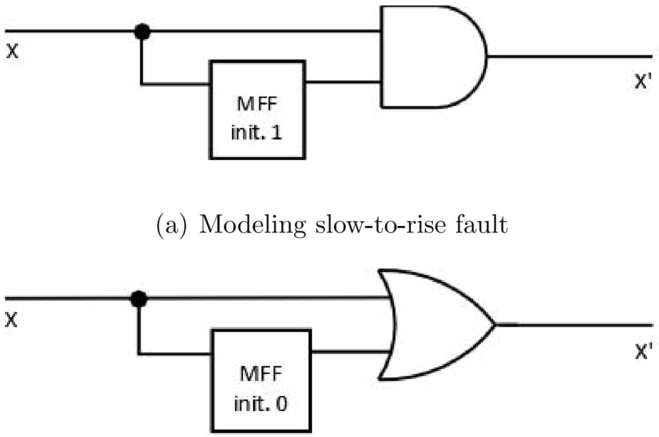
Figure 2.4: a NAND gate stuck-at fault illustration.

becomes '1'. To sensitize this stuck-at 0 fault in input line 1, we need to apply '1' at this line and non-effect signal '1' on the other line.

2.3.2 Transition Delay Fault Model

Timing issue which required high clock frequency often emerges due to manufacturing defects and VLSI design errors. Delay fault models are generally divided into two categories, path delay fault and transition delay fault. A path delay fault model spreads the excessive delay along a circuit path from flip-flop to flip-flop (latch to latch). A transition delay fault is assumed to occur at a single gate input or output in the circuit that has excessive delay. Similar to a stuck-at fault fixed at '0' or '1', a transition delay fault is shown as 'slow-to-rise' and 'slow-to-fall' in Figure 2.5.

To detect both delay faults, a pair of sequence test vector needs to be applied, the first vector set the initial state on the target signal line and the second one sensitizes the specific fault and propagates the effect to a primary output. In Figure 2.5(a), the model consists of a modeling flip-flop (MFF) which initialized to 1 and an AND gate, when we apply '00', '11' and '10', the model will get a corresponding output. However, when '01' is applied, the output response is '00', a slow-to-rise fault propagates from x to x'. Similarly, the model in Figure 2.5(b) where the MFF initialized to 0, when we apply '10', the output is '11' which is supposed to be '10', thus a slow-to-fall fault is modeled.



(a) Modeling slow-to-rise fault

(b) Modeling slow-to-fall fault

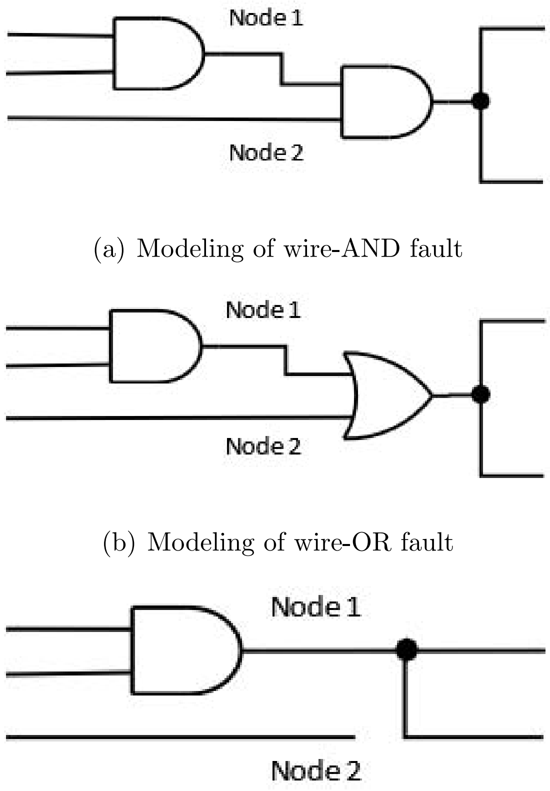
Figure 2.5: Modeling transition delay faults [44].

2.3.3 Bridging Fault Model

During the process of VLSI fabrication, defects in an interconnection area, like two segments that are too close to each other, may result in logic errors. Bridging between a signal line to VDD or ground is analogous to a stuck-at fault model. The common bridging fault models as shown in Figure 2.6 are: wire-AND, wire-OR and dominant [21]. In the wire-AND model, two bridged nodes are logic 0 dominant, which means either of two nodes being '0' will lead to '0' on the other node. Similarly, the wire-OR model is a 1-dominant bridging fault, the value on either of two shorted wire is '1' will generate '1' on both lines. In the dominant bridging fault, the signal value on one line is driven by the value on other line with stronger capability.

2.3.4 Other Fault Models

There is a wide variety of other fault models, for example, transistor level stuck-at fault [39] can be classified as stuck-on (stuck-short) and stuck-off (stuck-open) which can be modeled as a conventional stuck-at fault. A stuck-on fault is a transistor always conducting, and the performance of a stuck-off fault is that it never conduct. An IDDQ fault [19] is that the circuit may pass logical testing but its power supply current



(a) Modeling of wire-AND fault

(b) Modeling of wire-OR fault

(c) Modeling of dominant fault, node1 dominate node 2

Figure 2.6: Modeling of bridging fault.

is larger than the expected limit current in a static condition, and IDDQ testing can be utilized in transistor level stuck-at fault testing. A gate-delay fault [25, 30, 31, 32] refers to the issue that the delay through input to output of a logic gate above a certain amount may result in performance degradation. Unlike transition delay fault is an independent, instant-effect fault, gate-delay fault model is a quantitative model, thus it is essential that one consider the delay fault size when evaluating delay fault test capability.

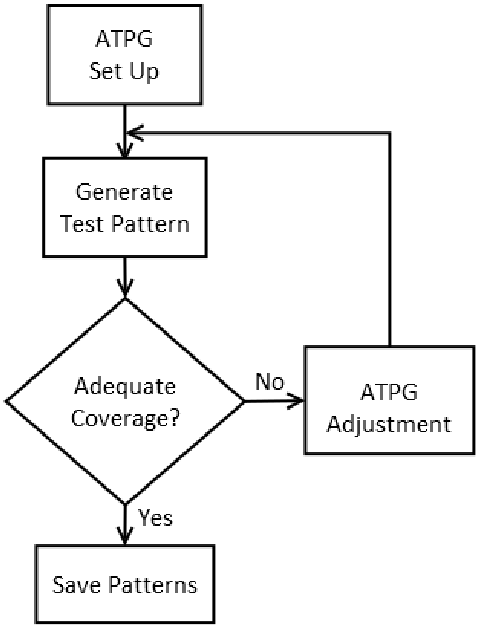


Figure 2.7: Flowchart of procedure of ATPG.

2.4 Automatic Test Pattern Generation (ATPG)

2.4.1 ATPG procedure

Automatic Test Pattern Generation (ATPG) is a technology used to generate test patterns to achieve adequate coverage, a flowchart of it shown in Figure 2.7. Various ATPG test generation algorithms [1, 17] have been proposed, such as D algorithm, FAN algorithm, PODEM, WASP and Pseudorandom test generation, the core thought of these algorithms is to distinguish a good circuit and a faulty circuit caused by physical failures.

An ATPG at first applies ”random patterns” to detect as many faults as possible with reasonably small effort. Then, it aims at the remaining undetected faults to generate particular test patterns. To detect a fault, the ATPG inserts the fault at a node in the circuit under test (CUT). Two procedures applied to the targeted fault are fault activation and fault propagation. Fault activation sets a value, which is

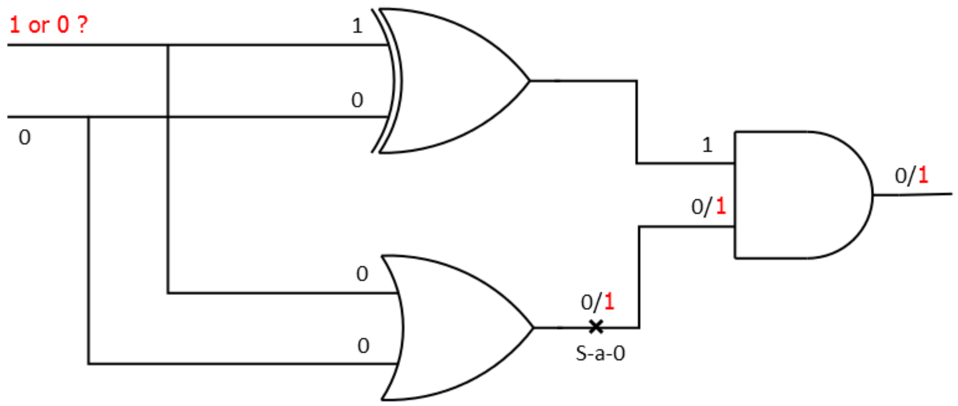


Figure 2.8: An example of a redundant fault.

opposite to the value caused by the failure, at the fault site, and the state of the fault site termed fault effect is propagated through logic gates from fault site to a primary output. To sensitize a fault propagation path, ATPG will keep non-controlling values on other inputs of each gate on the path. For example, if the fault propagates through an OR gate, the non-controlling value '0' should be held on other inputs.

When observing the fault effect on the primary output and comparing the output of test patterns of a fault-free circuit and CUT, if the output of CUT is different from the value expected, the fault is said to be detected by the test pattern.

2.4.2 Failure generation of ATPG

ATPG does not works perfect every time, it occurs failure generation for certain kinds of faults. If ATPG can not generate a pattern for a fault, it is usually because of are two possible reason. One is the circuit contain redundant logic, which means after applying all kinds of test vectors, the fault is still can not be activated and the output response can not be changed, this kind of fault is a redundant fault [20]. Figure 2.8 shows an example of redundant fault.

In Figure 2.8, a stuck-at 1 fault at the output of OR gate, to detect this fault, both of the input of OR gate should be 0, and the first input of AND gate should be 1. For the XOR gate, the second input is 0, so the first input should be 1, there is a conflict at the first input of XOR gate. No test vector exits can meet this condition, consequently stuck-at 1 fault at the output of OR gate is a redundant fault.

Another possible reason is ATPG indeed fail to find a test pattern which is supposed to exist for a particular fault is due to its inner mechanism, this kind of fault is called "ATPG untestable fault." ATPG cannot prove redundant, but is also unable to generate a test. After changing the instruction constraints and abort limitations, this kinds of faults may be detected.

2.5 Fault Simulation

Fault simulation is the process that figures out which are detected and undetected faults, and determines the fault coverage for the test pattern set generated by ATPG. Thus, fault simulation is a reverse process of ATPG to a certain extent [10].

A fault simulator applies all the generated test patterns to a target fault, or a fault set, then simulate faults, and observe output responses. If output of the faulty circuit is different from the expected value with the good circuit, we get a detected fault. However a fault is undetected when value of POs are identical to expected responses. The fault simulator will repeat the steps until all faults are targeted. Fault coverage then will be calculated with detected fault number and total modeled faults in fault list. A measurement of fault simulator efficiency will be introduced detailed in Chapter 3.

2.6 Design for Testability (DFT)

In the testing process, we need to compare the response of a good circuit and those of the CUT. However, most digital circuits on the market are sequential circuits,

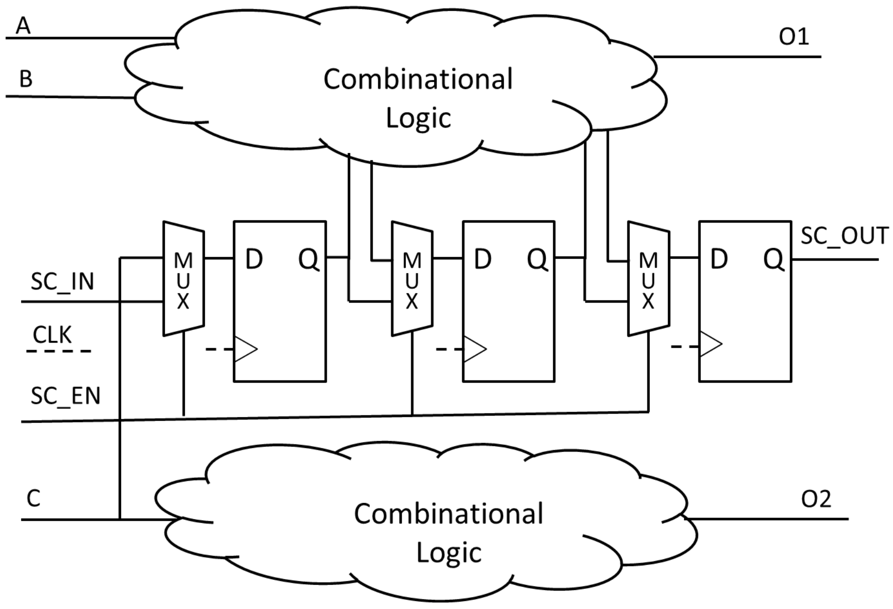


Figure 2.9: An illustration of DFT structure.

in which internal signals and states are hard to observe and control. In consequence, it is hard to generate tests. Testing can be improved by scan based test, in which memory elements (D flip-flops, latchs) consist of one or more scan chains to hold the previous value and state. A DFT structure is shown in Figure 2.9.

From Figure 2.9 we can see that in the scan based design, a multiplexer and a D flip-flop form a 'scan' flip-flop, and all scan flip-flops in one scan chain constitute a shift register. The scan-enable (SC_ EN) signal will determine whether the system is under normal mode or scan mode, When SC_ EN = 0, the circuit is under normal mode and the system operate under scan mode when SC_ EN = 1; the scan-in (SC_ IN) data signal and original data signal from combinational circuit are chosen by a multiplexer according to different system mode. During the process of scan test, test vectors can be shifted in or out through scan chains while other inputs are still given by primary inputs (PIs). In this way every input can be controlled bit by bit and outputs can be observed more clearly. Consequently, the sequential circuit with scan based design has better controllability and observability. The concern with this scan based design it may cost a lot of time on shift in and shift out vectors bit by bit.

Built-in self-test (BIST) is another DFT technique which allows circuits to test themselves. BIST [2] is capable of generating test vectors inside the CUT and verifying the circuit internal functionality. Therefore, BIST has high reliability, and lower complexity.

Chapter 3

Concepts and Review of Previous Work on Fault Diagnosis

In detection test, we can only figure out whether the chip is defect or not. To improve product quality, we still need to know what and where the chip goes wrong. In this case, fault diagnosis assumes a vital role in locating possible defects in the circuit and improving yields.

This chapter presents a brief review of fundamental concepts of fault diagnosis, contributions of previous work, mainly focusing on diagnostic metrics definition, and a detailed description of an automatic test generation system which can generate both detection test and also exclusive tests to distinguish fault pairs, thus give a better diagnostic metric.

3.1 Fault Diagnostic Algorithm

Generally, fault diagnosis algorithms can be divided into two categories: effect-cause and cause-effect [7].

In effect-cause algorithm [7, 13], depending on the set of actual failing response, effect-cause diagnosis first traces back the error propagation path from failing POs by multiple methods. Then, fault simulation on suspected faults, compares the failing response and rank faults, eliminate the low ranking suspects, thus narrow down the failure location. These types of diagnosis methods have less memory storage and take advantage of higher accuracy and better resolution [40].

Cause-effect algorithm [18, 22, 42] is based on a database of simulated data for all model faults, which is called “a dictionary” of “fault signatures”. The failing device response is compared with stored data to determine which fault might be the reason of the failure. Since this algorithm uses a pre-calculated database, one concern is the requirement of large amount of memory and huge dictionary construction time, especially when the circuit becomes large.

3.2 Fault Dictionary

Fault dictionary is the pre-simulated test responses data which records the tests activation and faults propagation information. Fault dictionary is constructed for a theoretical good chip, and is used for silicon debug during the process of diagnostic fault simulation [16, 46] on a failing chip.

Two common forms of dictionaries are *full response dictionary* and *pass/fail dictionary*. A full response dictionary is a complete record which contains failing pattern indexes and all the failing output responses for each fault-vector pair. A pass-fail dictionary only stores the single pass or failing pattern index for each fault-vector pair The problem with pass-fail dictionaries is that without considering all of the failing outputs, it is hard to distinguish fault candidates that fail the same set of tests, thus leading to a lower resolution diagnosis [29].

Table 3.1 shows a full response dictionary with a single fault and three primary outputs and Table 3.2 shows a corresponding index pass-fail dictionary. If a test pattern fails, then place a '1', on the contrary, '0' is placed where the pattern passes the test. In the full response dictionary, we can easily locate each fault fails on specific primary output.

In the table 3.2, the indexes in a row provide a signature for the corresponding fault. We can see fault f1 passes 1 test vector and fails the other 3 test vectors, thus having a signature of '0111'. However, it is common that two faults may have the same signature in a Pass/Fail dictionary, such as f2 and f3, they both have the signature '1101', which obviously cannot be distinguished. If a circuit under diagnosis (CUD) passes f1, f4 and f5, but fails f2 (or f3), we cannot identify whether the failure

Table 3.1: Full response dictionary.

Faults	Outputs Response			
	t1	t2	t3	t4
f1	000	010	111	100
f2	110	101	000	001
f3	001	010	000	001
f4	000	101	100	000
f5	001	000	111	010

Table 3.2: Pass/fail dictionary.

Faults	Outputs Response			
	t1	t2	t3	t4
f1	0	1	1	1
f2	1	1	0	1
f3	1	1	0	1
f4	0	1	1	0
f5	1	0	1	1

is solely caused by f2 or f3. However, they can be clearly distinguished in the full response dictionary.

3.3 Diagnostic Metrics

3.3.1 Fault Coverage

Fault diagnosis is the combination of fault detection and fault isolation, which means at first figure out whether the fault exits then locate the fault in the circuit. To better quantify the diagnostic capability, we need introduce some metrics to estimate the test effectiveness. Fault coverage (FC) is a quantitative measure of the effectiveness of a detection test. It is the ratio of the number of faults detected by a test vector set and the total number of faults in fault list. FC is defined as:

$$Fault\,Coverage = \frac{Number\,of\,detected\,faults}{Total\,number\,of\,faults} \tag{3.1}$$

If the fault coverage is 0, which means there are no faults can be detected, and when FC is close to 100%, it means that almost all faults are detected. However, in real applications, an adequate fault coverage detection test may not ensure sufficient diagnostic performance.

3.3.2 Diagnostic Coverage

Similar to FC in detection test, diagnostic coverage (DC) metric [45] is defined to evaluate the effectiveness of a given set for fault diagnosis. It is given as:

$$Diagnostic\,coverage = \frac{Total\,number\,of\,detected\,fault\,groups}{Total\,number\,of\,faults} \tag{3.2}$$

A distinguished fault [12, 48] means that there is at least one different output response at a scan flip-flop or primary output when applying a test vector. In contrast to a distinguished fault, equivalent faults [33, 43] have the exactly same output response at any signal lines for a input test vector, that is to say they cannot be distinguished by this test vector.

Zhang and Agrawal [45] propose that, we first group all faults which are not distinguished from each other in a single group for a given set of vectors. Then if we can produce a newly distinguishable fault subset after applying a vector, a new group will be constructed which consists of the new fault subset, and the other undetected faults still remains in the other group. In this way fault subsets in one group have exactly same fault signature.

A fault group contains two or more equivalent faults at very beginning, if all the faults are targeted, and distinguished from each other, one updated fault group will only contain one fault, which means DC is equal to 1. If DC is 0, it indicates the initial before any diagnostic test; and more faults pair can be distinguished, the higher diagnostic coverage will be achieved. In this thesis work, we use DC to measure the capability of diagnosis.

3.3.3 Other Diagnostic Metrics

Diagnostic Resolution (DR) is also a measurement of fault diagnosis effectiveness, which define as:

$$Diagnostic\,resolution = \frac{Total\,number\,of\,faults}{Total\,number\,of\,signatures} \tag{3.3}$$

We can see DR defines the average number of per fault groups (each fault group has its unique signature, different fault groups have various fault signiture), which actually is a reciprocal of DC. When DR is equal to 1, it is the perfect circumstance which every fault has its own distinct signature. Similarly, fault pair coverage (FPC) is given as:

$$Fault\,pair\,coverage = \frac{Total\,number\,of\,fault\,pairs\,distinguished}{Total\,number\,of\,fault\,pairs} \tag{3.4}$$

If every fault pair can be distinguished, the fault pair coverage will be 1. All these diagnostic metrics introduced above describe the diagnostic capability from different aspects.

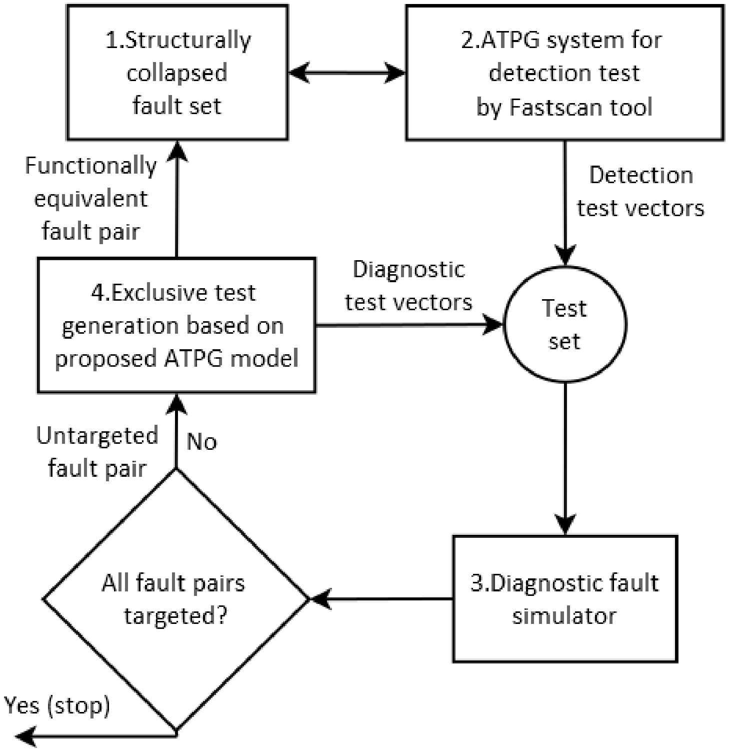

1.Structurally collapsed fault set
2.ATPG system for detection test by Fastscan tool
Functionally equivalent fault pair
Detection test vectors
4.Exclusive test generation based on proposed ATPG model
Diagnostic test vectors
Test set
Untargeted fault pair
No
All fault pairs targeted?
3.Diagnostic fault simulator
Yes (stop)

Figure 3.1: Flowchart of exclusive test generation.

3.4 Automatic Exclusive Test Generation System

Exclusive test [8] is defined as a test vector that can detect only one fault of the targeted fault pair. Two faults are independent [9] if no test vector exists to detect both of them simultaneously. Authors in [45] proposed an exclusive test generation system which combines the detection test and diagnostic test in an automatic system. The flowchart is shown in Figure 3.1.

In the flowchart, blocks 1 and 2 are conventional ATPG system for detection test by Mentor Graphics Fastscan tools to calculate the fault coverage, total number of detection test vectors and construct fault dictionary. Based on the test set generated by detection test, block 3 is a diagnostic fault simulator. At first determine detected faults for an input test vector, group faults with the same fault signature, and then calculate DC. After identifying undiagnosed fault groups, if all faults in groups are targeted, the diagnosis will stop. If not, exclusive test in block 4 will continue work on untargeted fault pairs. There are two possible results for exclusive test: first, exclusive test exits. This generated test vector will be combined to the previous test vector set. Then the diagnostic fault simulator will recalculate the new DC, update undistinguished group and start a new diagnostic cycle. Second, there is no exclusive test can be generated. When the fault pair is redundant faults, the fault pair is functionally equivalent and no test exists to distinguish them, and then they will be collapsed into one fault.

During the diagnostic fault simulation process, we apply fault dropping [26, 27, 34] when a fault is targeted in a single fault group, deleting this fault from previous fault list. Fault dropping decreases memory storage and time consumption, thus decreasing the complexity of fault simulation. Besides, if no exclusive test vector can be generated for a fault pair and the two faults are redundant, we delete one of them from the targeted fault group, which will improve DC but has no effect on FC.

Chapter 4

Diagnostic Test Generation for Transition Delay Faults

In this chapter we will review an existing single timeframe ATPG model for exclusive test generation using sequential ATPG. The model inserts a modeling stuck-at fault for targeted transition delay fault pair. Then, a new two- timeframe expansion ATPG model will be presented using a pure combinational ATPG procedure both for launch-off-capture (LOC) and launch-off-shift (LOS) test generation mode.

During scan test, for a vector pair generation, the second vector totally depends on the output of the first vector under LOC mode, which suffers from low controllability and observability of inner signals. However, the alternative LOS test generation mode needs a high frequency scan capture clock, which is hard to realize and will increases the manufacturing cost.

An effective diagnostic test generation procedure is presented, and an illustration of two-time frame expansion method for exclusive test of s27 sequential benchmark circuit is given. The proposed procedure can achieve higher diagnostic coverage (DC) with relatively simple operations.

4.1 XOR gate exclusive test model

A test on stuck-at-0 at the output of a XOR gate is an exclusive test of two faults in two copies of circuits under test (CUT). Because of the characteristics of a XOR gate, if a test on stuck-at-0 at the output exits, which means the two inputs should be different from each other, thus ATPG tool can generate exclusive test for two faults. The model is shown in Figure 4.1. We use an ATPG model developed

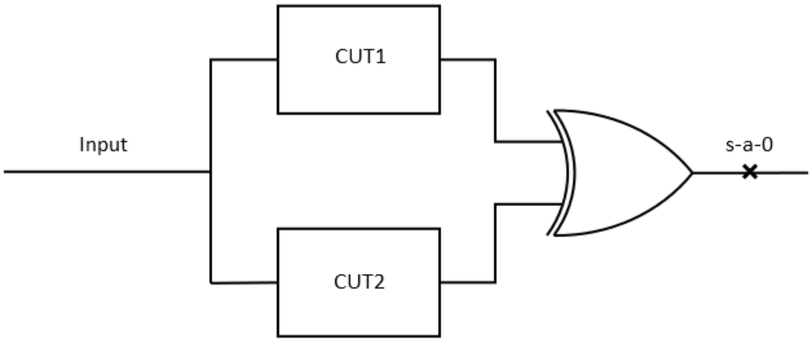


Figure 4.1: An XOR gate exclusive test model.

from this XOR gate model, it substitutes the XOR gate for a multiplexer which will be presented in the following sections.

4.2 Single Time Frame ATPG Model

When a diagnostic metric value is not adequate, we need to perform additional exclusive tests. To generate exclusive tests, we use the ATPG-model based methods. An ATPG model is

constructed to model a transition delay fault, as shown is Figure 4.2. In this model X1 represents a fault free circuit, X2 consists of an AND gate and a modeling flip-flop (MFF), which initially set to 1. When sequentially apply ‘01’ to input, signal line X2 will get a response ‘00’, which is a slow-to-rise fault as we discussed in Chapter 2. In this case, a stuck-at-0 or stuck-at-1 detection test on multiplexer selection signal y will generate a pair of vectors which are different between fault free signal X1 and transition delay fault signal X2, and thus models a slow-to-rise fault.

An ATPG-model based exclusive test generation model [48], which is inserted into circuit under test (CUT) to assume as a stuck-at-fault. Without varying the

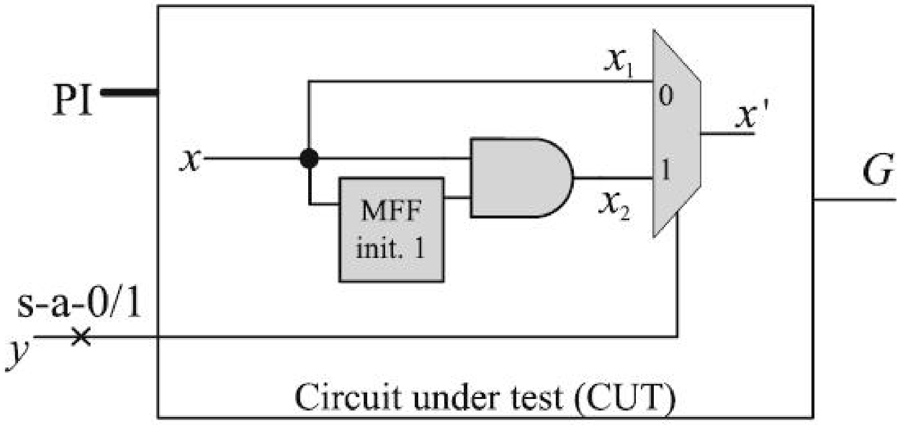


Figure 4.2: An ATPG model for slow-to-rise fault [47].

original function of the CUT, the purpose of modification in the netlist is only for facilitating the determination of an exclusive test for a targeted fault pairs.

A single time frame ATPG model [48] for exclusive test generation, as shown in figure 4.2, consists of only one single copy of the CUT. The bottom part of this model, as we discussed above, formed with a MFF and an AND gate, indicates a similar way that the test on y s-a-0/1 is an exclusive test for fault free and slow-to-rise TDF. In the upper part of this model, a slow-to-fall TDF can be modeled by a modeling flip-flop (MFF) and an OR gate and propagated from x_1 to x'_1. Meanwhile any two-vector test on y s-a-0/1 can produce exclusive outputs from the fault free and slow-to-fall TDF circuit. Like the model in figure 4.3, we can also produce other similar models according the TDF type only by substituting the OR/AND gate. If any stuck-at-fault test on selection signal y exists, meaning the targeted TDF pairs are distinguished, the exclusive test is generated successfully.

Comparing with previous ATPG models that use two copies of the CUT or four copies of the CUT [22], this single time frame ATPG model reduces modeling complexity. A limitation of this model, however, is that it contain modeling flip-flops, that is to say it must be processed under sequential ATPG. In this thesis work, a combinational ATPG based model will be introduced later.

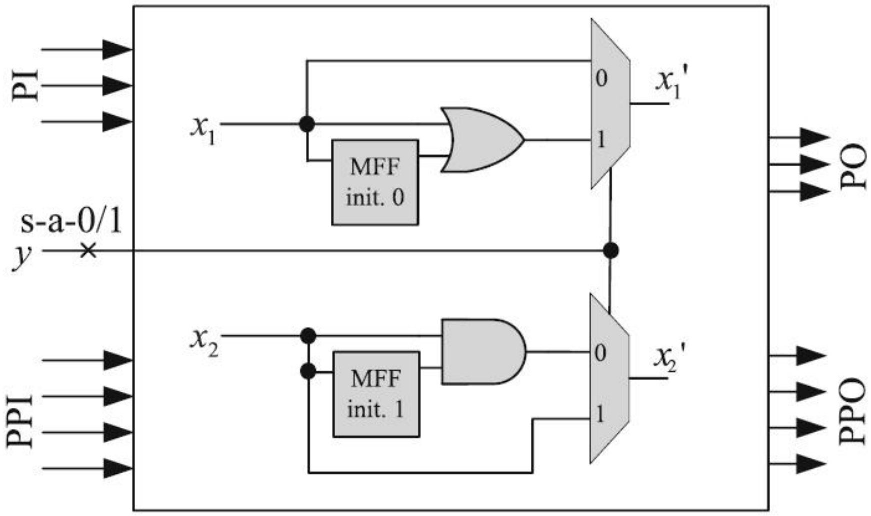


Figure 4.3: A single time frame ATPG model for exclusive test generation [48].

4.3 A New Two-Timeframe ATPG Model for Combinational ATPG

For undistinguished fault pairs, most of them are redundant (equivalent). However a small part of them are ATPG uncontrolled. Sequential ATPG requires a sequence of vectors and is also restrict to previous state, as well as limited controllability and observability of internal signals. Thus the complexity of sequential ATPG to detect redundant faults is much higher than combinational-circuit ATPG.

The combinational ATPG [28] works with a combinational ATPG algorithm, which enables it to test single nodes of the circuit and generate test vectors for combinational logic without considering the other part of the whole circuit. Meanwhile, it is worth to notice that in a combinational circuit, a redundant fault means the fault pair in the equivalent class, which also indicates the unnecessary hardware.

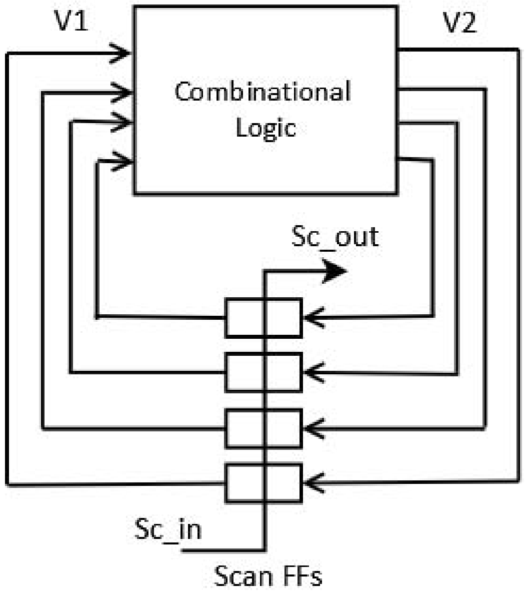


Figure 4.4: Example of scan based transition delay test.

4.3.1 Test Generation Modes

Transition delay test requests two sequence test vector which can be generated either by LOC mode or LOS mode. The illustration of a scan based transition delay test is shown in Figure 4.4.

Under LOC test generation mode, the first vector V1 is shifted through scan flip-flops (SFFs) to the combinational circuit, then output V2 will be generated, which is later captured by SFFs as the second input vector. However, the output V2 is only used for observation in LOS mode; the second vector comes from V1 shifted one bit and adds one more single scan-in bit. Thus LOC works as sequential ATPG. However a transition launched by shift works as combinational ATPG may lead to a higher fault coverage.

4.3.2 Two-Timeframe ATPG Model

To use combinational ATPG, we need to expand the CUT and the ATPG model to two-timeframe. A two-timeframe ATPG model of transition delay fault LOC test generation is shown in Figure 4.5(a).

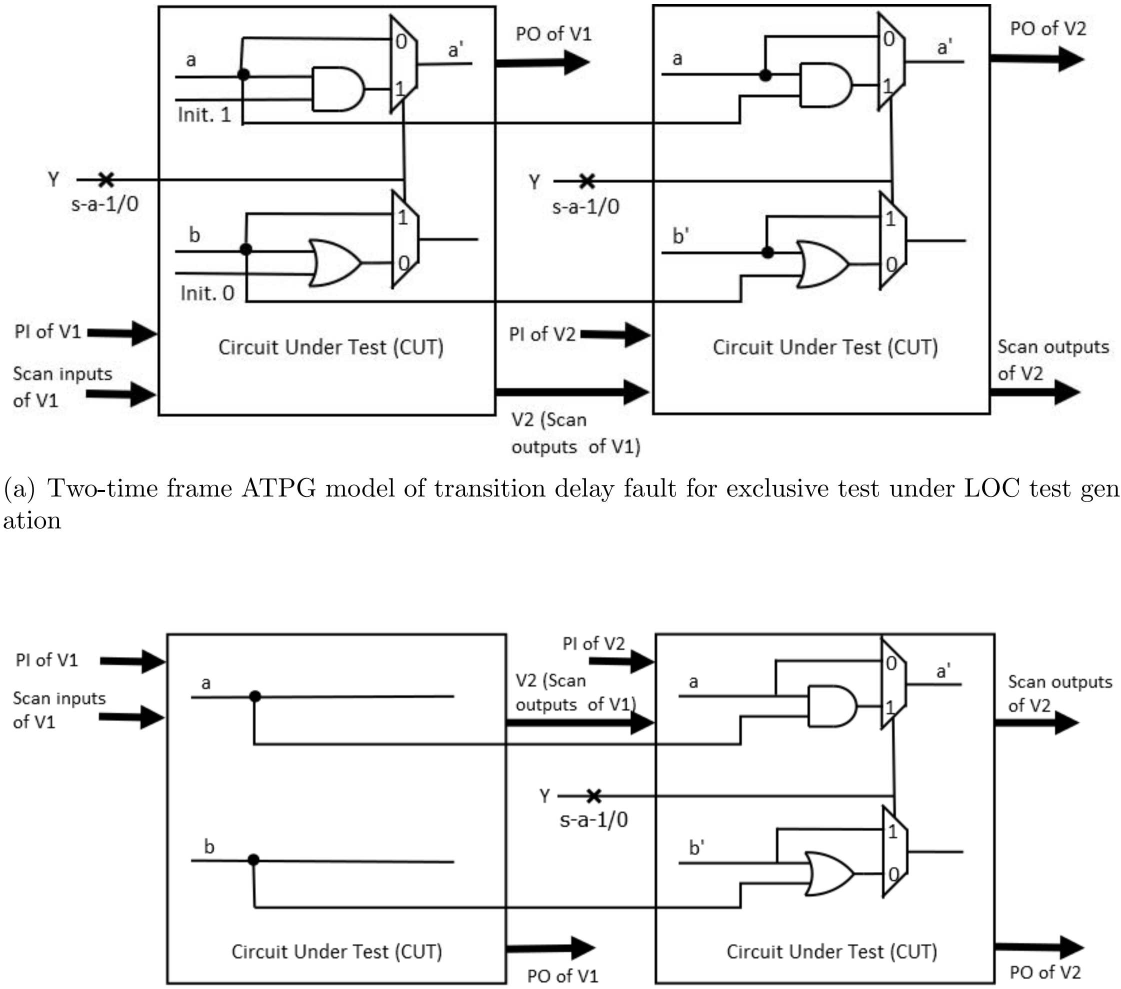
As a sequence test vector pair in transition delay test, the first vector is only to initialize the circuit and will not have any effect on the logic, so the model can be simplified as Figure 4.5(b).

Figure 4.5(a) consists of two copies of the CUT and ATPG model. PI and the scan inputs form the first test vector V1. Under LOC mode the scan outputs of the first-time frame are the scan inputs of the second-time frame. The second vector V2 comprises the scan inputs and PI of the second-time frame where the final output response can be observed through PO and scan outputs of V2. A test for stuck-at 0/1 on Y distinguishes the targeted transition delay fault pair. After two-time frame expansion, the modeling flip-flop (MFF) is eliminated, the two copy CUTs are totally combinational logic.

Figure 4.6(a) shows a combinational circuit of transition delay fault under LOS test generation. The only difference between LOS and LOC test generation is that under LOS mode we will leave the first scan output vector without concerning. The second vector directly consists of the scan inputs of first-time frame, shifted one bit, and one more single shift-in bit. An simplified LOS two-time frame model is shown in Figure 4.6(b).

4.4 Scan Test Generation

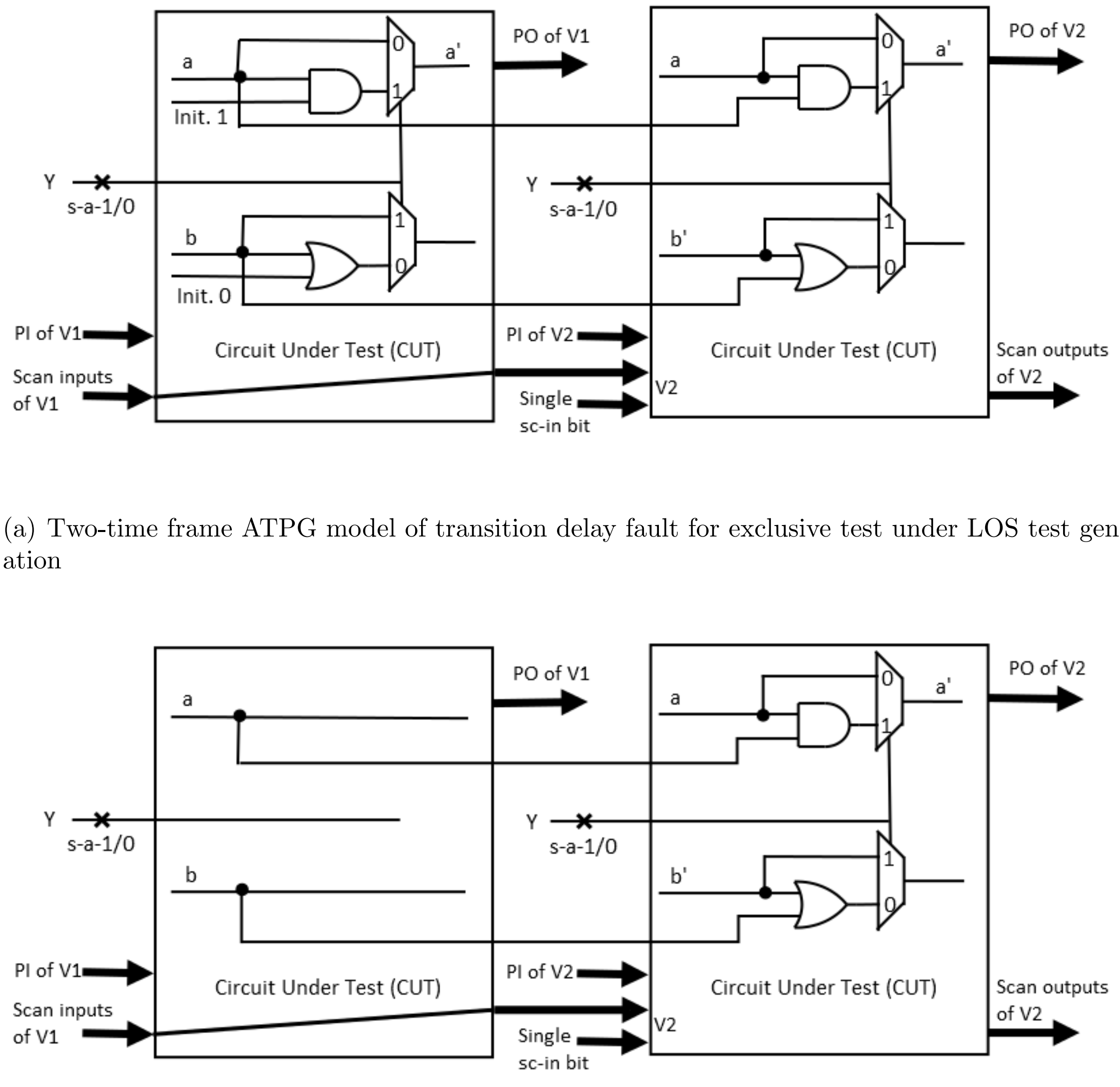
When a sequential benchmark circuit is used for a CUT, several Mentor Graphic tools are used to generate detection and diagnostic tests. At first, we need to insert internal scan circuitry to construct full-scan circuit by using DFT Advisor [3, 6]. DFT includes test point insertion, meaning the DFT Advisor tool inserts test points



(a) Two-time frame ATPG model of transition delay fault for exclusive test under LOC test generation

(b) Simplified two-time frame ATPG model of transition delay fault for exclusive test under LOC test generation

Figure 4.5: Two-timeframe ATPG model of transition delay fault under LOC mode using combinational ATPG.



(a) Two-time frame ATPG model of transition delay fault for exclusive test under LOS test generation

(b) Simplified two-time frame ATPG model of transition delay fault for exclusive test under LOS test generation

Figure 4.6: Two-timeframe ATPG model of transition delay fault under LOS mode using combinational ATPG.

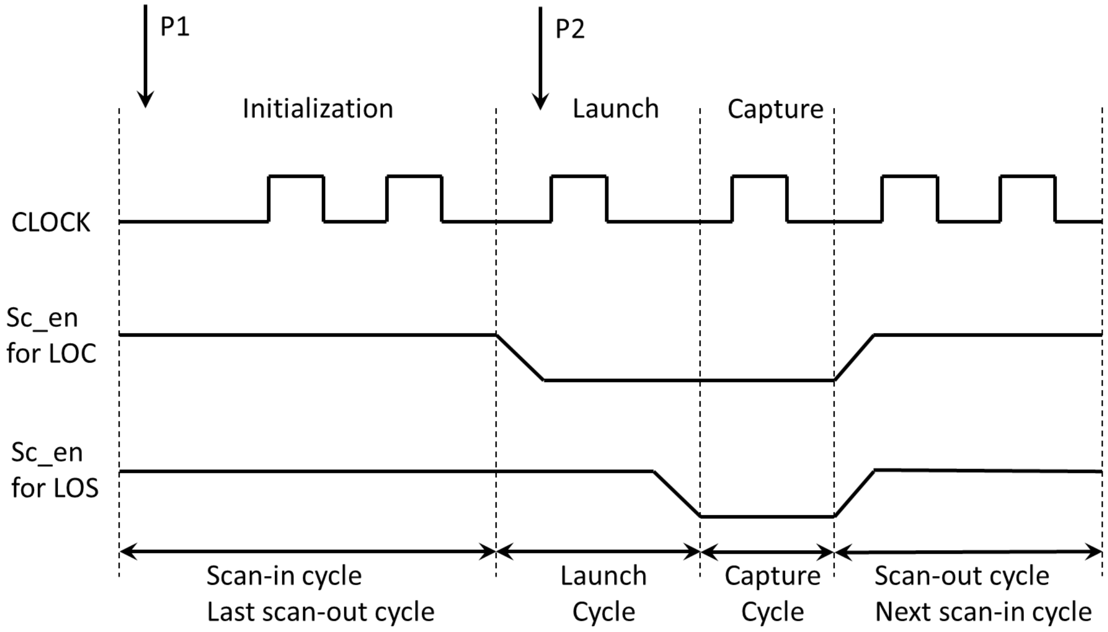


Figure 4.7: Waveforms of transition delay fault test patterns under LOC and LOS modes.

in the CUT for observation and control purpose. To convert normal flip-flops to scan flip-flops, scan_in signal, scan_out signal, scan_enable signal and a multiplexer will be added.

For both detection test and diagnosis exclusive test, we use Mentor's Fastscan [5] as the ATPG tool and integrate Fastscan instruction with the test generation system together in python. For detection tests, Fastscan has the capability to produce the vector pairs for transition delay faults. For a test vector pair, Fastscan can generate the second test vector either by launch-off-capture (LOC) test generation mode or launch-off-shift (LOS) test generation mode.

LOC launches and vector captures are controlled by the system clock pulse [35] and with no need of high speed scan enable signals. However LOS mode requires fast switch scan enable signals. The detailed procedure of transition delay fault pattern generation under LOC and LOS is shown in Figure 4.7.

During the initialization period, sc_en signal is always '1', meaning the system operates in 'scan mode'. We apply the first primary input vector, and shift in scan input vectors. For LOC mode, after shifting in the scan inputs V1, we set sc_en signal to '0' (normal mode). In the launch cycle, we apply the second primary input vector and clock the circuit, at which time the second scan input vector V2 is launched. Then, entering the capture cycle, we clock the circuit again, and we will obtain the output response. At last, we set the scan_en signal to '1', to select scan mode again, and shift out all the captured scan outputs after several clocks. We notice that the value of V2 that is launched relies on the response of V1, hence the controllability of launch path is relatively low compared to LOS mode.

However, during LOS test generation mode, sc_en signal remains '1' for one more cycle, thus in the launch cycle the system still works in 'scan mode' in which the second scan input vector V2 is launched consisting of the first scan input vector V1 shifted one bit and an additional single scan-in bit. The sc_en signal is set to '0' in the capture cycle, and the remaining is the same with LOC mode. To achieve the capture, the clock period of capture should be approximately equal to system function cycle and the sc_en signal should be changed as quick as the before capture cycle starts. That is to say, although LOS mode takes advantage of better controllability, it needs a high speed sc_en signal. However, LOC, since it has no timing constraint, is much easier to accomplish in practice.

4.5 Diagnostic Test Generation

As described in Chapter 3, the automatic test generation system is utilized in this thesis. The whole system is implemented using Python language [4]. Diagnostic exclusive test for transition delay fault process as following procedure:

1. Locate the characteristics of a fault pair and then determine the corresponding transition delay model as introduced in Figure 2.5.

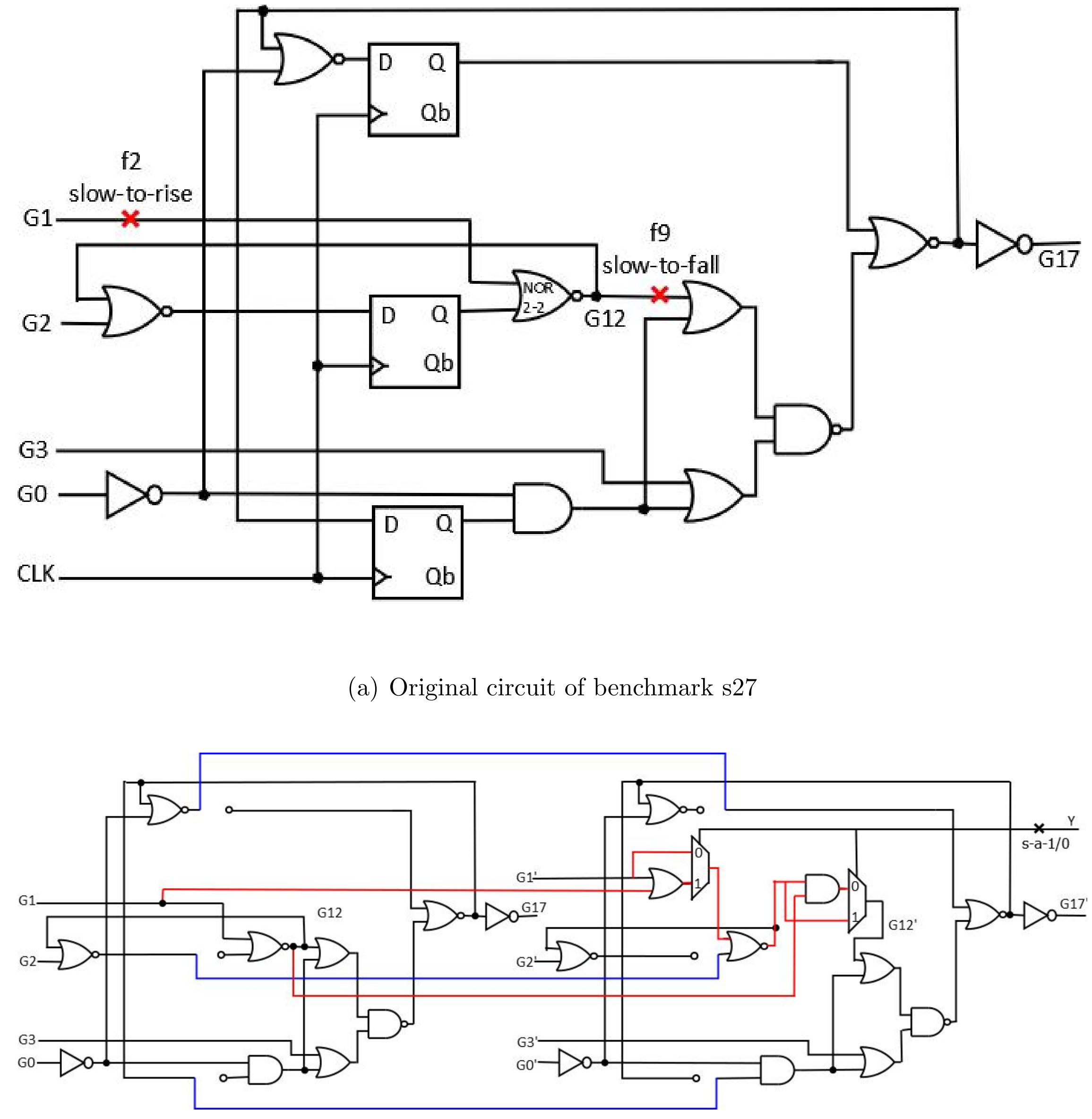
2. Construct a two-timeframe ATPG model, then insert the model into two copies of the CUT. Figure 4.7 is an illustration of a two-time frame expansion for benchmark circuit s27 under LOC mode, targeting a fault pair, f2 [slow-to-rise, G1] and f9 [slow-to-fall, NOR2_2/OUT] (f2 is slow-to-rise fault locating in PI G1, f9 is slow-to-fall fault locating in output of NOR gate 2_2).

Figure 4.8(a) shows how to expand the two-time frame CUT and how to insert the two-time frame ATPG model in detail. At first make two copies of sequential benchmark s27, eliminate flip-flops, and connect the node which is D in the original flip-flop in the first time frame to the node which is Q in the original flip-flop in the second time frame. The connection is shown in blue lines in the Figure 4.8(b).

To insert the ATPG mode properly, we need to figure out the information of the targeted fault pair. Taking f2 and f9 for example, f2 is a slow-to-rise fault with position in G1 and f9 as a slow-to-fall fault locates in output of NOR gate 2_2, then according to Figure 4.5(a), we construct ATPG model with one part modeling slow-to-rise fault and the other part modeling slow-to-fall part, and insert this model to two copied of CUT. The red lines presents the signal lines of the insertion part.

3. Generate exclusive combinational test vectors for a targeted fault pair which may be distinguished under the expanded combinational circuit. A test for stuck-at fault on y select signal is used in the exclusive generator, if there exits an exclusive test, it means the fault pair f2 and f9 can be distinguished, and one of them will be eliminated from the fault group.

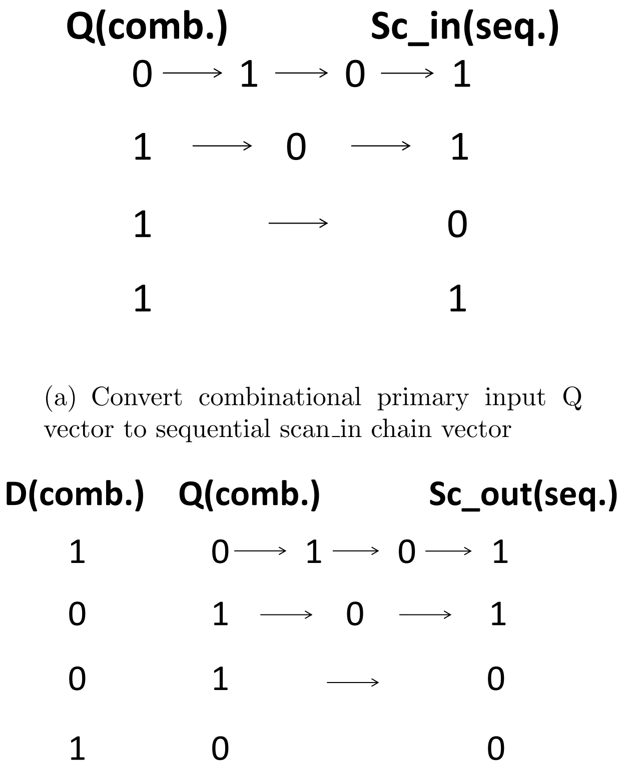
4. As a diagnostic fault simulator working on original full-scan circuit needs to be fed a sequential test vector, thus we need to convert the combinational test vector generated from procedure 3 to a sequential one. Since the scan chain is connected from Qb in the first SFF to Scan_in in next SFF by DFT Advisor, the corresponding sequential chain value is transferred from the value in combinational vector which



(a) Original circuit of benchmark s27

(b) Two-time frame expansion of s27 and ATPG model for specified transition delay fault

Figure 4.8: An illustration of two-timeframe expansion of ATPG model and benchmark circuit s27 under LOC.



(a) Convert combinational primary input Q vector to sequential scan_in chain vector

(b) Convert combinational primary input D vector to sequential scan_out chain vector

Figure 4.9: An illustration of convert principle from combinational vector to sequential vector.

should be inverted from 1 to 0 or 0 to 1 several times. An illustration of conversion principle is shown in Figure 4.9.

The way that the procedure converts combinational primary input Q vector to sequential scan_in chain vector is shown is Figure 4.9(a). As a rule, we observed that if the vector length is an even number, the even bits will be inverse and odd bits will remain invariable. Meanwhile, if the vector length is an odd number, the even bits will keep fixed and odd bits will be inverse.

While applying combinational primary input D vector to sequential scan_out chain vector as presented in Figure 4.9(b), there is one more inversion from D to Qb. Thus if the vector length is an even number, the even bits will keep constant and odd bits will become inverse, as well as if the vector length is an odd number, the even bits are turnover and odd bits are unchanged.

5. Fault simulation for transition delay faults in the original full-scan sequential benchmark circuit calculates the new DC, and generates updated undistinguished groups and a new dictionary. The diagnostic fault simulator will stop working after all faults pairs have been targeted and getting an adequate DC.

Chapter 5

Experimental Results

In this thesis, we use the Fastscan ATPG system to generate both detection test for full-scan sequential circuit and diagnostic exclusive test for transition delay fault pair under LOC mode.

An experimental application of this work to ISCAS'89 [14] benchmark circuits is presented and provides comparison with previous work.

5.1 Exclusive Test Generation for Sequential Benchmark Circuits

Table 5.1 shows the results of detection test for TDF in an ISCAS'89 full-scan sequential benchmark circuit. Table 5.2 presents the results of diagnostic exclusive test. As an example, consider benchmark s1423 for which:

Detection test:

Number of transition delay fault: 2199

Number of detection test patterns: 114

Number of undistinguished fault pairs: 355

Fault coverage (FC): 92.1% , Diagnostic coverage (DC): 80.5%

Detection CPU time: 16.7s

Diagnosis test:

Number of diagnostic test vector pairs: 73

Number of undistinguished fault pairs: 98

Largest Groups size: 5

Diagnostic coverage (DC): 92.1%

Diagnosis CPU time: 279.3

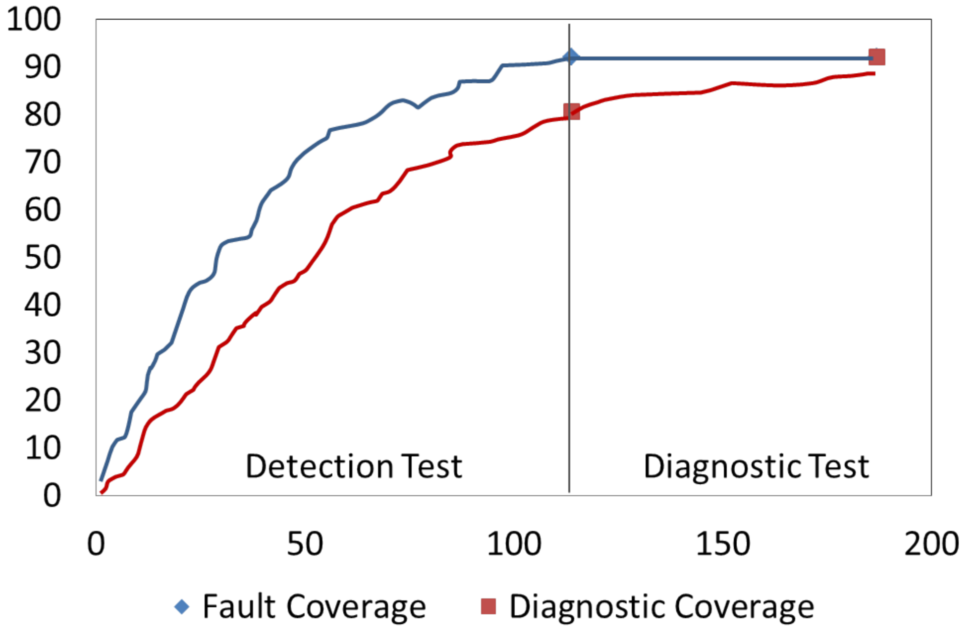


Figure 5.1: Diagnostic fault simulation on s1423.

Figure 5.1 presents the fault coverage and diagnostic coverage on benchmark circuit s1423. The detection tests have 144 patterns and have 92.1 % fault coverage. For diagnostic test, the DC raises from 80.5 % to 92.1 % using extra 73 exclusive test vector pairs.

In Table 5.1, the first column lists the benchmark circuit name. The second column presents the number of detected transition delay faults and the third column presents the number of detection test patterns. With increasing gate numbers and flip-flop numbers in these benchmark circuits, more test patterns can be generated and thus more faults can be detected. Meanwhile during the test, ATPG can detect some of redundant faults.

Diagnostic coverage is shown in column 4. Because of the limitation of LOC test vector generation mode, the DC for sequential circuits is not adequate. Column 5 shows the number of undistinguished groups. A group may contain two or more equivalent faults, and the largest size of an equivalent fault group is listed in column 6.

Table 5.1: Detection test for TDF in ISCAS'89 sequential benchmark circuits.

Circuit Name	No. of TDFs	No. of Det. Test Patterns	DC (%)	No. of Undist. Groups	Largest Groups Size	No. of Undist. Pairs	FC (%)	Det. Time (sec.)
s27	46	11	54.3	14	4	29	100.0	0.8
s298	385	44	62.4	66	4	106	79.9	2.6
s420	622	113	54.9	128	8	429	85.4	3.5
s526	570	86	47.8	92	7	226	62.5	2.9
s838	1254	231	53.3	261	8	895	83.6	8.7
s1423	2199	114	80.5	256	6	355	92.1	16.7
s5378	6007	230	83.1	354	11	877	90.6	65.7

Table 5.2: Two-timeframe diagnostic exclusive tests for TDF in ISCAS'89 sequential benchmark circuits.

Circuit Name	No. of TDFs	No. of Det. Test vector pairs	DC (%)	No. of Undist. Groups	Largest Groups Size	No. of Undist. Pairs	Diag. CPU* Time (sec.)
s27	46	17	100	1	2	1	12.9
s298	385	28	77.7	39	4	47	44.2
s420	622	47	81.0	116	7	170	155.1
s526	570	28	58.1	68	6	98	52.1
s838	1254	95	78.5	240	7	356	539.8
s1423	2199	73	92.1	174	5	199	279.3
s5378	6007	340	91.0	84	7	98	1879.8

* Hardware configuration: 2.6GHz CPU, 3.86G RAM, Intel Core i5

We observe that the number in column 7, of undistinguished pairs during detection test, is much larger than the number of undistinguished groups, which demonstrates that there are more than two undistinguished faults in one equivalent group.

Fault coverage (FC) is presented in column 8. The FC of s27 is 100, which means ATPG with highest fault detection capability is able to detect all the faults. Generally, FC is referred as the upper bound of DC [44]. The last column in Table 5.1 lists the CPU time for detection test. As the complexity of a circuit increases, the detection time increases. The results of diagnosis exclusive test are shown in Table 5.2. The column 2 is also the number of TDFs, and third column lists the number of exclusive test patterns generated by ATPG to distinguish fault pairs.

After CUT two-timeframe expansion and the two-timeframe ATPG model insertion, column 3 presents the number of generated exclusive test vector pairs, and the diagnostic coverage listed in column 4, which shows obvious improvement compared with no exclusive test before in Table 5.1. For example, DC for s27 raises from 54.3% to 100%, which means every targeted fault pair can be distinguished or proved redundant (equivalent) after the insertion of two-timeframe expansion ATPG model.

The results in columns 5, 6 and 7 are corresponding data as in Table 5.1. The enhancement observed in all these values in Tables 5.2, the undistinguished fault pairs and the undistinguished fault groups are much smaller than those before diagnostic test, which means more faults pairs have been distinguished. It is interesting to observe that, in combinational ATPG when there is no exclusive test found for the fault pair in a combinational circuit, most targeted stuck-at faults are redundant (TDF pair is equivalent). Very few cases are still ATPG untestable. However, in sequential ATPG it is much harder to prove a fault to be redundant. The two timeframe expansion with combinational ATPG thus proves its superiority in distinguishing fault pairs.

Time overhead appears especially high when a circuit become larger, due to the library constructing every time after each exclusive test. If a library compaction technique can be applied, this timing issue surely can be reduced.

Table 5.3: Comparing this work and previous work.

Circuit	No. of Det. Test vector	No. of Det. Test vector pairs	DC (%)	No. of Undist. Groups	Largest Groups Size	No. of Undist. Pairs	Diag. CPU* Time (sec.)
s27	(A) 11	17	100	1	2	1	12.9
	(B) 11	17	97.8	1	2	1	29
s298	(A) 44	28	77.7	39	4	47	44.2
	(B) 44	26	70.1	39	4	80	55
s1423	(A) 114	73	92.1	174	5	199	279.3
	(B) 102	90	84.2	182	5	208	845
s5378	(A) 230	340	91.0	84	7	98	1879.8
	(B) 208	395	89.6	85	7	99	488

*(A) This work with two-timeframe expansion ATPG model
*(B) Previous work using a single timeframe ATPG model

5.2 Comparison with Previous Work

Previous authors proposed single timeframe ATPG model [47, 48] for targeting transition faults. Table 5.3 presents experimental results comparing two-timeframe expansion ATPG model of this work (A) with single timeframe ATPG model (B). The second column shows number of generated detection test vectors, as the detection test of both is the same procedure; there are only slightly differences because of the ATPG tools self-mechanism. The number of diagnostic test vector pairs and diagnostic coverage (DC), respectively, are presented in column 3 and column 4. The DC of A are all higher than B, especially for s1423 and s5378, with less exclusive test vectors but higher DC. Column 5 and column 7 display the decreased number of undistinguished fault groups and number of undistinguished fault pairs, meaning the two-timeframe expansion ATPG model is able to distinguish more fault pairs than the single timeframe model. The diagnostic CPU time of our approach is less than that of [47, 48], except for s5378. The time overhead of s5378 is mainly because of the library reconstruction time especially when circuit becomes large, this can be solved by data compact techniques. Once this problem eliminated, the two-timeframe expansion ATPG model in combinational ATPG has better performance in various aspects than sequential ATPG logic.

Chapter 6

Conclusion and Future Work

6.1 Conclusion

The advanced fault diagnosis techniques improved with the increasing of chip density. The proposed two-timeframe expansion ATPG model under combinational logic ATPG is used to target transition delay fault pairs. If there exist an exclusive test vector for the stuck-at fault, the transition delay fault pair is distinguished, to the contrary the fault pair is proved equivalent fault when the system cannot generate test vector.

After determining wether the fault pair is targeted or not, the diagnosis fault simulator calculates the updated diagnostic efficiency by Diagnostic Coverage (DC), a diagnostic metric similar to fault coverage, to measure the ability of fault diagnosis. This is followed by repeated diagnostic exclusive test generation, then fault simulation. All the procedures are integrated in an automatic test generation system that are implemented by Python programming language.

In Chapter 3, diagnostic basic knowledge and the automatic test generation system are introduced. The automatic test generation system only uses conventional ATPG tools without raising the diagnostic complexity.

Chapter 4 provides a new two-timeframe expansion ATPG model for exclusive test. Transition delay faults are modeled by only inserting a few logic gates in the original circuit netlist, and a multiplexer inserted to distinguish the fault pair. If a test vector for a stuck-at fault at the output of the multiplexer can be generated, the faults are distinguished, in other words, the fault pairs have different output response at least at one output signal line.

Using two-timeframe expansion the ATPG model allows scan test generation to work in pure combinational ATPG program, which has been shown to have better capability to identify redundant faults. After every fault pair targeted (achieving adequate DC), the scan test generation system will stop, and the final undistinguished fault pairs, DC, and CPU consumption time will be obtained.

6.2 Future Work

6.2.1 Test Generation under LOS Mode

The test vector pair for transition delay faults can be generated by either LOC mode or LOS mode. The work in this thesis was is completed by LOC test generation mode with easy clock requirement and simple practical application, but less randomness of internal signal.

A direction for future work would two-timeframe expansion in LOS mode, in which the second vector comes from the first scan-in vector shifted a bit and one more single scan-in bit, thus guarantee its better controllability in LOS mode. In addition, fault pairs that cannot be targeted under LOC mode may still be distinguished in LOS mode [47, 48]. As a consequence, a higher DC may achieved by LOS test generation mode, and the only difference that would need to be modified in the whole test process is that changing ATPG test generation mode from LOC to LOS in Fastscan tool.

6.2.2 Utilization on stuck open fault diagnostic test

The stuck-open-fault (SOF) model reflects a signal line in the circuit is broken and thus stuck in the open state. Detection of stuck-open faults require two-vector be applied sequentially, the first vector is used to initialize the circuit and the second vector examines the output response of the fault. Transition delay fault model is also required two-vector to test delay. We can transform SOF test to TDF test problem. Which means the automatic test generation system and the two-timeframe expansion

ATPG model in this thesis work can be also applied on stuck open fault diagnostic test.

6.2.3 Integration of Data Compaction Techniques with Test Generation System

With the circuit becoming larger, CPU time overhead is mainly because of the full-response fault library reconstruction during every diagnostic test. Various researches focusing on this area have already made, such as test set compaction [11, 23] to reduce test application time, diagnostic library size reduction techniques [24, 36, 37, 38, 41] develop to reduce memory storage requirement and time consumption.

The future works combining such data compaction techniques to out automatic test generation system improve fault diagnosis efficiency.

Bibliography

[1] "Automatic Test Pattern Generation, Wikipedia, the Free Encyclopedia." http://en.wikipedia.org/w/index.php?title=Automatic_test_pattern_generation &oldid=628793647., accessed on 10/08/2014.

[2] "Built-in Self-Test, Wikipedia, the Free Encyclopedia." http://en.wikipedia.org/w/index.php?title=Built-in_self-test&oldid=648914790, accessed on 02/26/2015.

[3] "Dftadvisor Tutorial, Scribd. October 22, 2005." https://www.scribd.com/doc/-65735154/Dftadvisor-Tutorial.

[4] "The Python Tutorial, Release 2.7.10rc1." https://docs.python.org/2/tutorial/.

[5] "Mentor Graphics, FastScan and FlexTest Reference Manual," 2004.

[6] "Mentor Graphics, FastScan and FlexTest Reference Manual," 2009.

[7] M. Abramovici, M. A. Breuer, and A. D. Friedman, *Digital Systems Testing & Testable Design*. New York: Wiley-IEEE Press, first edition, 1994.

[8] V. D. Agrawal, D. H. Baik, Y. C. Kim, and K. K. Saluja, "Exclusive Test and Its Applications to Fault Diagnosis," in *Proc. 16th International Conference on VLSI Design*, 2003, pp. 143–48.

[9] S. B. Akers, C. Joseph, and B. Krisnamurthy, "On the Role of Independent Fault Sets in the Generation of Minimal Test Sets," in *Proc. International Test Conference*, Oct. 1987, pp. 1100–1107.

[10] C. Alagappan, *Dictionary-Less Defect Diagnosis as Real or Surrogate Single Stuck-At Faults* Master's thesis, Auburn University, ECE Department, Auburn, AL, 2013.

[11] V. Andreeva, "Test Set Compaction Procedure for Combinational Circuits Based on Decomposition Tree," in *Proc. 9th IEEE East-West Design Test Symposium (EWDTS)*, 2011, pp. 251–54.

[12] T. Bartenstein, "Fault Distinguishing Pattern Generation," in *Proc. International Test Conference*, 2000, pp. 820–828.

[13] N. K. Bhatti and R. D. Blanton, "Diagnostic Test Generation for Arbitrary Faults," in *Proc. International Test Conference*, 2006. Paper 19.2.

[14] F. Brglez, D. Bryan, and K. Kozminski, "Combinational Profiles of Sequential Benchmark Circuits," in *Proc. IEEE International Symp. Circuits and Systems*, volume 3, May 1989, pp. 1929–1934.

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

[16] S. C. Chen and J. M. Jou, “Diagnostic Fault Simulation for Synchronous Sequential Circuits,” *IEEE Trans. Comput-Aided Design*, vol. 16, no. 3, pp. 299–308, Mar. 1997.

[17] A. L. Crouch, *Design for Test for Digital ICs and Embedded Core Systems*. New Jersey: Prentice Hall, 1999.

[18] S. K. Devarakond, S. Sen, S. Bhattacharya, and A. Chatterjee, “Concurrent Device/Specification Cause-Effect Monitoring for Yield Diagnosis Using Alternate Diagnostic Signatures,” *IEEE Design & Test of Computers*, vol. 29, no. 1, pp. 48–58, Feb. 2012.

[19] J. Esch, “Prolog to IDDQ Testing for CMOS VLSI,” *Proceedings of the IEEE*, vol. 88, no. 4, pp. 542–543, Apr. 2000.

[20] A. D. Friedman, “Fault Detection in Redundant Circuits,” *IEEE Transactions on Electronic Computers*, vol. EC-16, no. 1, pp. 99–100, Feb. 1967.

[21] I. A. Grout, *Integrated Circuit Test Engineering: Modern Techniques*. Springer Science & Business Media, 2005. Pages 68-70.

[22] Y. Higami, Y. Kurose, S. Ohno, H. Yamaoka, H. Takahashi, Y. Shimizu, T. Aikyo, and Y. Takamatsu, “Diagnostic Test Generation for Transition Faults Using a Stuck-at ATPG Tool,” in *Proc. International Test Conference*, 2009. Paper 16.3.

[23] Y. Higami, K. K. Saluja, H. Takahashi, S. Kobayashi, and Y. Takamatsu, “Compaction of Pass/fail-Based Diagnostic Test Vectors for Combinational and Sequential Circuits,” in *Proc. Asia and South Pacific Conference on Design Automation*, 2006, pp. 659–664.

[24] Y.-R. Hong and J.-D. Huang, “Fault Dictionary Size Reduction for Million-Gate Large Circuits,” in *Proc. Asia and South Pacific Design Automation Conference*, 2007, pp. 829–34.

[25] V. S. Iyengar, B. K. Rosen, and J. A. Waicukauski, “On Computing the Sizes of Detected Delay Faults,” *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 9, no. 3, pp. 299–312, Mar. 1990.

[26] N. K. Jha and S. Gupta, *Testing of Digital Systems*. Cambridge University Press, 2003. Page 82.

[27] Y. Karkouri, E. M. Aboulhamid, E. Cerny, and A. Verreault, “Use of Fault Dropping for Multiple Fault Analysis,” *IEEE Transactions on Computers*, vol. 43, no. 1, pp. 98–103, Jan. 1994.

[28] L. Lavagno, G. Martin, and L. Scheffer, *Electronic Design Automation for Integrated Circuits Handbook, Volumes 1 and 2*. Boca Raton, FL, USA: CRC Press, Inc., 2006.

[29] D. B. Lavo and T. Larrabee, "Making Cause-Effect Cost Effective: Low-Resolution Fault Dictionaries," in *Proc. International Test Conference*, 2001, pp. 278–286.

[30] A. K. Majhi, *Algorithms for Test Generation and Fault Simulation of Path Delay Faults in Logic Circuits*. PhD thesis, Indian Institute of Science, ECE Dept., Bangalore, India, Nov. 1996.

[31] A. K. Majhi and V. D. Agrawal, "Delay Fault Models and Coverage," in *Proc. 11th International Conference on VLSI Design*, 1998, pp. 364–369.

[32] A. K. Majhi, V. D. Agrawal, J. Jacob, and L. M. Patnaik, "Line Coverage of Path Delay Faults," *IEEE Trans. VLSI Systems*, vol. 8, pp. 610–614, Oct. 2000.

[33] E. J. McCluskey and F. W. Clegg, "Fault Equivalence in Combinational Logic Networks," *IEEE Transactions on Computers*, vol. C-20, no. 11, pp. 1286–1293, Nov. 1971.

[34] S. R. Medidi, M. Medidi, and S. Gavini, "Detecting Packet-Dropping Faults in Mobile Ad-Hoc Networks," in *Proc. Thirty-Seventh Asilomar Conference on Signals, Systems and Computers*, volume 2, 2004, pp. 1708–1712. Conference date 9-12 Nov., 2003.

[35] I. Park and E. J. McCluskey, "Launch-on-Shift-Capture Transition Tests," in *Proc. International Test Conference*, 2008. Paper 35.3.

[36] M. A. Shukoor, *Fault Detection and Diagnostic Test Set Minimization* Master's thesis, Auburn University, ECE Dept., Auburn, Alabama, USA, May 2009.

[37] M. A. Shukoor and V. D. Agrawal, "A Two Phase Approach for Minimal Diagnostic Test Set Generation," in *Proc. 14th IEEE European Test Symposium*, 2009, pp. 115–120.

[38] M. A. Shukoor and V. D. Agrawal, "Diagnostic Test Set Minimization and Full-Response Fault Dictionary," *J. Electronic Testing: Theory and Applications*, vol. 28, no. 2, pp. 177–187, Apr. 2012.

[39] W. Sleszynski, J. Nieznanski, and A. Cichowski, "Open-Transistor Fault Diagnostics in Voltage-Source Inverters by Analyzing the Load Currents," *IEEE Transactions on Industrial Electronics*, vol. 56, no. 11, pp. 4681–4688, Nov. 2009.

[40] H. Tang, C. Liu, W.-T. Cheng, S. M. Reddy, and W. Zou, "Improving Performance of Effect-Cause Diagnosis with Minimal Memory Overhead," in *Proc. 16th Asian Test Symposium*, 2007, pp. 281–87.

[41] G. Tomasz, G. Krzysztof, and H. Andrzej, "How to Reduce Size of a Signature-Based Diagnostic Dictionary Used for Testing of Connections," in *Proc. 13th IEEE International Symposium on Design and Diagnostics of Electronic Circuits and Systems (DDECS)*, 2010, pp. 201–204.

[42] A. Veneris, R. Chang, M. S. Abadir, and M. Amiri, "Fault Equivalence and Diagnostic Test Generation Using ATPG," in *Proc. International Symposium on Circuits and Systems*, volume V, 2004, pp. 221–224.

[43] X. Yu, M. E. Amyeen, S. Venkataraman, R. Guo, and I. Pomeranz, "Concurrent Execution of Diagnostic Fault Simulation and Equivalence Identification during Diagnostic Test Generation," in *Proc. 21st IEEE VLSI Test Symposium*, 2003, pp. 351–356.

[44] Y. Zhang, *Diagnostic Test Pattern Generation and Fault Simulation for Stuck-at and Transition Faults*. PhD thesis, Auburn University, ECE Dept., Auburn, AL, Aug. 2012.

[45] Y. Zhang and V. D. Agrawal, "A Diagnostic Test Generation System," in *Proc. International Test Conference*, 2010. Paper 12.3.

[46] Y. Zhang and V. D. Agrawal, "An Algorithm for Diagnostic Fault Simulation," in *Proc. 11th IEEE Latin American Test Workshop (LATW)*, 2010, pp. 1–5.

[47] Y. Zhang and V. D. Agrawal, "Reduced Complexity Test Generation Algorithms for Transition Fault Diagnosis," in *Proc. 29th IEEE International Conference on Computer Design (ICCD)*, 2011, pp. 96–101.

[48] Y. Zhang, B. Zhang, and V. D. Agrawal, "Diagnostic Test Generation for Transition Delay Faults Using Stuck-At Fault Detection Tools," *J. Electronic Testing: Theory and Applications*, vol. 30, no. 6, pp. 763–80, Dec. 2014.