

NAIST-IS-DT0261007

Doctor's Thesis

Studies on Design for Delay Testability and Delay Test Generation for Sequential Circuits

Tsuyoshi Iwagaki

July 28, 2004

Department of Information Processing
Graduate School of Information Science
Nara Institute of Science and Technology

Doctor's Thesis
submitted to Graduate School of Information Science,
Nara Institute of Science and Technology
in partial fulfillment of the requirements for the degree of
DOCTOR of ENGINEERING

Tsuyoshi Iwagaki

Thesis committee: Hideo Fujiwara, Professor
Katsumasa Watanabe, Professor
Michiko Inoue, Associate Professor

Studies on Design for Delay Testability and Delay Test Generation for Sequential Circuits*

Tsuyoshi Iwagaki

Abstract

VLSI (Very Large Scale Integration) circuits are basic components of today's complex digital systems. In order to realize dependable digital systems, VLSI circuits should be highly reliable. VLSI *testing* plays an important role in satisfying this requirement. VLSI testing is to check whether faults exist in a circuit, and it consists of two main phases: *test generation* and *test application*. In test generation, a *test sequence* that is an input sequence to detect faults in a circuit is generated. In test application, the generated test sequence is applied to the circuit.

Conventional testing deals with *stuck-at faults* only. For today's high-speed VLSI circuits, testing for stuck-at faults is not sufficient to guarantee the timing correctness of the circuits. *Delay testing* that is to check whether *delay faults* exist in a circuit is an important technology to guarantee the timing correctness. For delay faults in a combinational circuit, two-pattern tests are required to detect them. On the other hand, for delay faults in a sequential circuit, we need a test sequence whose length is two or more. Test generation for sequential circuits under simple fault models such as the single stuck-at fault model is generally a hard task. Delay test generation for sequential circuits is a more challenging problem. For such sequential circuits, *design for testability (DFT)* is an important approach to reduce the test generation complexity. *Fully enhanced scan design* has been proposed as a straightforward DFT method for delay faults. In this design, every flip-flop (FF) is replaced by an *enhanced scan FF (ESFF)*.

*Doctor's Thesis, Department of Information Processing, Graduate School of Information Science, Nara Institute of Science and Technology, NAIST-IS-DT0261007, July 28, 2004.

Each ESFF can store arbitrary two bits to apply two consecutive vectors. For a sequential circuit designed by this technique, we can use a combinational delay test generation algorithm to generate a test sequence for the original circuit. Therefore, high fault efficiency, which is defined as the ratio of the sum of the number of detected faults and identified untestable faults to the total number of target faults under test generation, can be achieved with short test generation time. However, this method has the following disadvantages: (1) area overhead caused by extra DFT elements of ESFFs is high, (2) test application time is long, (3) test application at the rated speed of a given circuit, called *at-speed testing*, cannot be performed, and (4) many untestable faults, which do not affect the circuit performance, can be detected (over-testing). In this dissertation, we try to overcome these disadvantages by using two different approaches: a *partially enhanced scan* approach and a *non-scan* one.

We first propose a delay test generation method based on a partially enhanced scan technique. A new structure, called *discontinuous reconvergence structure (DR-structure)*, of sequential circuits is presented, which has easy testability for delay faults. We show that the delay test generation problem for sequential circuits with DR-structure can be reduced to that for their time-expansion models, which are combinational circuits. On the basis of the reducibility, we present a test generation method for delay faults in sequential circuits with DR-structure. Our method can be applied to several delay fault models. In order to apply our method to general sequential circuits, we use a partially enhanced scan technique. This method can improve (1) of the above disadvantages. Experimental results show that the proposed method is effective in area overhead, test generation time and fault efficiency.

Next, we propose a non-scan design scheme to enhance delay testability of sequential circuits synthesized from *state transition graphs (STGs)*. In this scheme, we utilize a given STG to test delay faults in its synthesized sequential circuit. The original behavior of the STG is used during test application. For faults that cannot be detected by using the original behavior, we design an extra logic, called an *invalid test state and transition generator*, to make those faults detectable. Our scheme allows achieving short test application time and at-speed testing, and it can improve (2)–(4) of the above disadvantages. We show the effectiveness of our

method by experiments.

Keywords:

sequential circuit, delay fault, test generation, partially enhanced scan design, non-scan design, at-speed test

List of Publications

Journal Papers

1. Tsuyoshi Iwagaki, Satoshi Ohtake and Hideo Fujiwara, "A test generation method for delay faults in sequential circuits with discontinuous reconvergence structure," *Trans. of IEICE (DI)*, Vol. J86-D-I, No. 12, pp. 872–883, Dec. 2003. (In Japanese)
2. Tsuyoshi Iwagaki, Satoshi Ohtake and Hideo Fujiwara, "A design scheme for delay testing of controllers using state transition information," *IEICE Trans. on Fundamentals of Electronics, Communications and Computer Sciences (Special Section on VLSI Design and CAD Algorithms)*. (To be published)
3. Tsuyoshi Iwagaki, Satoshi Ohtake and Hideo Fujiwara, "Delay test generation for sequential circuits based on reducibility to combinational test generation," *Journal of Electronic Testing: Theory and Applications*. (Submitted)

International Conferences

1. Tsuyoshi Iwagaki, Satoshi Ohtake and Hideo Fujiwara, "A path delay test generation method for sequential circuits based on reducibility to combinational test generation," *Digest of Papers 8th IEEE European Test Workshop*, pp. 307–312, May 2003.
2. Tsuyoshi Iwagaki, Satoshi Ohtake and Hideo Fujiwara, "Reducibility of

sequential test generation to combinational test generation for several delay fault models,” *Proc. 12th IEEE Asian Test Symposium*, pp. 58–63, Nov. 2003.

3. Tsuyoshi Iwagaki, Satoshi Ohtake and Hideo Fujiwara, ”An approach to non-scan design for delay fault testability of controllers,” *Digest of Papers 4th IEEE Workshop on RTL and High Level Testing*, pp. 79–85, Nov. 2003.
4. Tsuyoshi Iwagaki, Satoshi Ohtake and Hideo Fujiwara, “A design methodology to realize delay testable controllers using state transition information,” *Proc. 9th IEEE European Test Symposium*, pp. 168–173, May 2004.

Technical Reports

1. Tsuyoshi Iwagaki, Satoshi Ohtake and Hideo Fujiwara, “A method of test generation for path delay faults in sequential circuits of discontinuous re-convergent path structure,” *45th FTC Workshop*, July 2001. (In Japanese)
2. Tsuyoshi Iwagaki, Satoshi Ohtake and Hideo Fujiwara, “A method of partially enhanced scan design for path delay faults based on discontinuous re-convergence structure,” *Technical Report of IEICE (FTS2001-84)*, Vol. 101, No. 658, pp. 53–60, Feb. 2002. (In Japanese)
3. Tsuyoshi Iwagaki, Satoshi Ohtake and Hideo Fujiwara, “An approach to enhancing two-pattern testability of controllers,” *48th FTC Workshop*, Jan. 2003. (In Japanese)
4. Tsuyoshi Iwagaki, Satoshi Ohtake and Hideo Fujiwara, “A method of design for delay fault testability of controllers,” *Technical Report of IEICE (DC2003-38)*, Vol. 103, No. 476, pp. 25–30, Nov. 2003.

Contents

1	Introduction	1
2	Delay Testing	5
2.1.	Introduction	5
2.2.	Delay Testing for Combinational Circuits	6
2.3.	Delay Testing for Sequential Circuits	7
2.4.	Summary	7
3	Delay Test Generation for Sequential Circuits Based on Re-	
	ducibility to Combinational Delay Test Generation	9
3.1.	Introduction	9
3.2.	Preliminaries	11
3.2.1	Target Fault Models	11
3.2.2	Transformations	14
3.3.	Discontinuous Reconvergence Structure	17
3.4.	Test generation	19
3.4.1	Test Generation Method	19
3.4.2	Correctness of Our Method	20
3.5.	Design for Testability	24
3.5.1	Partially Enhanced Scan Design	24
3.5.2	Test Application	25
3.6.	Evaluation of Our Test Generation Method	26
3.6.1	Characteristics of This Work and Prior Works	26
3.6.2	Experimental Results	28
3.7.	Summary	30

4	Non-Scan Delay Testing of Sequential Circuits Using State Transition Information	35
4.1.	Introduction	35
4.2.	Preliminaries	36
4.2.1	Target Circuit and Fault Model	36
4.2.2	Terminologies	37
4.3.	Proposed Method	39
4.3.1	Test Architecture	39
4.3.2	Test Quality	41
4.3.3	Flow of Our Method	44
4.4.	Advantages of Our Method	47
4.4.1	Conventional Methods and Our Method	47
4.4.2	Experimental Results	50
4.5.	Summary	54
5	Conclusions and Future Work	57
	Acknowledgements	59
	References	60

List of Figures

2.1	Delay testing scheme for combinational circuits.	6
2.2	Slow-fast-slow testing strategy.	8
3.1	(a) Sequential circuit: S ; (b) Topology graph of S : G	12
3.2	Time-expansion graph of G : E	15
3.3	Time-expansion model of S based on E : $C_E(S)$	16
3.4	Fault transformation σ	17
3.5	Input vector pairs.	18
3.6	Enhanced scan flip-flop (ESFF).	25
3.7	Overall flow of test generation for a general sequential circuit. . .	26
3.8	Relation among three circuit structures.	27
4.1	State transition graph representing a controller.	37
4.2	Synthesized controller.	38
4.3	Combinational test generation model (CTGM).	39
4.4	Test states and transitions.	39
4.5	Test architecture.	41
4.6	Power-aware configuration.	42
4.7	Classification of untestable faults.	43
4.8	Flow chart of our method.	45

List of Tables

3.1	Two-pattern sequences.	18
3.2	Circuit characteristics.	32
3.3	Percentages of enhanced scan FFs and area overheads.	32
3.4	Test generation results for sequential circuits with DR-structure.	32
3.5	Test application time for sequential circuits with DR-structure.	33
3.6	Test generation results for respective methods.	33
3.7	Test application time for respective methods.	34
4.1	Truth table of an ISTG.	46
4.2	Distance matrix.	48
4.3	Circuit characteristics.	51
4.4	Hardware overheads.	52
4.5	Test generation results.	55
4.6	Detail of each step.	56

Chapter 1

Introduction

Nowadays, digital systems are widely used in various aspects of daily life. The key components of digital systems are VLSI (Very Large Scale Integration) circuits, and malfunctions of the circuits will affect the behavior of digital systems. The incorrect behavior of digital systems causes serious accidents if the digital systems are used as lifeline systems. Therefore, in order to realize dependable digital systems, VLSI circuits should be highly reliable. VLSI *testing* plays an important role in satisfying this requirement. VLSI testing is to check whether faults exist in a circuit, and it consists of two main phases: *test generation* and *test application*. In test generation, a *test sequence* that is an input sequence to detect faults is generated. In test application, the generated test sequence is applied to the circuit.

In the early days of digital systems, a main concern was the logical correctness of circuits. With improvements in the semiconductor process technology, the speed of modern circuits is drastically increasing. For such high-speed circuits, *delay testing* that checks whether *delay faults* exist in a circuit is becoming an important technology to guarantee the timing correctness of the circuit because conventional testing for *stuck-at faults* is not sufficient to guarantee it. For delay faults in a combinational circuit, two-pattern tests are required to detect them. On the other hand, for delay faults in a sequential circuit, we need a test sequence whose length is two or more. Test generation for sequential circuits under simple fault models such as the single stuck-at fault model is generally a hard task. Delay test generation for sequential circuits is a more challenging problem. For

such sequential circuits, *design for testability (DFT)* is an important approach to reduce the test generation complexity.

A *fully enhanced scan* method has been proposed as a straightforward DFT method for delay faults [11]. In this design, every flip-flop (FF) is replaced by an *enhanced scan FF (ESFF)*. Each ESFF can store arbitrary two bits to apply two consecutive vectors. For a sequential circuit designed by this technique, we can use a combinational *delay test generation algorithm (ATPG)* to generate a test sequence for the original circuit. Therefore, high fault efficiency¹ can be achieved with short test generation time. However, this method has the following disadvantages: (1) area overhead caused by extra DFT elements of ESFFs is high, (2) test application time is long, (3) test application at the rated speed of a given circuit, called *at-speed testing*, cannot be performed, and (4) many untestable faults, which do not affect the circuit performance, can be detected (over-testing).

One possible solution to improve (1) of the disadvantages in the fully enhanced scan method is to reduce the number of FFs to be replaced by ESFFs. This idea can be derived from experiences in testing of stuck-at faults using *partial scan* [2, 12] techniques. Several guidelines for partial scan design have been discussed [4, 13, 15, 19, 21]. On the basis of these guidelines, some methods using *partially enhanced scan* techniques, in which each FF in a subset of FFs in a circuit is replaced by ESFFs, have been proposed [5, 7, 25]. In a partially enhanced scan technique [5, 7], for a given sequential circuit, FFs to be replaced with ESFFs are selected such that feedback paths in the circuit are broken if these FFs are removed. For a sequential circuit designed by this technique, we can consider the circuit to be a feedback free circuit during test generation, and test generation for the feedback free circuit is easier than that for the original circuit. However, there is room for facilitating test generation because it still requires a sequential ATPG. In order to overcome this problem, a partially enhanced scan method based on *balanced structure* [15] has been proposed [25]. In this method, test sequences for delay faults in balanced sequential circuits can be generated by applying a

¹Fault efficiency is defined as $100 \times (n_{\text{det}} + n_{\text{unt}}) / n_{\text{total}}$, where n_{det} is the number of detected faults, n_{unt} is the number of identified untestable faults, and n_{total} is the total number of target faults under test generation.

combinational ATPG to their combinationally equivalent circuits. Therefore, this method can drastically ease the test generation complexity compared with that of the method [5] although the area overhead increases. In this dissertation, we discuss a partially enhanced scan method that has the same advantage as the method [25], i.e., test sequences can be generated by a combinational ATPG, and that can reduce area overhead.

Another possible solution to radically improve the disadvantages in the fully enhanced scan method is not to use enhanced scan techniques but to use alternatives such as *non-scan* techniques. Today's VLSI circuits are usually designed at register transfer (RT) level. An RT-level circuit is generally composed of a controller, which is represented by a state transition graph (STG), and a data path, which is represented by hardware elements such as registers, multiplexers (MUXs) and operational modules. Scan-based DFT methods are usually applied at logic level although VLSI circuits are designed at RT-level. In the recent decade, DFT methods at RT-level have been proposed [22]. For stuck-at faults, several non-scan DFT methods using RT-level information have been proposed for controllers and data paths [14, 24, 31]. For a given circuit designed at RT-level, these methods utilize its RT-level information to test stuck-at faults. These non-scan methods for stuck-at faults have improved the disadvantages of scan-based methods by using RT-level information. On the other hand, for delay faults, a non-scan method for data paths has been proposed [1]. Compared with the fully enhanced scan method, this method can achieve significantly short test application time with low area overhead. However, for controllers, there is no effective method based on non-scan techniques to overcome the disadvantages of the fully enhanced scan method. In this dissertation, we also discuss a non-scan design for delay testability of controllers.

The rest of this dissertation is organized as follows. Chapter 2 gives the basics of delay testing. In Chapter 3, we propose a delay test generation method based on a partially enhanced scan technique. A new structure, called *discontinuous reconvergence structure (DR-structure)*, of sequential circuits is presented, which has easy testability for delay faults. We show that the delay test generation problem for sequential circuits with DR-structure can be reduced to that for their time-expansion models, which are combinational circuits. On the basis of

the reducibility, we present a test generation method for delay faults in sequential circuits with DR-structure. This method can be applied to several delay fault models. In order to apply our method to general sequential circuits, we use a partially enhanced scan technique. The proposed method can improve (1) of the disadvantages of the fully enhanced scan method. Experimental results show that the proposed method is effective in area overhead, test generation time and fault efficiency. Chapter 4 proposes a non-scan design scheme to enhance delay testability of controllers. In this scheme, we utilize a given STG to test delay faults in its synthesized controller. The original behavior of the STG is used during test application. For faults that cannot be detected by using the original behavior, we design an extra logic, called an *invalid test state and transition generator*, to make those faults detectable. Our scheme allows achieving short test application time and at-speed testing, it can improve (2)–(4) of the disadvantages of the fully enhanced scan method. We show the effectiveness of our method by experiments. Finally, in Chapter 5, we conclude this work and discuss directions for future work.

Chapter 2

Delay Testing

2.1. Introduction

Delay faults in a circuit affect the performance of the circuit. Delay faults can be modeled in different ways such as the transition fault model, path delay fault model and segment delay fault model [20].

The *transition fault model* [10] is based on gross delay in a circuit. There are two transition faults associated with each gate: a slow-to-rise fault and a slow-to-fall fault. Under the transition fault model, the extra delay caused by a transition fault is assumed to be large enough to prevent the transition from reaching any primary output within a specified time. In other words, a transition fault can be observed independently of whether the transition propagates through a long or a short path to any primary output.

The *path delay fault model* [26] is based on distributed delay in a circuit. A path is a concatenation of signal lines and gates starting from a primary input or a flip-flop (FF) to a primary output or an FF. A path is regarded as faulty if its propagation delay exceeds a specified limit.

The *segment delay fault model* [16] represents a trade-off between the transition fault model and path delay fault model. A segment is a partial path and this model assumes that a segment delay fault is large enough to cause a delay fault on every path that includes the segment.

In this chapter, we summarize delay testing for combinational circuits and sequential circuits.

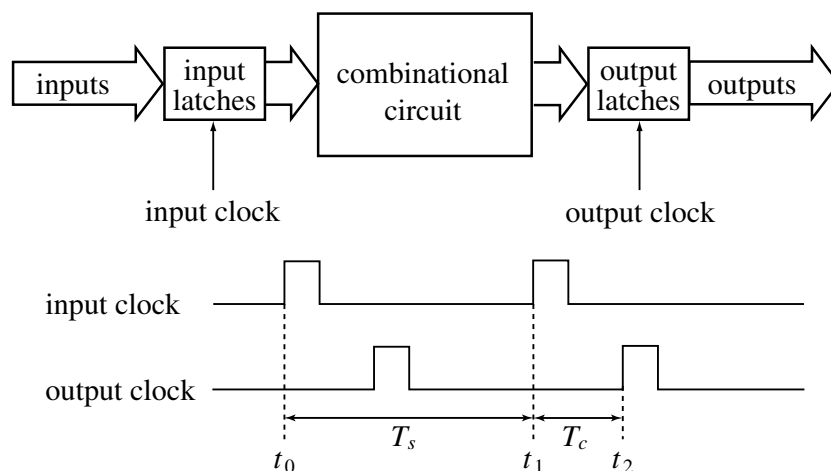


Figure 2.1. Delay testing scheme for combinational circuits.

2.2. Delay Testing for Combinational Circuits

In order to detect a delay fault, a two-pattern test $V = (v_1, v_2)$ is required. The first vector v_1 initializes a given circuit while the second vector v_2 causes the desired transitions. Figure 2.1 illustrates the test application scheme for combinational circuits. During normal operation, only one clock (system clock) is used to control the input and output latches and its period is T_C . During test mode, the input and output latches are controlled by two different clocks: the input and output clocks. It is assumed that the period of these clocks, T_S is larger than T_C . The input and output clocks are skewed by an amount equal to T_C . At time t_0 and t_1 , v_1 and v_2 are applied to the primary inputs, respectively. It is assumed that $T_S = t_1 - t_0$ is sufficient for all signals in the circuit to stabilize under v_1 . After applying v_2 , the circuit is allowed to settle down only until t_2 , where $t_2 - t_1 = T_C$. At t_2 , the primary output values are observed and compared to an expected response of a fault-free circuit to determine if there is a delay fault.

2.3. Delay Testing for Sequential Circuits

Delay testing for sequential circuits is significantly difficult compared with that for combinational circuits. This is because, for a given sequential circuit, it is hard to apply arbitrary vector pairs to FFs. In delay testing for sequential circuits, a sequence of vectors is required. This sequence is comprised of vectors for fault initialization, fault activation and fault effect propagation. In the fault initialization, values of FFs are set to the values required for the fault activation. In the fault effect propagation, the fault effect induced by the fault activation is propagated from an FF (a next state line) to a primary output. The fault initialization and fault effect propagation require a sequence of vectors while the fault activation requires a vector pair. The existence of delay faults in the fault initialization and fault effect propagation can interfere with the activation or observation of a target delay fault.

There are two ways to apply a test sequence to a given sequential circuit. One is a *slow-fast-slow* testing strategy [6]. The other is an at-speed testing strategy [3]. In the slow-fast-slow testing strategy, the vectors for the fault initialization and fault effect propagation are applied at a low clock speed such that the circuit can be considered as a delay fault-free circuit in those phases. In the fault activation, the first vector is applied at a slow clock speed while the second vector is applied at the rated clock speed of the circuit. Figure 2.2 illustrates the slow-fast-slow testing strategy. On the other hand, in the at-speed testing strategy, the test sequence is always applied at the rated clock speed. Therefore, delay faults are present in not only the fault activation but also the fault initialization and fault effect propagation. Faults that are untestable under the slow-fast-slow testing strategy remain untestable under the at-speed testing strategy. However, the converse may not be true [23].

2.4. Summary

This chapter introduced the basics of delay testing. Delay testing for combinational circuits and sequential circuits was discussed, and two test application schemes (slow-fast-slow testing and at-speed testing) were explained.

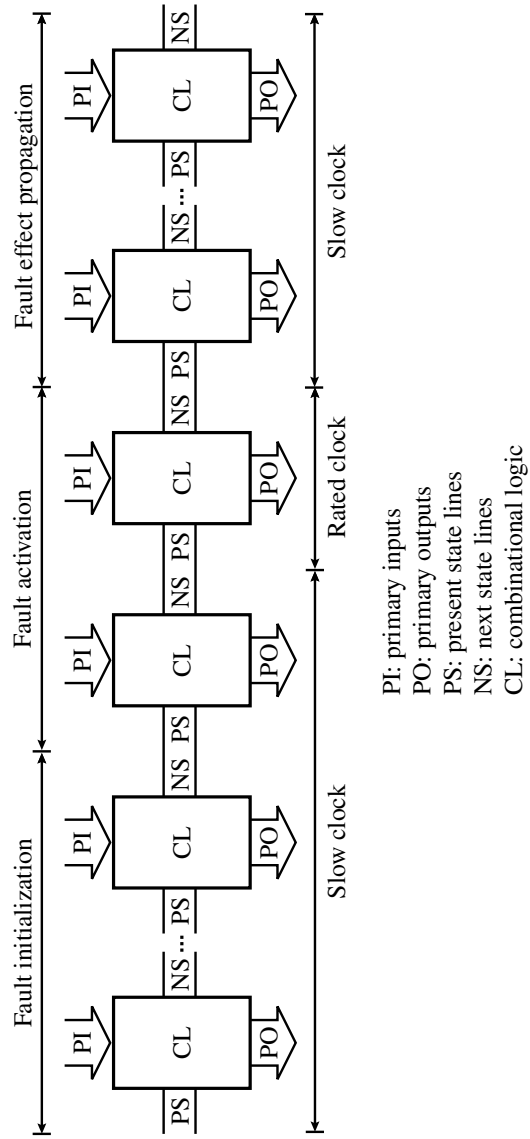


Figure 2.2. Slow-fast-slow testing strategy.

Chapter 3

Delay Test Generation for Sequential Circuits Based on Reducibility to Combinational Delay Test Generation

3.1. Introduction

Test generation for sequential circuits under simple fault models such as the single stuck-at fault model is itself generally a hard task. Delay test generation for sequential circuits is a more challenging problem. For such sequential circuits, *design for testability (DFT)* is an important approach to reduce the test generation effort. Given a sequential circuit, a fully enhanced scan technique [11] replaces each flip-flop (FF) by an *enhanced scan FF (ESFF)*. Each ESFF can store two bits to apply two consecutive vectors. For a sequential circuit designed by this technique, we can use a combinational *delay test generation algorithm (ATPG)* to generate test sequences. Therefore, high fault efficiency can be achieved with short test generation time. However, area overhead caused by extra DFT elements of ESFFs is high. It can be alleviated by using partially enhanced scan techniques [5, 25]. In a partially enhanced scan technique [5], for a sequential circuit, FFs to be replaced with ESFFs are selected such that feedback paths in

the circuit are broken if these FFs are removed. For a sequential circuit designed by this partial scan technique, we can consider the circuit to be a feedback free circuit during test generation, and test generation for the feedback free circuit is easier than that for the original circuit. However, there is room for facilitating test generation because it still requires a sequential ATPG. In order to overcome this problem, a partially enhanced scan method based on *balanced structure* [15] has been proposed [25]. In this method, test sequences for delay faults in balanced sequential circuits can be generated by applying a combinational ATPG to their combinationally equivalent circuits. Therefore, this method can drastically ease the test generation complexity compared with that of the method [5] although the area overhead increases. In this chapter, we discuss an extended class of sequential circuits for which test sequences can be generated by a combinational ATPG.

This chapter presents a new structure, called *discontinuous reconvergence structure (DR-structure)*, of sequential circuits. The relation among three classes of sequential circuits is as follows: {the class of acyclic sequential circuits} \supset {the class of sequential circuits with DR-structure} \supset {the class of balanced sequential circuits}. DR-structure has a property of easy testability for delay faults: test sequences for delay faults in sequential circuits with DR-structure can be generated by applying a combinational ATPG to their equivalent combinational circuits. For acyclic sequential circuits, notation of time-frames [21] and notation of *time-expansion models* [18] have been proposed as ways to denote equivalent combinational circuits. Our method employs time-expansion models as notation of equivalent combinational circuits. In this chapter, we show the reducibility of test generation for delay faults in a sequential circuit with DR-structure to that for the corresponding delay faults in its time-expansion model. On the basis of the reducibility, we propose a delay test generation method for sequential circuits with DR-structure. By experiments, we confirm the following: test generation time can be reduced and fault efficiency can be enhanced by using our method instead of an ordinary method using a sequential ATPG. In order to apply the proposed method to general sequential circuits, we use a partially enhanced scan technique. Theoretically, DR-structure can be extracted from the circuits with low area overhead compared with balanced structure. In this chapter, we also

confirm it experimentally.

3.2. Preliminaries

In general, a sequential circuit consists of combinational logic blocks (CLBs) connected with each other directly or through FFs. A CLB in the circuit is a region of connected combinational logic. The circuit can be modeled by a weighted directed graph defined as follows.

Definition 1 The *topology graph* for a sequential circuit S is a weighted directed graph $G = (V, A, w)$, where

- V is the set of vertices representing primary inputs, primary outputs and CLBs in S ,
- $A \subset V \times V$ is the set of arcs representing FFs and wires in S , and
- $w : A \mapsto \{0\} \cup N$, where N is the set of natural numbers, defines the weights of the arcs, and $w(u, v)$ ($u, v \in V$) denotes the number of FFs on a connection $(u, v) \in A$.

□

Example 1 Examples of a sequential circuit and its topology graph are shown in Figure 3.1. In Figure 3.1(a), 1, 2, \dots , 6 are CLBs, and black blocks are FFs. □

In this chapter, we assume that FFs have no hold capability, and those are of D-type. This assumption does not impose restriction on circuit representation because any FF with hold capability or the other types of FFs can be modeled by a D-type FF and some logic gates.

3.2.1 Target Fault Models

In this chapter, we consider three delay fault models: the path delay fault model, segment delay fault model and transition fault model. However, in the following discussion, we focus on the segment delay model because it can represent both the path delay fault model and the transition fault model.

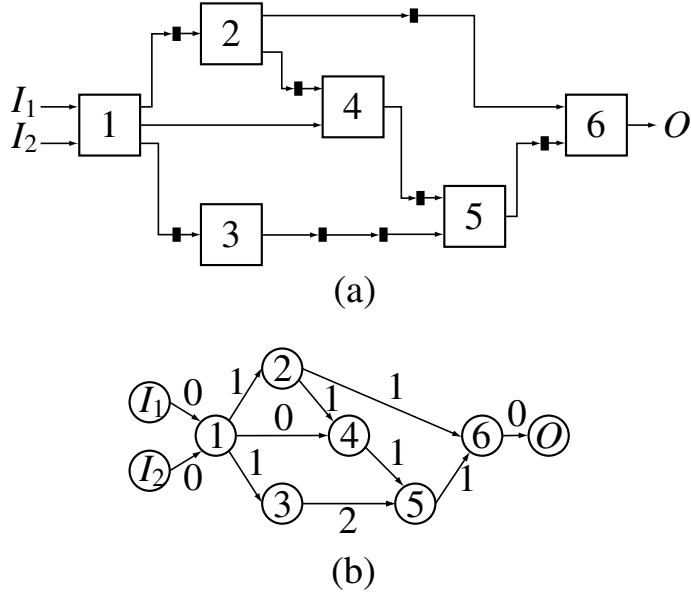


Figure 3.1. (a) Sequential circuit: S ; (b) Topology graph of S : G .

In a circuit, a segment is defined as an ordered set of gates (g_1, g_2, \dots, g_L) , where L is the length of the segment, and gate g_i is an input to gate g_{i+1} ($1 \leq i \leq L-1$). The length of the segment, L , can be anywhere from 1 to L_{\max} , where L_{\max} represents the number of gates in the longest path in the circuit. A segment has a delay fault if propagation time of rising or falling transition through the segment exceeds a specified limit. Such a delay fault on a segment is said to be a *segment delay fault (SDF)* [16]. It is assumed that a segment delay fault is large enough to cause delay faults on all paths that include the segment. In test generation for the segment delay fault model, the fault list consists of all segments whose length is L and all paths whose length is less than L . When $L = 1$, the segment delay fault model reduces to the transition fault model. When $L = L_{\max}$, it is equivalent to the path delay fault model [16].

Next, we define the testability of an SDF in both sequential circuits and combinational circuits. Note that, in the following discussion, we do not distinguish conditions of off-inputs for simplicity although testable delay faults can be classified as follows: robust testable, non-robust testable and functional sensitiz-

able [20].

Definition 2 Let S and s be a sequential circuit and a segment in S , respectively. Let f and S_f be the SDF on s and the faulty circuit of S with f , respectively. Let C be the combinational circuit composed of all the CLBs on s , and let t be a specified clock period of S . In a *slow-fast-slow testing* [20] strategy, f is *testable* if there exists an input sequence T for S and S_f such that the following conditions hold.

1. By applying an input vector pair (v_1, v_2) to C , the desired transition is launched at the starting point of s , and the transition is propagated to the ending point of s along s . Then, at time t , the value induced by v_2 at the ending point of s in S_f is different from that in S .
2. By applying T to S_f , (v_1, v_2) is justified to C , and the fault effect of f at the ending point is propagated to a primary output.

Such an input sequence T is regarded as a *test sequence* for f . □

In this chapter, we assume a slow-fast-slow testing strategy in test application because a sequential circuit can be considered delay fault-free in both the fault initialization and the fault effect propagation phases.

Definition 3 Let C and s be a combinational circuit and a segment in C , respectively. Let f and C_f be the SDF on s and the faulty circuit of C with f , respectively. Let t be a specified limit time. The fault f is *testable* if there exists an input vector pair (v_1, v_2) for C and C_f such that the following conditions hold.

1. By applying (v_1, v_2) to C and C_f , the desired transition at the starting point of s is launched, and the transition is propagated to the ending point of s along s . Then, at time t , the value induced by v_2 at the ending point of s in C_f is different from that in C .
2. The fault effect of f at the ending point of s is propagated to a primary output by applying (v_1, v_2) to C_f .

Such an input vector pair (v_1, v_2) is regarded as a *two-pattern test* for f . □

3.2.2 Transformations

In the test generation method proposed in Section 3.4, test sequences for delay faults in sequential circuits with DR-structure are generated by applying a combinational ATPG to their equivalent combinational circuits. We employ *time-expansion models* [18] as notation of equivalent combinational circuits. A time-expansion model for an acyclic sequential circuit is defined based on the following *time-expansion graph* [18].

Definition 4 Let S be an acyclic sequential circuit, and let $G = (V, A, w)$ be the topology graph of S . Let $E = (V_E, A_E, t, l)$ be a directed graph, where V_E is the set of vertices, A_E is the set of arcs, t is a mapping from V_E to the set of integers, and l is a mapping from V_E to V . If E satisfies the following four conditions, E is said to be a *time-expansion graph (TEG)* of G .

C1 (CLB preservation) The mapping l is surjective, i.e., $\forall v \in V, \exists u \in V_E$ s.t. $v = l(u)$.

C2 (Input preservation) Let u be a vertex in E . For any direct predecessor of $l(u)$ in G , $v \in \text{pre}(l(u))$, there exists a vertex u' in E such that $u' \in \text{pre}(u)$ and $l(u') = v$, where $\text{pre}(x)$ is the set of direct predecessors of a vertex x .

C3 (Time consistency) For any arc $(u, v) \in A_E$, there exists an arc $(l(u), l(v)) \in A$ such that $t(v) - t(u) = w(l(u), l(v))$.

C4 (Time uniqueness) For any vertices $u, v \in V_E$, if $t(u) = t(v)$ and $l(u) = l(v)$, then the vertices u and v are identical, i.e., $u = v$.

□

Example 2 Figure 3.2 shows the TEG of G (Figure 3.1(b)). In Figure 3.2, the character denoted in a vertex is that of the corresponding vertex in G , and the number located at the top of each column denotes the value of the label of vertices in the column. The graph E satisfies all the conditions in Definition 4. □

Note that a TEG of an acyclic sequential circuit is unique if the circuit is a single-output one [18]. This property does not hold if C4 of Definition 4 is absent.

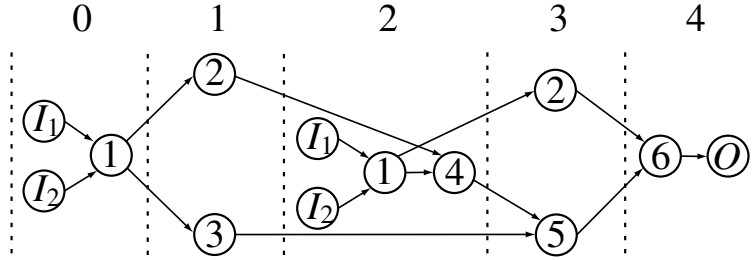


Figure 3.2. Time-expansion graph of $G: E$.

Definition 5 Let S be an acyclic sequential circuit, and let $G = (V, A, w)$ be the topology graph of S . Let $E = (V_E, A_E, t, l)$ be a TEG of G . The combinational circuit $C_E(S)$ obtained by the following procedure is said to be the *time-expansion model (TEM)* of S based on E [18].

1. For each vertex $u \in V_E$, let $l(u) \in V$ be the CLB corresponding to u .
2. For each arc $(u, v) \in A_E$, connect the output of u to the input of v with a wire in the same way as $(l(u), l(v)) \in A$. Note that the connection corresponding to (u, v) has no FF even if the connection corresponding to $(l(u), l(v))$ has some FFs (i.e., $w(l(u), l(v)) \neq 0$).
3. In each CLB, lines and logic gates that are reachable to neither other CLBs nor primary outputs are removed.

□

Example 3 Figure 3.3 shows the TEM of S (Figure 3.1(a)) based on E (Figure 3.2). In Figure 3.3, a highlighted part in a CLB represents a portion of the lines and gates removed by Step 3 in Definition 5. □

Next, we define the following transformation that represents the relation between segment delay faults in an acyclic sequential circuit and segment delay faults in its TEM.

Definition 6 Let S be an acyclic sequential circuit, and let $G = (V, A, w)$ be the topology graph of S . Let $E = (V_E, A_E, t, l)$ be a TEG of G , and let $C_E(S)$ be the

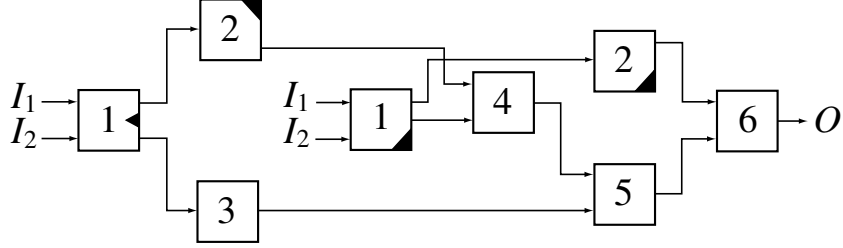


Figure 3.3. Time-expansion model of S based on E : $C_E(S)$.

TEM of S based on E . Let f be the SDF on a segment s in S , and let C be the combinational circuit composed of all the CLBs on s in S . Let B be the set of the combinational circuits corresponding to C in $C_E(S)$, and let B' the subset of B in which the input (resp. output) corresponding to the starting (resp. ending) point of s in $C_E(S)$ is not removed. A transformation such that $B' = \mu(C)$ is said to be the *sub-circuit transformation*¹. Let s' in each $b' \in B'$ be the segment corresponding to s , and let F_E be the set of SDFs composed of all the s' . A transformation such that $F_E = \sigma(f)$ is said to be the *fault transformation*². \square

Example 4 Figure 3.4 illustrates the fault transformation. In general, an SDF in S corresponds to one or more SDFs in $C_E(S)$. Notice that, from Definition 4, there exists at least one SDF in $C_E(S)$ corresponding to an SDF in S even though either lines or logic gates or both in $C_E(S)$ are removed by Step 3 in Definition 5. \square

Here, we define the following transformation that represents the relation between input sequences in an acyclic sequential circuit and input vector pairs in its TEM.

Definition 7 Let S be an acyclic sequential circuit, and let $G = (V, A, w)$ be the topology graph of S . Let $E = (V_E, A_E, t, l)$ be a TEG of G , and let $C_E(S)$ be the TEM of S based on E . Let t_{\min} be the minimum value of labels assigned to vertices in E , and let d be the sequential depth of S . Let $I_u = (v_1, v_2)$ be

¹Transforming $b' \in B'$ into C is denoted as μ^{-1} .

²Transforming $f_e \in F_E$ into f is denoted as σ^{-1} .

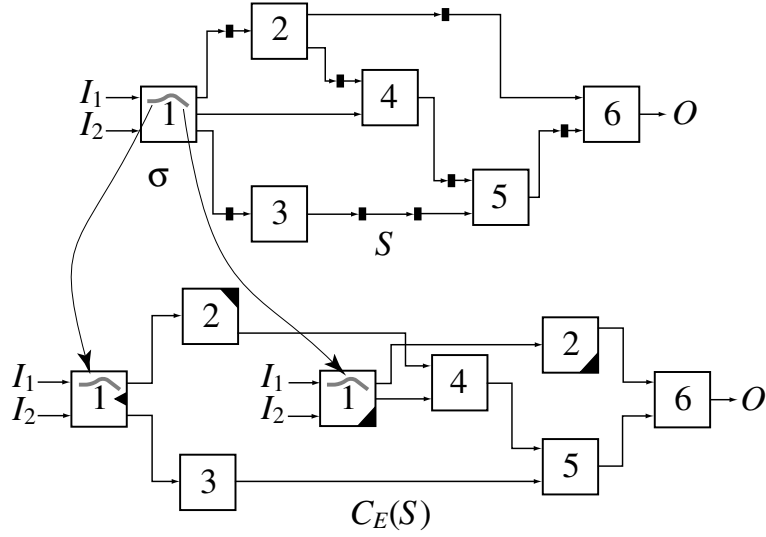


Figure 3.4. Fault transformation σ .

an input vector pair to each primary input $u \in V_E$ in $C_E(S)$. A procedure transforming I_u into the input pattern to the primary input $l(u) \in V$ of S at time k ($= 0, 1, \dots, d + 1$) is said to be the *sequence transformation* τ . That is, for each u ,

$$I_{l(u)}(k) = \begin{cases} v_1 & \text{if } k = t(u) - t_{\min} \\ v_2 & \text{if } k = t(u) - t_{\min} + 1 \\ \text{don't care} & \text{otherwise.} \end{cases}$$

Such an input sequence with the length $d+2$ is regarded as a *two-pattern sequence*. □

Example 5 The input vector pairs shown in Figure 3.5 are transformed into the two-pattern sequences shown in Table 3.1 by using the sequence transformation τ . In Table 3.1, X denotes *don't care* value. □

3.3. Discontinuous Reconvergence Structure

Our test generation method proposed in Section 3.4 generates test sequences for delay faults in sequential circuits with *discontinuous reconvergence structure*. We

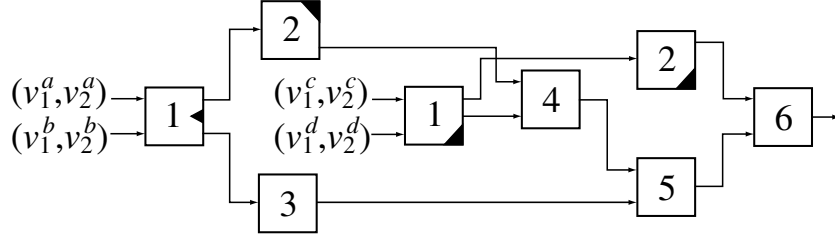


Figure 3.5. Input vector pairs.

Table 3.1. Two-pattern sequences.

Primary input	Time					
	0	1	2	3	4	5
I_1	v_1^a	v_2^a	v_1^c	v_2^c	X	X
I_2	v_1^b	v_2^b	v_1^d	v_2^d	X	X

define the structure as follows.

Definition 8 Let $G = (V, A, w)$ be the topology graph of an acyclic sequential circuit S , and let $P(u, v)$ be the set of paths from u to v ($u, v \in V$). Let $n(p)$ ($p \in P(u, v)$) be the number of FFs on a path p . The circuit S is said to be *discontinuous reconvergence structure (DR-structure)* if it satisfies the following condition.

$$|n(p_i) - n(p_j)| \neq 1 \quad (\forall u, v \in V, \forall p_i, p_j \in P(u, v))$$

□

If a sequential circuit is DR-structure, it is guaranteed that conflict of patterns does not occur in the sequence transformation. In general, an acyclic sequential circuit does not satisfy this property.

Example 6 An acyclic sequential circuit S (Figure 3.1(a)) satisfies Definition 8. Therefore, S is a sequential circuit with DR-structure. □

Notice that, from Definition 8, the class of sequential circuits with DR-structure properly includes that of balanced sequential circuits [15, 25].

3.4. Test generation

In this section, we propose a delay test generation method for sequential circuits with DR-structure, and discuss the correctness of the method.

3.4.1 Test Generation Method

Given a sequential circuit, S , with DR-structure, our test generation method proceeds as follows.

For each output cone S_c of S ,

1. Make an SDF list F of S_c .
2. Construct the topology graph G of S_c .
3. Create the TEG E of G .
4. Construct the TEM $C_E(S_c)$ of S_c based on E .

For each SDF $f \in F$,

- (a) For $C_E(S_c)$, obtain the set of SDFs corresponding to f , and generate a two-pattern test t_e for an SDF f_e in the set by using a combinational ATPG³.
- (b) Transform t_e into a test sequence T for f in S_c by using the sequence transformation.
- (c) Transform T into a test sequence T' for f in S .

As mentioned previously, a TEG of an acyclic sequential circuit is unique if the circuit is a single-output one. Therefore, in Step 3, E is also unique. Since we use a slow-fast-slow testing strategy in test application, a sequential circuit

³If all the SDFs corresponding to f are identified as untestable faults by a combinational ATPG, f is also untestable.

can be considered delay fault-free except in applying a fast clock. This implies that it is sufficient to generate a two-pattern test for at least one SDF in Step 4(a). In Step 4(c), T is always transformed into T' by applying T to the primary inputs of S corresponding to the primary inputs of S_c . Note that, for the other primary inputs of S , *don't care* values are assigned, i.e., each *don't care* value of T' is placed by 0 or 1.

3.4.2 Correctness of Our Method

In this subsection, we demonstrate the correctness of our test generation method.

Lemma 1 Let S be a single-output acyclic sequential circuit, and let $G = (V, A, w)$ be the topology graph of S . Let $E = (V_E, A_E, t, l)$ be the TEG of G . If S is a sequential circuit with DR-structure, S satisfies the following condition.

$$|t(u) - t(v)| \neq 1 \quad (\forall u, v \in V_E \text{ s.t. } l(u) = l(v))$$

□

(Proof) It will be demonstrated that if there exist u and v such that $l(u) = l(v)$ and $|t(u) - t(v)| = 1$, S is not DR-structure, i.e., the contraposition of Lemma 1.

Suppose that u and v satisfy $l(u) = l(v)$ and $|t(u) - t(v)| = 1$ ($t(u) = t$ and $t(v) = t + 1$). Since S has only one primary output, u and v share some vertex $w \in V_E$ ($t(w) = t'$) on paths to the vertex corresponding to the primary output. By C1 of Definition 4, $l(w) \in V$ exists. Furthermore, by C2 and C3 of Definition 4, there exist two paths, p, p' , from $l(u) = l(v)$ to $l(w)$ such that the number of FF on p , $n(p)$, is $t(w) - t(u) = t' - t$ and that on p' is $t(w) - t(v) = t' - t - 1$. Hence, S is not DR-structure because $n(p) - n(p')$ is one, and the proof is complete. □

Lemma 1 guarantees that a two-pattern test t_e is transformed into a test sequence $\tau(t_e) = T$ without conflict of patterns in Step 4 (b) of our test generation method. Notice that, from Lemma 1, if structure of a sequential circuit is not DR-structure but acyclic structure, conflict of patterns must occur in the sequence transformation. Hence, test generation for such a sequential circuit must be performed by using a sequential ATPG.

Lemma 2 Let S^{DR} be a sequential circuit with DR-structure, and let f be any SDF in S^{DR} . If f is testable, there exists a test sequence formed as a two-pattern sequence. \square

(Proof) Let $G = (V, A, w)$ be the topology graph of S^{DR} . Let $E = (V_E, A_E, t, l)$ and $C_E(S^{DR})$ be a TEG of G and the TEM of S based on E , respectively. Let C be the combinational circuit composed of all the CLBs on s with f , and let t_{\min} be the minimum value of labels assigned to vertices in E . Since S^{DR} has no cycle, the response for the value of any primary input will be appeared at a primary output after at most time d , where d is the sequential depth of S^{DR} . Therefore, if f is testable, there exists a test sequence T whose length is less than or equal to $d + 2$. From Definition 2, there exists an input vector pair (v_1, v_2) to C such that the desired transition is launched at the starting point of s , and the transition is propagated to the ending point of s along s . Let v_{PI} be a primary input that is reachable to C . Then, from C1 of Definition 4, a set of primary inputs, $\{u_{PI} | u_{PI} \in l^{-1}(v_{PI})\}$, exists in $C_E(S^{DR})$. Therefore, from C3 of Definition 4, the values to justify v_1 and v_2 must be applied to v_{PI} at times $t(u_{PI}) - t_{\min}$ and $t(u_{PI}) - t_{\min} + 1$, respectively. Let $v_{PO} \in V$ be a primary output at which the fault effect is observed, and let v'_{PI} be a primary input that is reachable to v_{PO} . Then, by the same reason as the above, a set of primary inputs, $\{u'_{PI} | u'_{PI} \in l^{-1}(v'_{PI})\}$, exists in $C_E(S^{DR})$, and the value needed to propagate the fault effect must be applied to v'_{PI} at time $t(u'_{PI}) - t_{\min} + 1$. For each unspecified part of T , 0 or 1 can be assigned. Hence, T is formed as a two-pattern sequence. \square

Lemma 3 Let S^{DR} be a single-output sequential circuit with DR-structure, and let $G = (V, A, w)$ be the topology graph of S^{DR} . Let $E = (V_E, A_E, t, l)$ be the TEG of G , and let $C_E(S^{DR})$ be the TEM of S based on E . Let t_{\min} be the minimum value of labels assigned to vertices in E , and let d be the sequential depth of S^{DR} . Let $I_C = (v_1, v_2)$ be an input vector pair to S^{DR} , and let $\tau(I_C)$ be the two-pattern sequence. Then, the value O_u observed from the primary output $u \in V_E$ by applying v_2 to $C_E(S^{DR})$ is equal to the value $O_{l(u)}(t(u) - t_{\min} + 1)$ observed from the primary output $l(u) \in V$ at time $t(u) - t_{\min} + 1$ by applying $\tau(I_C)$ to S^{DR} . \square

(Proof) Let $I_{u'} = (v_1^{u'}, v_2^{u'})$ be an input vector pair to a primary input $u' \in V_E$. The second vector $v_2^{u'}$ is transformed into an input pattern $I_{l(u')}(t(u') - t_{\min} + 1)$ to the primary input $l(u') \in V$ at time $t(u') - t_{\min} + 1$ by using the sequence transformation τ . From Lemma 1 and C1 of Definition 4, the input pattern applied to $l(u')$ at time $t(u') - t_{\min} + 1$ is unique. Let p be the path in S^{DR} corresponding to a path, p' , from u' to u , and let $n(p)$ be the number of FF on p . Then, the response for $I_{l(u')}(t(u') - t_{\min} + 1)$ will be appear at $l(u)$ after time $n(p)$. By C3 of Definition 4, $(t(u') - t_{\min} + 1) + n(p) = t(u) - t_{\min} + 1$ is satisfied. Furthermore, by C2 of Definition 4, the logic function of CLBs on p and that of CLBs on p' are identical. Hence, $O_u = O_{l(u)}(t(u) - t_{\min} + 1)$ is satisfied. \square

From Lemma 1–3, we can have the following theorem.

Theorem 1 Let S^{DR} be a single-output sequential circuit with DR-structure, and let $G = (V, A, w)$ be the topology graph of S^{DR} . Let $E = (V_E, A_E, t, l)$ be the TEG of G , and let $C_E(S^{DR})$ be the TEM of S^{DR} based on E . Let F be the set of all SDFs in S^{DR} . Then,

1. an SDF $f \in F$ is testable if and only if at least one SDF $f_e \in \sigma(f)$ is testable, and
2. a two-pattern test for the SDF $f_e \in \sigma(f)$ can be transformed into a test sequence for the SDF $f = \sigma^{-1}(f_e)$.

(Proof) Let S_f^{DR} be the faulty circuit with f on a segment s of S^{DR} , and let $C_{E_{f_e}}(S^{DR})$ be the faulty circuit with f_e of $C_E(S^{DR})$. Let C be the combinational circuit composed of all the CLBs on s , and let t_{\min} be the minimum value of labels assigned to vertices in E . Let d be the sequential depth of S^{DR} , and let τ^{-1} be the inverse transformation of τ .

First, we show that if f is testable, at least one f_e is also testable. From Lemma 2, there exists a two-pattern sequence T_f if f is testable. From Definition 2, if f is testable, T_f must justify input patterns v_1 and v_2 to C at time i and $i + 1$, respectively. Let C' be the combinational circuit composed of CLBs such that $t(c) = i + t_{\min}$, where c is a CLB in $\mu(C)$. From Definition 4 and Lemma 3, if we apply $\tau^{-1}(T_f)$ to $C_{E_{f_e}}(S^{DR})$, (v_1, v_2) is justified to C' . From Definition 4,

since the logic function of the combinational circuit on s with f and that on the corresponding segment s_e with f_e are identical, the value appeared from the ending point of s_e by applying the 2nd vector of $\tau^{-1}(T_f)$ to $C_{E_{f_e}}(S^{DR})$ is equal to the value appeared from the ending point of s at time $i + 1$ by applying T_f to S_f^{DR} . From the above discussion and Lemma 3, in a slow-fast-slow testing strategy, the value observed from a primary output $u \in V_E$ by applying the 2nd vector of $\tau^{-1}(T_f)$ to $C_{E_{f_e}}(S^{DR})$ is equal to the value observed from the primary output $l(u) \in V$ at time $t(u) - t_{\min} + 1$ by applying T_f to S_f^{DR} . $C_{E_{f_e}}(S^{DR})$ and the TEM $C_E(S_f^{DR})$ of S_f^{DR} based on E are isomorphic because f_e is an SDF on s_e corresponding to s . Therefore, for the 2nd vector of $\tau^{-1}(T_f)$, the output response of $C_E(S_f^{DR})$ is different from that of $C_E(S^{DR})$. Hence, if f is testable, at least one $f_e \in \sigma(f)$ is also testable.

Next, we show that if f_e is testable, $f = \sigma^{-1}(f_e)$ is also testable. If f_e is testable, there exists a two-pattern test t_{f_e} . Let s' and $C_{s'}$ be the segment with f_e and the combinational circuit composed of all the CLBs on s' , respectively. Let $t_{s'}$ be the label of CLBs in $C_{s'}$, and let (v'_1, v'_2) be a vector pair to the input of $C_{s'}$. From Definition 4 and Lemma 3, we can justify (v'_1, v'_2) to the input of $\mu^{-1}(C_{s'})$ in S_f^{DR} by applying $\tau(t_{f_e})$ to S_f^{DR} . From Definition 4, since the logic function of the combinational circuit on s' and that on the corresponding segment s'_e are identical, the value appeared from the ending point of s' by applying the 2nd vector of t_{f_e} to $C_{E_{f_e}}(S^{DR})$ is equal to the value appeared from the ending point of s'_e by applying $\tau(t_{f_e})$ to S_f^{DR} at time $t_{s'} - t_{\min} + 1$. From the above discussion and Lemma 3, the value observed from a primary output $u' \in V_E$ by applying the 2nd vector of t_{f_e} to $C_{E_{f_e}}(S^{DR})$ is equal to the value observed from the primary output $l(u') \in V$ at time $t(u') - t_{\min} + 1$ by applying $\tau(t_{f_e})$ to S_f^{DR} in a slow-fast-slow testing strategy. By the same reason as previously, $C_{E_{f_e}}(S^{DR})$ and the TEM $C_E(S_f^{DR})$ of S_f^{DR} based on E are isomorphic. Therefore, for $\tau(t_{f_e})$, the output response of S^{DR} and that of S_f^{DR} are different. Hence, if f_e is testable, $f = \sigma^{-1}(f_e)$ is also testable.

Finally, from Lemma 1, any two-pattern test for f_e can be always transformed into a test sequence for $f = \sigma^{-1}(f_e)$ by using the sequence transformation τ . Thus, the theorem is proved. \square

From this theorem and the contraposition of condition 1 in the theorem, we can see that our test generation method can not only generate test sequences for all the testable SDFs in sequential circuits with DR-structure, but also identify all the untestable SDFs in the circuits. Note that Theorem 1 still holds for both the path delay fault model and the transition fault model because the segment delay fault model can represent the both models.

3.5. Design for Testability

3.5.1 Partially Enhanced Scan Design

In order to apply our test generation method to general sequential circuits, we use the following partially enhanced scan technique:

1. For a given sequential circuit, FFs to be replaced by ESFFs (Figure 3.6) [2] are selected such that the remaining part of the circuit, called *kernel*, become DR-structure if those FFs are removed.
2. The FFs selected in the previous step are replaced by ESFFs.

Given a sequential circuit S , the overall flow of test generation for S is performed as Figure 3.7. In this figure, each step works as follows.

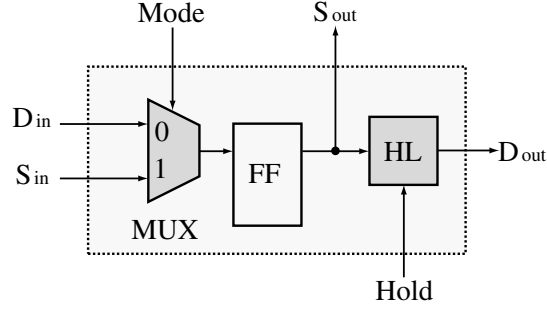
Step 1: Extract a sequential circuit with DR-structure, S^{DR} , from S as a kernel circuit.

Step 2: Transform S^{DR} into its time-expansion model $C_E(S^{DR})$.

Step 3: Generate two-pattern tests for $C_E(S^{DR})$ by using a combinational ATPG.

Step 4: Transform the two-pattern tests generated in Step (3) into test sequences for S^{DR} .

Step 5: Transform the test sequences constructed in Step (4) into test sequences for S .



D_{in} : data input	MUX: multiplexer
D_{out} : data output	Mode: normal/test signal
S_{in} : scan input	HL: hold latch
S_{out} : scan output	Hold: hold/transparence signal

Figure 3.6. Enhanced scan flip-flop (ESFF).

3.5.2 Test Application

In this subsection, we discuss test application for a sequential circuit S designed by the above partially enhanced scan technique. Let T and d_{DR} be a set of two-pattern tests for $C_E(S^{DR})$ and the sequential depth of S^{DR} , respectively. Test sequences for S^{DR} are obtained by using the sequence transformation, and the length of the test sequences is $|T| \cdot (d_{DR} + 2)$. Therefore, the test application time for the original circuit S can be estimated as follows:

$$|T| \cdot (d_{DR} + 2)(n_{ESFF} + 1) + n_{ESFF} \quad (\text{CC: clock cycles}), \quad (3.1)$$

where n_{ESFF} is the number of ESFFs. Note that we assume that a single scan chain is used to apply test sequences, and, during applying the scan clock to ESFFs, we must disable the system clock for normal FFs. In test application, since a slow-fast-slow testing strategy is used, we need to determine when the rated clock is applied to the circuit. Here, we consider it. In $C_E(S^{DR})$, let F be the set of all the SDFs detected by a two-pattern test $t \in T$. Let l and l_{\min} be the label of CLBs in which some $f \in F$ exists and the minimum value of labels, respectively. When we apply the test sequence $\tau(t)$ obtained by the sequence transformation τ to the circuit, the rated clock is used in applying $(l - l_{\min} + 2)$ th-pattern of $\tau(t)$. The other patterns are applied at a slow clock

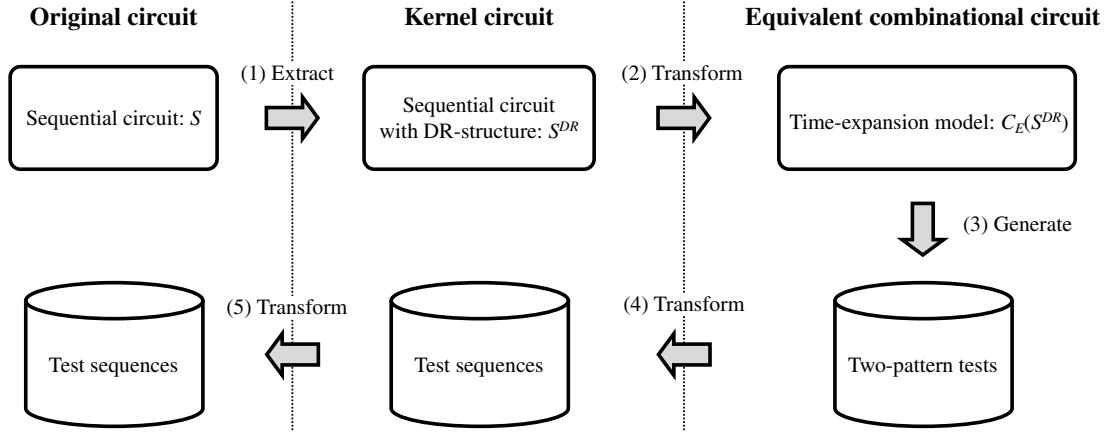


Figure 3.7. Overall flow of test generation for a general sequential circuit.

speed. Note that l can take $l_{\min}, l_{\min} + 1, \dots, l_{\min} + d_{DR}$. Eq. (3.1) is estimated under the assumption that the values of all labels associated with faults in F are identical. If the values are not identical, in the worst case, we need to apply $\tau(t)$ to the circuit $(d_{DR} + 1)$ times. In this case, therefore, the test application time will be as follows:

$$|T| \cdot (d_{DR} + 2)(n_{\text{ESFF}} + 1)(d_{DR} + 1) + n_{\text{ESFF}} \quad (\text{CC}). \quad (3.2)$$

3.6. Evaluation of Our Test Generation Method

3.6.1 Characteristics of This Work and Prior Works

From Definition 8, we can see that the relation among the three classes is as follows: {the class of acyclic sequential circuits} \supset {the class of sequential circuits with DR-structure} \supset {the class of balanced sequential circuits} (Figure 3.8). In general, a sequential circuit is classified as none of these circuit structures. Therefore, if we generate test sequences for delay faults in such a sequential circuit by using the method [5], [25] or our method, we need to extract respective circuit structures by using DFT techniques, e.g., partially enhanced scan techniques. In the following discussion, we suppose that partially enhanced scan techniques are

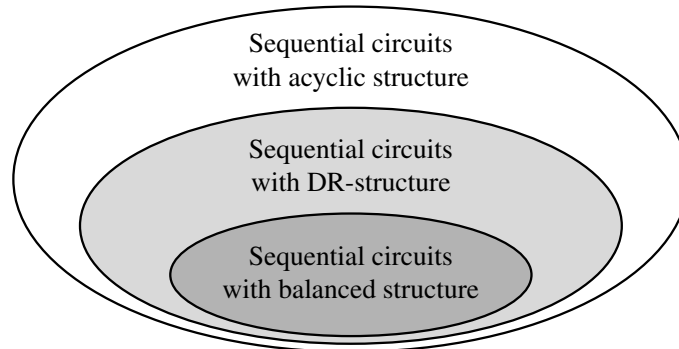


Figure 3.8. Relation among three circuit structures.

used to extract respective circuit structures.

Here, we discuss test generation complexity for each class of sequential circuits and area overhead (the number of ESFFs) required for extracting each structure.

Acyclic structure: The area overhead for making a general sequential circuit acyclic is lowest among the three structures. However, given an acyclic sequential circuit, the test generation is more complex than the others because a sequential ATPG is required for generating test sequences.

Balanced structure: In the test generation method [25], given a balanced sequential circuit, test sequences for delay faults in the circuit are generated by applying a combinational ATPG to its combinationally equivalent circuit. The combinationally equivalent circuit is obtained by just replacing each FF with a wire, and the sizes of the original circuit and the transformed circuit are equal except for FFs. Therefore, the test generation is much easier than the ordinary test generation using a sequential ATPG. However, the area overhead is highest among the three structures.

DR-structure: The area overhead is lower than that of balanced structure. Furthermore, we can also generate test sequences for delay faults in a sequential circuit with DR-structure by applying a combinational ATPG to its time-expansion model. Therefore, the test generation can be much easier than the

ordinary test generation using a sequential ATPG.

In the next subsection, we evaluate the effectiveness of our test generation method.

3.6.2 Experimental Results

Here, we evaluate the effectiveness of the proposed method in area overhead required for extracting DR-structure, test generation time, fault efficiency and test application time. The following experiments were performed on a Sun Fire 6800 workstation, and TetraMAX ATPG (Synopsys) was used as a combinational/sequential delay test generation tool. We considered a fault model in test generation as the transition fault model. The difference between test generation for the transition fault model and that for the other fault models (the path delay fault model and segment delay fault one) is only the number of mandatory assignments in propagating a desired transition along a faulty site. Therefore, test generation result for the transition fault model would be similar to that for the other fault models.

First, we compare area overheads required for extracting acyclic structure, DR-structure and balanced structure from a sequential circuit. In this experiment, we designed three data path circuits, which are shown in Table 3.2, at register transfer level, and the three circuits were synthesized by Design Compiler (Synopsys). Our method was applied to their gate-level circuits. Columns “#PIs”, “#POs” and “#FFs” denote the number of primary inputs, primary outputs and FFs, respectively. Column “Area” represents circuit size. The size was estimated by Design Compiler, and the value of “Area” was calculated by considering the area of a 2-input NAND gate to be 2. Table 3.3 shows area overheads required for extracting respective structures. Columns “Acyclic”, “DR” and “Balanced” correspond to the results in the cases of acyclic structure, DR-structure and balanced structure, respectively. For reference, the result of the fully enhanced scan method, denoted as “Combinational”, is shown in Table 3.3. Columns “#ESFFs”, “Scan [%]”, and “Area OH [%]” denote the number of ESFFs, the percentage of ESFFs and area overhead respectively, where the area of DFT elements in each ESFF is considered to be 14. Note that we obtained each sequential circuit with acyclic structure, S^A , by an exact algorithm [8], and

each sequential circuit with DR-structure, S^{DR} , and each sequential circuit with balanced structure, S^B , were extracted by applying a greedy algorithm to S^A . Here, let us explain the greedy algorithm for S^{DR} briefly. The greedy algorithm traverses S^A from the primary inputs to the primary outputs in a depth-first fashion. In traversing S^A , if paths of S^A do not satisfy the condition of Definition 8, an FF on the paths is replaced by an ESFF in order for the paths to satisfy the condition. Thus, we obtained S^{DR} from S^A . S^B was obtained in a similar way. In Table 3.3, “Area OH” of S^{DR} was larger than that of S^A . However, “Area OH” of S^{DR} achieved low area overhead compared to that of S^B . From this result, we can see that DR-structure can be obtained from a sequential circuit by paying low area overhead compared to balanced structure.

Next, we evaluate our test generation method. In order to evaluate our method, we first carried out the following experiment: we compared test generation for S^{DR} using a sequential ATPG with that using a combinational ATPG. Table 3.4 shows the test generation results in the cases of using a sequential ATPG (“Sequential ATPG”) and a combinational ATPG (“Combinational ATPG”). Columns “TGT [s]” and “FE[%]” denote test generation time and fault efficiency, respectively. In this experiment, “Combinational ATPG” achieved significantly short test generation time (about 1/93 shorter on average) under high fault efficiency compared to that of “Sequential ATPG”. Table 3.5 shows the results of test application time, respectively. Columns “Depth”, “#Vec” and “TAT [CC]” list the sequential depth of a circuit, the length of test sequences or the number of two-pattern tests, and test application time. In “Sequential ATPG”, test application time was estimated as follows. Generally, in a slow-fast-slow testing strategy, test sequences whose length is l that are generated by a sequential ATPG should be applied to the circuit $(l - 1)$ times because, in each turn, only one pattern of the test sequences will be applied at the rated clock speed while the other patterns will be applied at a slow clock speed [2]. However, in a sequential circuit with no loop, test sequences are only needed to apply $(d + 1)$ times, where d is the sequential depth of the circuit. Hence, the test generation time will be $l(n_{\text{ESFF}} + 1)(d + 1) + n_{\text{ESFF}}$. In “Combinational ATPG”, test generation time was calculated by using the equations shown in Section 3.5.2. From the above results, we can see that, for sequential circuits with DR-structure, test generation using

a combinational ATPG is effective compared with conventional test generation using a sequential ATPG.

Finally, we evaluate test generation time, fault efficiency and test application time for S^A , S^{DR} and S^B . In Table 3.6, column “Acyclic” denotes the test generation result using a sequential ATPG for S^A , and column “DR” denotes the result using a combinational ATPG for the time-expansion model of S^{DR} . Column “Balanced” denotes the result using a combinational ATPG for the combinationally equivalent circuit of S^B [25]. For reference, the result of the fully enhanced scan method, denoted as “Combinational”, is shown in Table 3.6. In this experiment, our method achieved high fault efficiency with very short test generation time compared to that of “Acyclic”. Moreover, we obtained almost the same fault efficiency as “Balanced” with a similar test generation time. Thus, our method can significantly improve test generation time and fault efficiency by paying large area overhead compared to acyclic structure. Table 3.7 shows the results of test application time. In “Combinational”, test application time was estimated as $2n_{\text{TPT}}(n_{\text{ESFF}} + 1) + n_{\text{ESFF}}$, where n_{TPT} is the number of two-pattern tests. In general, test application time depends on the number of ESFFs. If the number of ESFFs decreases, test application time will also be reduced. The tendency of test application time in Table 3.7 was not similar to the general tendency. Note that, in our method, test application time can be reduced if a test compaction technique (e.g., [17]) is used.

From the above results, we can see that our method is effective in hardware overhead, test generation time and fault efficiency.

3.7. Summary

This chapter presented a new structure, called discontinuous reconvergence structure (DR-structure), of sequential circuits with easy testability for delay faults. We proposed a delay test generation method for sequential circuits with the structure. In our method, instead of a sequential ATPG, a combinational ATPG is used to generate test sequences for delay faults. We theoretically proved the correctness of the proposed method. Our method can handle several delay fault models in which faults can be detected by two-pattern tests, e.g., the path delay

fault model, segment delay fault model and transition fault model. We confirmed that our test generation method can reduce test generation time and can enhance fault efficiency compared to the ordinary test generation method using a sequential ATPG. To apply our method to general sequential circuits, we used a partially enhanced scan technique. Theoretically, the class of sequential circuits with DR-structure properly includes that of balanced sequential circuits. Therefore, it is conceivable that DR-structure is extracted from a sequential circuit with low area overhead compared with balanced structure). We also confirmed it experimentally.

Table 3.2. Circuit characteristics.

Circuit name	#PIs	#POs	#FFs	Area
C1	16	24	80	5,528
C2	24	32	112	6,151
C3	128	96	288	20,239

Table 3.3. Percentages of enhanced scan FFs and area overheads.

Circuit name	Acyclic			DR			Balanced			Combinational		
	#ESFFs	Scan [%]	Area OH [%]	#ESFFs	Scan [%]	Area OH [%]	#ESFFs	Scan [%]	Area OH [%]	#ESFFs	Scan [%]	Area OH [%]
C1	24	30.0	6.1	32	40.0	8.1	48	60.0	12.1	80	100.0	20.3
C2	24	21.4	5.5	32	28.6	7.3	48	42.9	10.9	112	100.0	25.5
C3	128	44.4	8.9	160	55.6	11.1	192	66.7	13.3	288	100.0	19.9

Table 3.4. Test generation results for sequential circuits with DR-structure.

Circuit name	Sequential ATPG		Combinational ATPG	
	TGT [s]	FE [%]	TGT [s]	FE [%]
C1	273.2	98.08	3.8	100.00
C2	2,664.4	97.37	24.0	98.57
C3	2,455.7	99.94	25.4	99.96

Table 3.5. Test application time for sequential circuits with DR-structure.

Circuit name	Depth	Sequential ATPG		Combinational ATPG	
		#Vec	TAT(CC)	#Vec	TAT(CC) Eq. (3.1) Eq. (3.2)
C1	4	1,082	178,562	233	46,166 230,702
C2	3	695	91,772	178	29,402 117,512
C3	2	974	470,602	384	247,456 742,048

Table 3.6. Test generation results for respective methods.

Circuit name	Acyclic (Sequential ATPG)		DR (Combinational ATPG)		Balanced (Combinational ATPG)		Combinational (Combinational ATPG)	
	TGT [s]	FE [%]	TGT [s]	FE [%]	TGT [s]	FE [%]	TGT [s]	FE [%]
C1	1,171.5	98.08	3.8	100.00	7.1	99.93	2.9	99.94
C2	17,934.9	89.33	24.0	98.57	13.5	99.12	2.1	99.97
C3	36,570.0	97.72	25.4	99.96	28.1	99.95	22.2	99.98

Table 3.7. Test application time for respective methods.

Circuit name	Acyclic (Sequential ATPG)			DR (Combinational ATPG)			Balanced (Combinational ATPG)			Combinational (Combinational ATPG)		
	#Vec	Depth	TAT(CC)	#Vec	Depth	TAT(CC) Eq. (3.1) Eq. (3.2)	#Vec	Depth	TAT(CC) Eq. (3.1) Eq. (3.2)	#Vec	Depth	TAT(CC)
C1	707	4	88,399	233	4	46,166 230,702	206	3	50,518 201,928	195	0	31,670
C2	364	3	36,424	178	3	29,402 117,512	175	3	42,923 171,548	217	0	49,154
C3	482	2	186,662	384	2	247,456 742,048	391	2	302,044 905,748	386	0	223,396

Chapter 4

Non-Scan Delay Testing of Sequential Circuits Using State Transition Information

4.1. Introduction

As the speed of modern VLSI circuits increases, delay testing is becoming essential to guarantee the timing correctness of the circuits. Delay test generation for such circuits is a challenging problem. This is because there exist many sequentially untestable delay faults in a circuit [7], and the task of identifying those faults is very time-consuming. It is virtually impossible to identify all the untestable faults in a large circuit. To facilitate delay test generation, standard scan methods [27, 28] and enhanced scan ones [7, 11] have been proposed. Given a sequential circuit, these design methods make most or all of the sequentially untestable faults detectable by making every flip-flop (FF) controllable and observable. As a result, the test generation time is significantly reduced and the fault coverage becomes higher. However, in scan-based delay testing, the test application time becomes longer because of the scan-shift operation. In addition, the scan-shift operation is generally performed at a low clock speed while the second vectors of two-pattern tests are launched at a rated clock speed. This situation may cause the IR-drop [29] because the operating speed rapidly changes, and it makes apparent circuit delay increase temporarily. In consequence, the test may detect

temporary delay faults and it poses over-testing. Therefore, it is desirable that the operating speed is constant during test application.

In the recent decade, design for testability (DFT) methods at register transfer (RT) level have been proposed [22]. An RT-level circuit is generally composed of a controller, represented by a state transition graph (STG), and a data path, represented by hardware elements such as registers, multiplexers (MUXs) and operational modules. For delay faults, a non-scan DFT method of data paths, which overcomes the drawbacks of scan-based testing, has been proposed [1]. On the other hand, a DFT method for stuck-at faults in controllers has been proposed [24]. This method is also non-scan based, and achieves complete fault efficiency, short test application time and at-speed testing. In this method, the above merits are realized by utilizing a given STG and by appending an extra logic, called an *invalid test state generator (ISG)*, to the original controller.

This chapter proposes a non-scan design scheme, which is an extension of one in [24], to enhance delay testability of controllers. In this scheme, we utilize a given STG to test delay faults in its synthesized controller. The original behavior of the STG is used during test application. For faults that cannot be detected by using the original behavior, we append an extra logic, called an *invalid test state and transition generator (ISTG)*, to the original controller. In this dissertation, we discuss a classification of untestable faults in a controller, and show our DFT flow based on the classification. Our scheme allows achieving short test application time and at-speed testing, which is always performed at a constant clock speed. By some experiments, we show the effectiveness of our method.

4.2. Preliminaries

4.2.1 Target Circuit and Fault Model

Our target circuit is a controller represented by an STG, and we target delay faults that can be tested by two-pattern tests (e.g., transition faults, path delay faults, etc.) in the circuit. In the following discussion, we focus on the transition fault model for simplicity. Figure 4.1 shows an example of a controller represented by an STG. In this chapter, we assume that a gate-level implementation

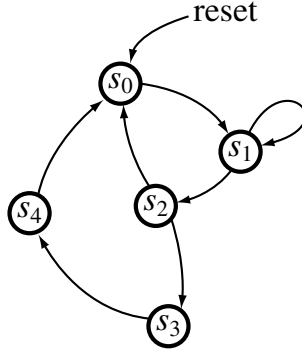


Figure 4.1. State transition graph representing a controller.

of a controller is given, and the controller has a reset signal, i.e., we can make a transition from any state to the reset state by activating the reset signal. Figure 4.2 represents a sequential circuit which can be obtained by synthesizing a given STG. We also assume that, for a given controller, the mapping information between each state in the STG and the value of the state register (SR) (state encoding information) is available.

4.2.2 Terminologies

Here, we define several terminologies. For any value of the SR in a sequential circuit synthesized from a given STG, the state corresponding to the value is called a *valid state* if it is reachable from the reset state in the STG. Otherwise, it is called an *invalid state*. For a synthesized controller, a combinational circuit extracted from the controller by replacing the SR with pseudo primary inputs (PPIs) and pseudo primary outputs (PPOs) is called a *combinational test generation model (CTGM)* (Figure 4.3). Every two-pattern test for a CTGM, (V_1, V_2) , can be denoted as $(I_1 \& S_1, I_2 \& S_2)$, where I_1 and I_2 are the values of primary inputs (PIs), S_1 and S_2 are the values of PPIs, and “&” is the concatenation operator. A two-pattern test for a CTGM, $(I_1 \& S_1, I_2 \& S_2)$, is said to be a *valid two-pattern test* if there exists an arc (transition) (I, P, N, O) in a given STG such that $I = I_1$, $P = S_1$ and $N = S_2$, where I is an input value, P is a present state value, N is a next state value, and O is an output value. Otherwise, it is called

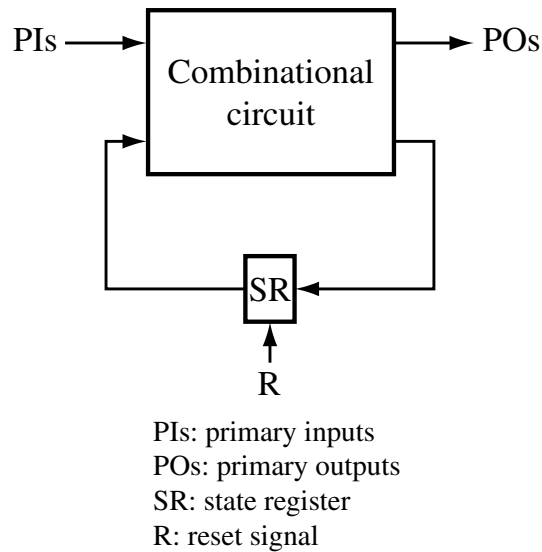


Figure 4.2. Synthesized controller.

an *invalid two-pattern test*. The transition corresponding to a valid two-pattern test (resp. an invalid two-pattern test) is called a *valid test transition* (resp. an *invalid test transition*). Valid test transitions and invalid ones are collectively called *test transitions*. For each state included in a test transition, the state is called a *valid test state* (resp. an *invalid test state*) if it is a valid state (resp. an invalid state). Also, valid test states and invalid ones are collectively called *test states*.

Figure 4.4 shows an example of test states and test transitions. When two-pattern tests are generated for the CTGM (Figure 4.3) of Figure 4.1, the test transitions corresponding to the generated two-pattern tests can be classified into five types:

- valid test transition (Figure 4.4(1)),
- invalid test transition from a valid test state to a valid test state (Figure 4.4(2)),
- invalid test transition from a valid test state to an invalid test state (Figure 4.4(3)),

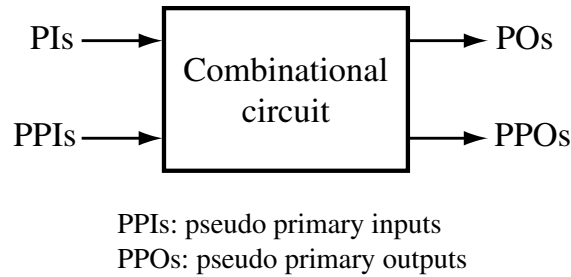


Figure 4.3. Combinational test generation model (CTGM).

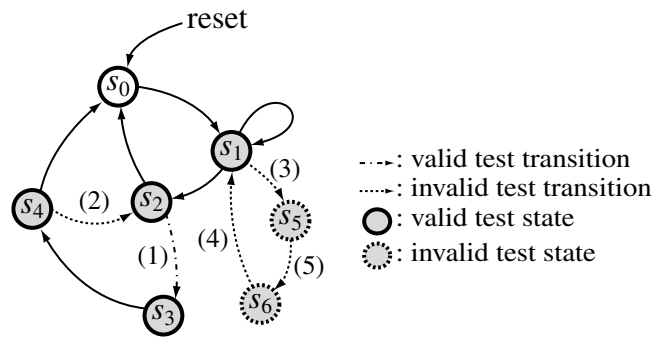


Figure 4.4. Test states and transitions.

- invalid test transition from an invalid test state to a valid test state (Figure 4.4(4)) and
- invalid test transition from an invalid test state to an invalid test state (Figure 4.4(5)).

4.3. Proposed Method

4.3.1 Test Architecture

In our testing scheme, the original behavior of a given STG is used during test application, i.e., valid two-pattern tests are applied by using the original behavior.

Faults that cannot be detected by using the original behavior are tested by an extra logic, called an *invalid test state and transition generator (ISTG)*. Our test architecture is shown in Figure 4.5, which is an extension of the test architecture [24]. In Figure 4.5, the respective DFT elements play the following roles.

- The ISTG is used to generate invalid test states and invalid test transitions.
- The extra pin of t_{mode} is used to select between the normal mode and the test mode.
- The extra pins of t_{out} are used to observe the value of the SR. The bit width of t_{out} is the same as that of the SR.
- The extra pins of t_{sel} are used to distinguish among invalid two-pattern tests¹.
- The MUX is used to switch between the signal from the combinational part of the controller and that from the ISTG.

The differences between our test architecture and the previous one [24] are as follows. Unlike the ISG of the previous test architecture, the ISTG of our test architecture is used to generate not only invalid test states needed to apply the first vectors of invalid two-pattern tests but also invalid test transitions corresponding to the invalid two-pattern tests. Notice that the ISG has no t_{sel} . Furthermore, in our test architecture, t_{out} is appended to the output side of the SR while t_{out} of the previous test architecture is appended to the input side of the SR. In consequence, our test architecture can test delay faults appropriately because the value captured by the SR can be observed.

Our test architecture can achieve short test application time and at-speed testing because the scan-shift operation is never used. Note that we use the terminology of “at-speed test” only if test application can always be performed at a rated clock speed.

Here, we mention testing of delay faults in the ISTG. Since the ISTG is only used in the test mode, we do not need to care about its behavior in the normal mode. Therefore, to test delay faults in the ISTG, we need only to check whether

¹The details will be described in Section 4.3.3.

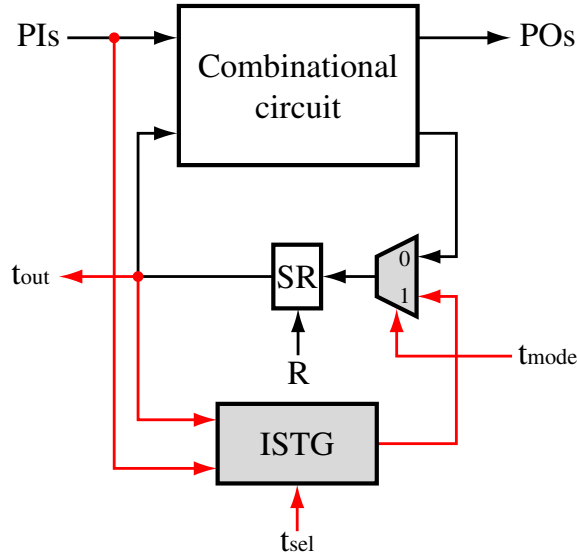


Figure 4.5. Test architecture.

the value captured by the SR is correct or not by using t_{out} in the test mode. It can be performed during testing of the controller simultaneously.

Let us consider an impact of power consumption on a controller, which is induced by the ISTG. Although the ISTG is only used during testing, it might consume power in the normal mode. If the impact is serious, we can avoid it by configuring the controller as Figure 4.6. In Figure 4.6, “AND” is used to suppress the power consumption in the ISTG during normal operation. If the value of t_{mode} is 1, “AND” supplies the values of the PIs and the SR to the ISTG. In the normal operation, the ISTG receives the constant value of zeros by setting the value of t_{mode} to 0. Note that, in the following discussion, we do not consider the power impact of an ISTG for simplicity.

4.3.2 Test Quality

In a sequential circuit, untestable delay faults generally exist. Here, we classify untestable delay faults in a controller into five categories in terms of “logic level”, “RT-level” and t_{out} . The classification is shown in Figure 4.7. Let F be a whole

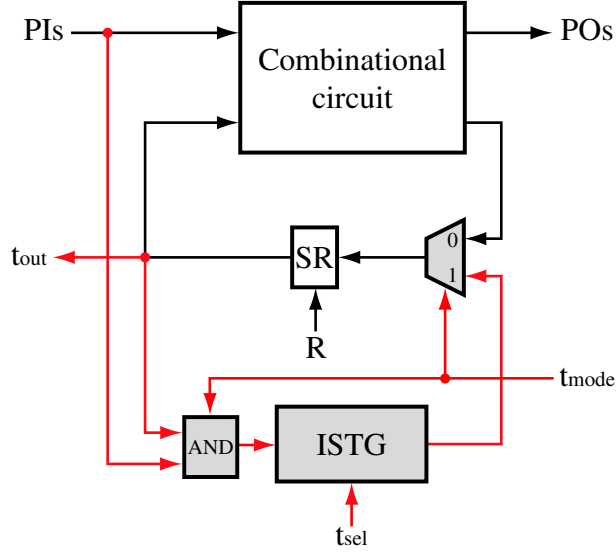
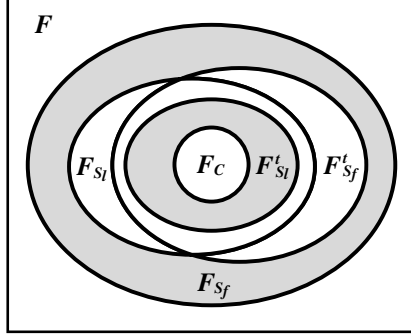


Figure 4.6. Power-aware configuration.

set of faults. All the faults in a set of F_C are untestable in the combinational part of the controller if we consider the SR as POs and PIs, i.e., any two-pattern test can be applied to the combinational part and any response from that can be observed. These faults are called combinational untestable faults. Some combinational testable faults in $\overline{F_C}$ ($= F - F_C$) are untestable because the value of the SR is restricted by the available state transitions in its synthesized controller. Such faults belong to a set of F_{S_l} ($\supset F_C$). We call these faults sequentially untestable faults at logic level without t_{out} . When a given STG is synthesized, some new states and transitions are generally implemented in the synthesized controller. This implies that some testable faults in $\overline{F_{S_l}}$ are untestable if the original behavior of the given STG is only considered. We classify these faults into a set of F_{S_f} ($\supset F_{S_l}$). These faults are called sequentially untestable faults at RT-level without t_{out} . Let us consider appending t_{out} to the synthesized controller here. Appending t_{out} makes some untestable faults in F_{S_l} and F_{S_f} testable. Thus, F_{S_l} and F_{S_f} change into sets of $F_{S_l}^t$ ($\supset F_C$) and $F_{S_f}^t$ ($\supset F_{S_l}^t$), respectively. Let M be a given STG, and M' be the STG corresponding to a sequential circuit derived by synthesizing M . In Figure 4.7, some faults in F_{S_l}



- F : a whole set of faults
- F_C : a set of combinational untestable faults
- F_{S_i} : a set of sequentially untestable faults at logic level without t_{out}
- $F_{S_i}^t$: a set of sequentially untestable faults at logic level with t_{out}
- F_{S_f} : a set of sequentially untestable faults at RT-level without t_{out}
- $F_{S_f}^t$: a set of sequentially untestable faults at RT-level with t_{out}

Figure 4.7. Classification of untestable faults.

can be activated by using the behavior of M but the effects of the faults cannot be propagated to a PO by using the behavior of M' . Such faults can be detected by using the behavior of M if t_{out} is appended to the circuit. These faults belong to $F_{S_i} - F_{S_f}^t$.

In test generation for a given sequential circuit, a sequential *test generator* (ATPG) tries to identify all the untestable faults in F_{S_i} and to generate tests for all the testable faults in $\overline{F_{S_i}}$. The goal of this task can be achieved if the ATPG has enough time to complete the task. However, it is infeasible for a large circuit. For such a circuit, DFT approaches should usually be used to facilitate test generation. In our method, t_{out} is appended to a given circuit in order to facilitate test generation. As a result, some faults in F_{S_i} are made detectable. However, the number of these faults is not so large [7]. Our method aims to detect faults in $\overline{F_{S_i}^t}$ and identify faults in $F_{S_i}^t$ as much as possible in a reasonable time. Although a fault in $F_{S_f} - F_{S_i}^t$ itself does not affect the performance of a controller under the original behavior of its STG and under the single fault assumption, we try to detect the fault. The reason is as follows. Suppose that there exist an untestable fault f' in $F_{S_f} - F_{S_i}^t$ and a testable fault f in $\overline{F_{S_f}}$ simultaneously in a circuit, and the effect of f' cannot be propagated to a PO but f' can be activated during normal operation. In this case, if f' is not tested, we cannot evaluate whether a test generated for f is invalidated by f' . This implies that

f can be missed if there are no tests that detect f in a generated test set. In order to avoid such a situation, we should test not only testable faults in $\overline{F_{S_f}}$ but also untestable faults in $F_{S_f} - F_{S_f}^t$. Under the single fault assumption, the above discussion dose not make sense. However, from a practical point of view, it is very useful because there can exist multiple faults in a circuit.

4.3.3 Flow of Our Method

Given a controller, the procedure of our method is performed as Figure 4.8. In Step 1, we try to generate valid two-pattern tests as much as possible. As a result, area overhead can be reduced. Moreover, this step can contribute to alleviating over-testing. Step 2 also can contribute to reducing area overhead and alleviating over-testing. In Steps 3 and 4, faults that cannot be detected in normal operation are taken into consideration. Step 5 can lead to achievement of short test application time.

In the following paragraphs, we explain each step of Figure 4.8 in detail.

Step 1: For the CTGM of a given controller, we use a combinational ATPG. In order to generate valid two-pattern tests, we give some information (constraint) to a combinational ATPG. A constraint is defined as a vector pair $(I_1^C \& S_1^C, I_2^C \& S_2^C)$. Each bit of a constraint can take the value of 0, 1 or *don't care* (X). When we give a constraint to a combinational ATPG, the ATPG tries to generate two-pattern tests under the constraint, i.e., for every X of the constraint, a suitable value is specified.

A constraint C is derived as follows. First, a transition $T = (I, P, N, O)$ is selected from a given STG. Then, T is used as $C = (I \& P, "Xs" \& N)$. It is obvious that two-pattern tests generated under this constraint can always be applied by using the original behavior of the controller. Thus, we can obtain valid two-pattern tests. Note that, because of the presence of a delay fault f , we may fail to justify a valid two-pattern test generated for f by using the original behavior. However, it does not matter because any error induced by f in the SR can always be observed from t_{out} . In Step 1, if we use all the constraints corresponding to the transitions in the STG, we can identify all the untestable faults in $F_{S_f}^t$, and valid two-pattern tests can be generated for all the faults in $\overline{F_{S_f}^t}$. The detected

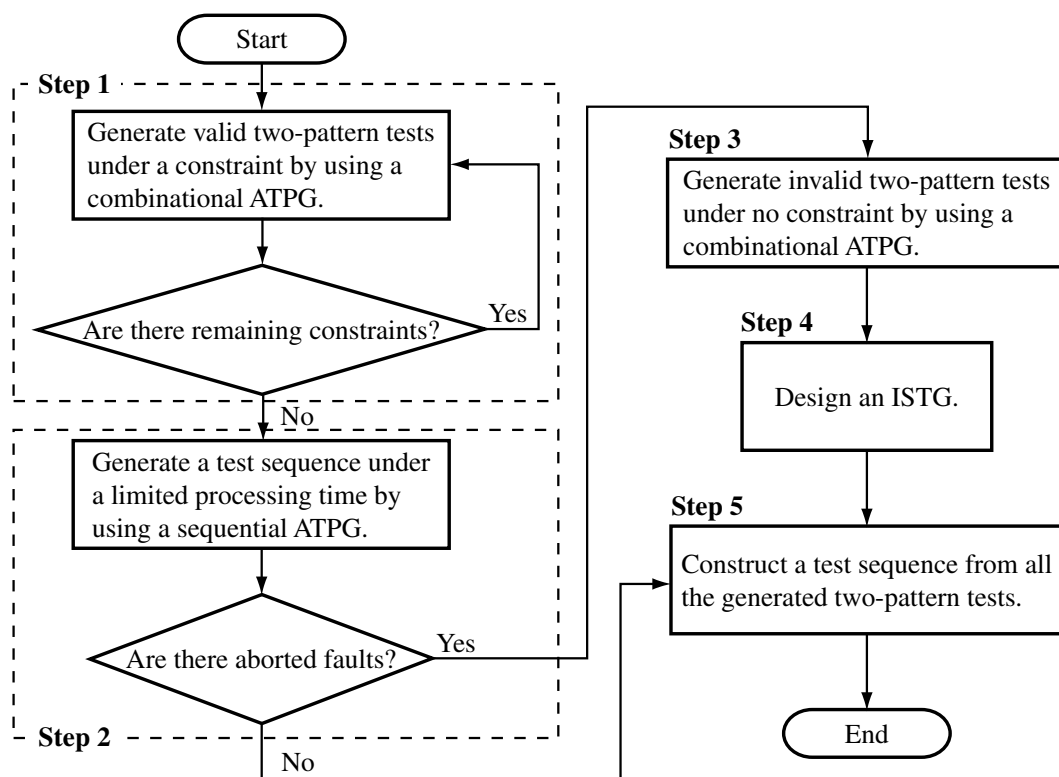


Figure 4.8. Flow chart of our method.

faults are dropped from the fault list.

Step 2: For the remaining faults in Step 1, we try to identify untestable faults in $F_{S_i}^t$ and generate a test sequence for faults in $F_{S_f}^t - F_{S_i}^t$ by applying a sequential ATPG to the controller with t_{out} . Since the circuit has t_{out} , test generation for it is easier than that for the original one. Moreover, the number of target faults in this step is much reduced compared with the total number of faults. Nevertheless, this task is very time-consuming. Therefore, we use a sequential ATPG under a limited processing time (or a limited number of backtracks) per fault. This implies that, in this step, we take into account faults that can be easily identified as untestable faults and easily detectable faults. The detected faults and the untestable faults are dropped from the fault list. Notice that if there are no

Table 4.1. Truth table of an ISTG.

Inputs	Outputs
$I_1^1 \& S_1^1$	S_2^1
$I_1^2 \& S_1^2$	S_2^2
\vdots	\vdots
$I_1^n \& S_1^n$	S_2^n

aborted faults in this step, we do not need to perform Steps 3 and 4. This means that only the pins of t_{out} are added to the original controller as a DFT element.

Step 3: We generate two-pattern tests, which are invalid, for the remaining faults in Step 2 under no constraint by using a combinational ATPG. This step can identify all the untestable faults in F_C .

Step 4: We design an ISTG to test the faults detected in Step 3 because we do not identify whether these faults belong to $F_{S_i}^t - F_C$ or not. An ISTG must realize functions to apply invalid two-pattern tests to the combinational part of the controller. Furthermore, it must also have functions that make all of the invalid test states in the invalid two-pattern tests reachable from the reset state. For example, given n invalid two-pattern tests $t_1 = (I_1^1 \& S_1^1, I_2^1 \& S_2^1)$, $t_2 = (I_1^2 \& S_1^2, I_2^2 \& S_2^2)$, \dots , $t_n = (I_1^n \& S_1^n, I_2^n \& S_2^n)$, an ISTG must realize the functions shown in the truth table (Table 4.1). Note that, in Table 4.1, if there exist m invalid two-pattern tests such that $I_1^1 \& S_1^1 = I_1^2 \& S_1^2 = \dots = I_1^m \& S_1^m$ and $S_2^i \neq S_2^j$ ($\forall i, j, 1 \leq i, j \leq m, i \neq j$), we need t_{sel} to distinguish among them. The bit width of t_{sel} is $\lceil \log m_{\text{max}} \rceil$, where m_{max} is the maximum number of two-pattern tests that satisfy the above conditions.

We touch on a problem to reduce the area of an ISTG here. If the truth table shown in Table 4.1 includes X values, i.e., X values are included in two-pattern tests, we can make use of them to reduce the area of an ISTG. This problem is considered as a type of an input encoding problem [30]. Therefore, we could apply some heuristics [30] for the input encoding problem to design an ISTG. However,

this is still an open problem. In this chapter, it is assumed that two-pattern tests generated in Step 3 do not include X values.

Step 5: In order to construct a test sequence for the original circuit, we determine an order of applying all the generated two-pattern tests. Note that the test sequence generated in Step 2 is applied to the circuit before or after applying the test sequence obtained in this step. Here, we consider a problem to construct the test sequence that has the minimum length. It is solved as an asymmetric traveling salesperson problem (ATSP) on a complete weighted directed graph represented by a distance matrix, where a vertex t corresponds to a two-pattern test, an arc (t_i, t_j) corresponds to the path between t_i and t_j , and the weight of the arc corresponds to the distance from t_i to t_j . The distance $d(t_i, t_j)$ means the minimum clock cycles that are needed to apply the first vector of t_j after applying t_i . Note that if t_j is a valid two-pattern test and the values of the second vector of t_i and the first vector of t_j are identical, the value of $d(t_i, t_j)$ is -1 . Thus, we can construct a test sequence by solving the corresponding ATSP.

For example, let us consider the ATSP for Figure 4.4. In Figure 4.4, there are five test transitions. These test transitions (1), (2), (3), (4) and (5) correspond to two-pattern tests t_1, t_2, t_3, t_4 and t_5 , respectively. Let the reaching states after applying t_1, t_2, t_3, t_4 and t_5 be S_4, S_3, S_U, S_1 and S_U respectively, where S_U denotes an unknown state. Table 4.2 shows the distance matrix for the five two-pattern tests. Note that, in this table, $d(R, t)$ (resp. $d(t, R)$) denotes the minimum distance from the reset state R (resp. S_a) to S_b (resp. R), where S_a is the reaching state after applying t , and S_b is the state in applying the first vector of t . A solution of this problem is $R \rightarrow t_1 \rightarrow t_2 \rightarrow t_4 \rightarrow t_3 \rightarrow t_5 \rightarrow R$. From this solution, t_1, t_2, t_4, t_3 and t_5 are applied in that order.

4.4. Advantages of Our Method

4.4.1 Conventional Methods and Our Method

In this subsection, we summarize the proposed method and conventional methods (standard scan and enhanced scan ones). In the following discussion, we assume

Table 4.2. Distance matrix.

	R	t_1	t_2	t_3	t_4	t_5
R	—	2	4	1	3	2
t_1	1	—	0	2	4	3
t_2	1	-1	—	2	4	3
t_3	1	3	5	—	4	3
t_4	1	1	3	0	—	1
t_5	1	3	5	2	4	—

that MUXs are used in the scan-based methods although there are some ways to implement a standard scan FF (SSFF) and an enhanced scan FF (ESFF).

Standard scan method: Test generation for a controller designed by this method requires a combinational ATPG that supports the skewed-load [27] mode and/or the broad-side [28] one. Generated two-pattern tests are applied to the controller through a scan chain in the skewed-load fashion and/or the broad-side one. The test application time is estimated as $n_{\text{TPT}}(n_{\text{SSFF}} + 2) + n_{\text{SSFF}}$, where n_{TPT} and n_{SSFF} are the number of two-pattern tests and SSFFs, respectively. In this method, each SSFF in the controller has an additional MUX. Therefore, the area overhead is $A_{\text{MUX}} \times n_{\text{SSFF}}$, where A_{MUX} is the area of the additional MUX. As a result, the delay of an MUX are added as the additional circuit delay. This method needs three additional pins. Note that we assume that this method has a single scan chain for simplicity.

Enhanced scan method: We can generate tests for a controller designed by this method by using a combinational ATPG. The test application time is estimated as $2n_{\text{TPT}}(n_{\text{ESFF}} + 1) + n_{\text{ESFF}}$, where n_{ESFF} is the number of ESFFs. Each ESFF in the controller has an additional MUX and a hold latch (HL) [2]. The area overhead is, therefore, $(A_{\text{MUX}} + A_{\text{HL}}) \times n_{\text{ESFF}}$. Note that, although the area overhead can be reduced by using some techniques (e.g., [9]), we estimate it as the above equation for simplicity. The delay penalty is higher than that of the standard scan method because of the HL. The extra circuit delay is equal to the

sum of the delays of an MUX and an HL. Furthermore, the pin overhead of this method is high compared with that of the standard scan method because the HL has to be controlled by an additional pin. The total number of additional pins is four. Note that it is also assumed that this method has a single scan chain.

Our method: In our method, we first generate tests for the combinational test generation model of a given controller by using a combinational ATPG under the constraints extracted from its STG. The test generation is repeated n_c times, where n_c is the number of constraints. Next, we generate a test sequence for the remaining faults under a limited processing time by using a sequential ATPG. Then, we try to generate two-pattern tests for the aborted faults in the previous step under no constraint. The test application time is determined by an order of applying all the generated two-pattern tests to the controller. The area overhead is $A_{\text{MUX}} \times n_{\text{FF}} + A_{\text{ISTG}}$, where n_{FF} is the number of FFs, and A_{ISTG} is the area of an ISTG. The proposed method has the same delay penalty compared to that of the scan-based methods because ISTGs are not used during normal operation. However, in order to perform at-speed testing, we need to pay attention to the maximum delay of an ISTG. The maximum delay of an ISTG depends on its structure. In the next subsection, we evaluate the maximum delays of ISTGs by experiments. We believe that the ISTG of a given controller can be constructed with small maximum delay compared to that of the original circuit by contriving ways to synthesize the ISTG. The extra pins (t_{sel} , t_{out} and t_{mode}) are needed in our method. The sum of the bit width of these pins is $|t_{\text{sel}}| + |t_{\text{out}}| + 1$. Notice that, in the proposed method, if Steps 3 and 4 are not performed, the pin overhead is $|t_{\text{out}}|$. In a controller-data path circuit, if we consider that its controller part is tested independently of its data path part, the PIs and the POs of the data path can be used as t_{sel} and t_{out} during testing of the controller, respectively. The PIs of the data path are split and connected to t_{sel} , and the POs of the data path are shared with t_{out} by using MUXs. Let n_{DPI} and n_{DPO} be the bit widths of the PIs and the POs of the data path, respectively. In the sharing of test pins, if $n_{\text{DPI}} \geq |t_{\text{sel}}|$, no additional test pins are required for t_{sel} . Otherwise, the number of additional test pins is $|t_{\text{sel}}| - n_{\text{DPI}}$. For t_{out} , if $n_{\text{DPO}} \geq |t_{\text{out}}|$, we need one additional test pin to control MUXs. Otherwise, we need to apply one two-pattern

test $\lceil |t_{\text{out}}|/n_{\text{DPO}} \rceil$ times and observe the value of t_{out} in $\lceil |t_{\text{out}}|/n_{\text{DPO}} \rceil$ batches. In this case, the number of additional test pins is $\lceil |t_{\text{out}}|/n_{\text{DPO}} \rceil$. As a result of the sharing, although the area overhead increases to $A_{\text{MUX}} \times (n_{\text{FF}} + |t_{\text{out}}|) + A_{\text{ISTG}}$, the pin overhead decreases to $(|t_{\text{sel}}| - n_{\text{DPI}}) + \lceil |t_{\text{out}}|/n_{\text{DPO}} \rceil + 1$ in the worst case. Since it is expected that $n_{\text{DPI}} \geq |t_{\text{sel}}|$ and $n_{\text{DPO}} \geq |t_{\text{out}}|$ can be satisfied for practical RT-level circuits, the pin overhead decreases to 2. It is also reduced to 1 if Steps 3 and 4 are skipped.

We mention here some differences among the three methods. In the scan-based methods, every FF is just replaced with an SSFF or an ESFF injudiciously while ISTGs of our method are designed depending on generated invalid two-pattern tests. These features could cause the following. Suppose that additional two-pattern tests are required for some reasons (e.g., for fault diagnosis) after applying the respective methods to a circuit. In the enhanced scan method, these two-pattern tests can always be applied without modification of the circuit. However, the other methods could not cope with such a situation. The advantages of our method are as follows. Since the scan-shift operation is needed in the scan-based methods, at-speed test cannot be performed, i.e., a slow clock is used except in activating delay faults. However, our method can always apply tests at a rated clock speed. In this environment, the IR-drop will be suppressed. Moreover, our method can be performed flexibly according to a trade-off between area overhead and test generation time. The trade-off is determined by the number of constraints used in Step 1 of the proposed method and by the limited processing time per fault in Step 2. In the scan-based methods, all the FFs in a circuit are modified independently of the circuit function. Consequently, many untestable delay faults, which do not need to be tested, are made detectable. This implies that yield loss may potentially occur. In our method, over-testing is also caused by t_{out} and the ISTG. However, owing to Steps 1 and 2 of the proposed procedure, our method can alleviate over-testing compared with the scan-based methods.

4.4.2 Experimental Results

To evaluate our method, the following experiments were performed on a Sun Blade 2000 workstation. We used the MCNC '91 benchmark circuits shown in Table 4.3. A reset signal was appended to every benchmark circuit. Columns

Table 4.3. Circuit characteristics.

Circuit name	#PIs	#POs	#FFs	#States	#Arcs	Area
bbsse	7	7	4	16	72	295
keyb	7	2	5	19	189	459
kirkman	12	6	4	16	446	360
planet	7	19	6	48	163	937
s298	3	6	8	218	1,314	3,662
s420	19	2	5	18	155	122
sand	11	9	5	32	216	866
scf	27	56	7	121	407	1,378

“#PIs”, “#POs”, “#FFs”, “#States” and “#Arcs” denote the number of PIs, POs, FFs, states in an STG, and transitions in an STG, respectively. Column “Area” represents circuit size. The size was estimated by Design Compiler (Synopsys), and the value of “Area” was calculated by considering the area of a 2-input NAND gate to be 2. During logic synthesis, binary encodings were used. Note that, it is assumed that each benchmark circuit has a data path, which is controlled by the benchmark circuit, and the data path has enough PIs and POs to be shared with additional test pins. In the following experiments, we compared our method (NS) to the standard scan technique (SS) and the enhanced scan technique (ES). TestGen (Synopsys) and FlexTest (Mentor Graphics) were used as a combinational ATPG and a sequential one respectively, and the transition fault model was targeted. Note that, in SS and ES, we assumed that the both methods have a single scan chain. For SS, we compared only the hardware overhead because the ATPGs do not support the skewed-load mode and the broad-side one.

First, we show the hardware overhead of our method. Columns “Area OH [%]” and “Pin OH” of Table 4.4 denote the ratio of the area of additional hardware elements to that of the original circuit if the sharing of test pins was not adopted, and the number of additional test pins, respectively. To calculate “Area OH”,

Table 4.4. Hardware overheads.

Circuit name	Area OH [%]			Pin OH		
	SS	ES	NS	SS	ES	NS
bbsse	9.5	19.0	9.8	2 (3)	3 (4)	2 (5)
keyb	7.6	15.3	7.6	2 (3)	3 (4)	1 (5)
kirkman	7.8	15.6	7.8	2 (3)	3 (4)	1 (4)
planet	4.5	9.0	15.6	2 (3)	3 (4)	2 (7)
s298	1.5	3.1	23.3	2 (3)	3 (4)	2 (10)
s420	28.7	57.4	28.7	2 (3)	3 (4)	1 (5)
sand	4.0	8.1	4.2	2 (3)	3 (4)	2 (6)
scf	3.6	7.1	6.6	2 (3)	3 (4)	2 (8)

we considered both A_{MUX} and A_{HL} described in the previous section as 7. In Table 4.4, the area overhead of SS was the smallest of all. However, for three cases, our method achieved the same area overhead as that of SS, and low area overhead compared with that of ES except two cases. Note that, as mentioned in Section 4.3.3, if we utilize X values in two-pattern tests, the area overhead can be reduced. Besides, in a controller-data path circuit, the controller is generally much smaller than the data path. Therefore, even if the area overhead of a controller is large, it is not critical in the whole circuit. In the result of pin overheads, our method required a large number of additional test pins for each circuit, which is shown in a parenthesis, if the sharing of test pins is not adopted. However, if the sharing of test pins is adopted for the respective methods, the pin overheads can be reduced as shown in Table 4.4.

Next, we show the test generation results. In this experiment, our method was performed as follows. Column “#Arcs” in Table 4.3 corresponds to the number of constraints in Step 1 of our method. We used all the constraints in Step 1, i.e., for each circuit, test generation was repeated #Arcs times. In Step 2, the backtrack limit was set to 64, which is not so large value. In Step 5, we used a simple algorithm to solve the ATSP and the processing time was negligibly short. Table 4.5 shows the test generation results of the respective methods. Columns

“#All”, “#Det” and “#Unt” give the number of total faults, detected faults and untestable faults, respectively. Columns “#TPT” and “#Vec” list the number of two-pattern tests and the length of the test sequence generated in Step 2. Columns “TGT [s]”, “FC [%]” ($= (\#Det/\#All) \times 100$) and “TAT [CC (clock cycles)]” denote test generation time, fault coverage and test application time, respectively. For reference, in Table 4.6, we list the test generation results of our method in more detail. Columns “#Tgt” and “#Abt” give the number of target faults and aborted faults in each step. Column “FC [%]” represents the cumulative results of fault coverage. In ES, there was no aborted fault during test generation, i.e., 100% fault efficiency was achieved in all the cases. Our method encountered no aborted fault in Step 3 for all the circuits. This implies that our method also achieved 100% fault efficiency. In Table 4.5, the value in each parenthesis represents the result in the case of removing untestable faults identified in Step 2 from the fault list of ES in advance. This can evaluate test application time in the both method fairly. Note that, in “TGT” of ES, the value in each parenthesis does not include the identification time for the removed untestable faults, and in “FC” of ES, the value in each parenthesis was calculated as $(\#Det/\#All) \times 100$. In the test generation results, the test generation time of our method was longer than that of ES because we used all the constraints in Step 1, and sequential test generation was performed. However, we achieved low fault coverage under 100% fault efficiency compared with that of ES. This means that ES detected faults that do not need to be tested, and our method alleviated over-testing compared with the enhanced scan method. Furthermore, we obtained shorter test application time. Unlike ES, we can perform at-speed test in our method. It implies that the actual test application time of our method becomes much shorter than that of ES. If it is assumed that the scan clock speed of ES is 1/5 as slow as the rated clock speed, the test application time of our method is 10 or more times faster, on average, than that of ES. Notice that, if we use one-hot encodings during logic synthesis, the advantage of our method will stand out further. This is because the test application time of ES depends on the number of ESFFs.

Finally, we mention the maximum delays of ISTGs. For every case, the maximum delay of the ISTG was smaller than that of the original circuit. This means

that our method can always apply tests at a rated clock speed.

4.5. Summary

In this chapter, we proposed a non-scan design scheme to enhance delay testability of controllers. In this scheme, the original behavior of a given STG is used during test application. For faults that cannot be detected by using the original behavior, we append an extra logic, called an invalid test state and transition generator, to the original controller. Our scheme can achieve short test application time and at-speed testing. We showed that our method is effective compared with scan-based methods by experiments.

Table 4.5. Test generation results.

Circuit name	#All	#Det		#Unt		#TPT		#Vec		TGT [s]		FC [%]		TAT [CC]	
		ES	NS	ES	NS	ES	NS	ES	NS	ES	NS	ES	NS	ES	NS
bbsse	782	782 (779)	760	0 (3)	22 (55)	57 (55)	82	95	0.2 (0.4)	6.5	100.00 (99.62)	97.44	574 (554)	269	
keyb	1,196	1,196 (1,194)	1,184	0 (2)	12 (110)	110 (110)	170	279	0.5 (0.7)	33.1	100.00 (99.83)	99.00	1,325 (1,325)	641	
kirkman	944	944 (944)	937	0 (0)	7 (90)	86 (90)	144	174	0.4 (0.6)	16.6	100.00 (100.00)	99.26	864 (904)	514	
planet	2,580	2,579 (2,578)	2,553	1 (2)	27 (115)	122 (115)	169	191	1.6 (1.9)	60.1	99.96 (99.92)	98.95	1,714 (1,616)	542	
s298	10,260	10,259 (10,259)	10,256	1 (1)	4 (564)	561 (564)	1,653	858	16.6 (17.5)	1,219.6	99.99 (99.99)	99.96	10,106 (10,160)	4,069	
s420	254	232 (232)	216	22 (22)	38 (24)	27 (24)	30	82	0.1 (0.1)	5.8	91.34 (91.34)	85.04	329 (293)	156	
sand	2,408	2,408 (2,405)	2,388	0 (3)	20 (142)	146 (142)	286	103	2.0 (1.1)	33.1	100.00 (99.88)	99.17	1,757 (1,709)	695	
scf	3,850	3,844 (3,838)	3,784	6 (12)	66 (175)	188 (175)	331	628	2.7 (3.6)	209.2	99.84 (99.69)	98.29	3,015 (2,807)	1,300	

Table 4.6. Detail of each step.

Circuit name		#Tgt	#Det	#Unt	#Abt	#TPT or #Vec	TGT [s]	FC [%]
bbsse	Step 1	782	612	170	0	80	4.2	78.26
	Step 2	170	132	32	6	95	2.3	95.14
	Step 3	6	6	0	0	2	0.0	97.44
keyb	Step 1	1,196	905	291	0	170	21.7	75.67
	Step 2	291	279	12	0	279	11.4	99.00
	Step 3	—	—	—	—	—	—	—
kirkman	Step 1	944	889	55	0	144	14.8	94.17
	Step 2	55	48	7	0	174	1.8	99.26
	Step 3	—	—	—	—	—	—	—
planet	Step 1	2,580	2,206	374	0	150	32.4	85.50
	Step 2	374	264	48	62	191	27.6	95.74
	Step 3	62	62	0	0	19	0.0	98.95
s298	Step 1	10,260	9,398	862	0	1,541	680.3	91.60
	Step 2	862	513	9	340	858	538.7	96.60
	Step 3	340	339	1	0	112	0.6	99.96
s420	Step 1	254	160	94	0	30	5.0	62.99
	Step 2	94	56	38	0	82	0.8	85.04
	Step 3	—	—	—	—	—	—	—
sand	Step 1	2,408	2,322	86	0	284	30.5	96.43
	Step 2	86	64	20	2	103	2.6	99.09
	Step 3	2	2	0	0	2	0.0	99.17
scf	Step 1	3,850	3,438	412	0	323	147.4	89.30
	Step 2	412	307	90	15	628	61.8	97.27
	Step 3	15	15	0	0	8	0.0	98.29

Chapter 5

Conclusions and Future Work

Delay test generation for sequential circuits is a challenging problem. To facilitate test generation for sequential circuits, design for testability (DFT) methods should be used. Although a fully enhanced scan method, which is a straightforward DFT for delay faults, can drastically reduce the test generation complexity of a given sequential circuit, there are some drawbacks in this method. In order to overcome the drawbacks, this dissertation tackled the problem by using the two techniques: partially enhanced scan and non-scan techniques.

In Chapter 3, we presented a new structure, called discontinuous reconvergence structure (DR-structure), of sequential circuits with easy testability for delay faults. We proposed a delay test generation method for sequential circuits with the structure. In our method, instead of a sequential delay test generation algorithm (ATPG), a combinational ATPG is used to generate test sequences for delay faults. We theoretically proved the correctness of the proposed method. Our method can handle several delay fault models in which faults can be detected by two-pattern tests, e.g., the path delay fault model, segment delay fault model and transition fault model. We confirmed that our test generation method can reduce test generation time and can enhance fault efficiency compared to the ordinary test generation method using a sequential ATPG. To apply our method to general sequential circuits, we used a partially enhanced scan technique. Theoretically, the class of sequential circuits with DR-structure properly includes that of balanced sequential circuits. Therefore, it is conceivable that DR-structure is extracted from a sequential circuit with low area overhead compared to balanced

structure). We also confirmed it experimentally. In Chapter 3, we targeted a logic level circuit. However, our method is also suitable for the data path part of a register transfer (RT) level circuit because the connectivity information at RT-level can be used for the proposed partially enhanced scan design before logic synthesis.

In Chapter 4, we proposed a non-scan design scheme to enhance delay testability of controllers. In this scheme, the original behavior of a given STG is used during test application. For faults that cannot be detected by using the original behavior, we append an extra logic, called an invalid test state and transition generator, to the original controller. Our scheme can achieve short test application time and at-speed testing. We showed that our method is effective compared with scan-based methods by experiments.

Finally, we discuss our future work. In the semiconductor industry, standard scan technique is widely used to test not only stuck-at faults but also delay faults. From a practical point of view, for delay faults, it is important to develop an effective method using standard scan technique, especially partial scan technique. In this work, we considered a controller and a data path as separate entities. However, an RT-level circuit is composed of a controller and a data path, and the two entities are connected each other. Currently, we do not take the delay between the two entities into account. Therefore, we should investigate a delay testing method for a whole RT-level circuit.

Acknowledgements

Many people have supported me during my Ph.D. studies¹. I would like to acknowledge their kind support.

I am very much grateful to my supervisor Professor Hideo Fujiwara for his gracious guidance and advice.

I would like to thank Professor Katsumasa Watanabe of NAIST (Nara Institute of Science and Technology) for his valuable comments.

I also would like to express my gratitude to Associate Professor Michiko Inoue of our laboratory for her important and helpful suggestions.

I am deeply indebted to Assistant Professor Satoshi Ohtake of our laboratory for his frequent, stimulating and helpful discussions.

I also wish to thank Assistant Professor Tomokazu Yoneda of our laboratory for his friendly discussions and encouragement.

I would like to thank Professor Toshimitsu Masuzawa of Osaka University for his helpful advice.

I am grateful to Professor Tomoo Inoue of Hiroshima City University for his friendly advice and useful comments.

I would like to thank Dr. Hiroshi Date of System JD, Associate Professor Toshinori Hosokawa of Nihon University, Mr. Masahide Miyazaki of STARC (Semiconductor Technology Academic Research Center) and Assistant Professor Hideyuki Ichihara of Hiroshima City University for their useful comments.

My thanks also go to the present and former members of our laboratory.

Finally, I wish to thank my parents and my sister for their continuing support and encouragement.

¹This work was supported in part by 21st Century COE (Center of Excellence) Program “Ubiquitous Networked Media Computing”.

References

- [1] Md. Altaf-Ul-Amin, S. Ohtake and H. Fujiwara, “Design for hierarchical two-pattern testability of data paths,” *Proc. 10th IEEE Asian Test Symp.*, pp. 11–16, 2001.
- [2] M. L. Bushnell and V. D. Agrawal, *Essentials of electronic testing for digital, memory & mixed-signal VLSI circuits*, Kluwer Academic Publishers, 2000.
- [3] S. Bose, P. Agrawal and V. D. Agrawal, “A rated-clock test method for path delay faults,” *IEEE Trans. on VLSI Systems*, Vol. 6, No. 2, pp. 323–331, June 1998.
- [4] A. Balakrishnan and S. T. Chakradhar, “Sequential circuits with combinational test generation complexity,” *Proc. 9th Int. Conf. on VLSI Design*, pp. 111–117, Jan. 1996.
- [5] T. J. Chakraborty, V. D. Agrawal and M. L. Bushnell, “Design for testability for path delay faults in sequential circuits,” *Proc. 30th ACM/IEEE Design Automation Conf.*, pp. 453–457, 1993.
- [6] T. J. Chakraborty, V. D. Agrawal and M. L. Bushnell, “On variable clock methods for path delay testing of sequential circuits,” *IEEE Trans. on CAD*, Vol. 16, No. 11, pp. 1237–1249, Nov. 1997.
- [7] T. J. Chakraborty, V. D. Agrawal and M. L. Bushnell, “Improving path delay testability of sequential circuits,” *IEEE Trans. on VLSI Systems*, Vol. 8, No. 6, pp. 736–741, Dec. 2000.

- [8] S. T. Chakradhar, A. Balakrishnan and V. D. Agrawal, "An exact algorithm for selecting partial scan flip-flops," *Proc. 31st ACM/IEEE Design Automation Conf.*, pp. 81–86, 1994.
- [9] K.-T. Cheng, S. Devadas and K. Keutzer, "A partial enhanced-scan approach to robust delay-fault test generation for sequential circuits," *Proc. Int. Test Conf.*, pp. 403–410, 1991.
- [10] K.-T. Cheng, "Transition fault testing for sequential circuits," *IEEE trans. on CAD*, Vol. 12, No. 12, pp. 1971–1983, Dec. 1993.
- [11] B. I. Dervisoglu and G. E. Stong, "Design for testability: using scanpath techniques for path-delay test and measurement," *Proc. Int. Test Conf.*, pp. 365–374, 1991.
- [12] H. Fujiwara, *Logic testing and design for testability*, The MIT press, 1985.
- [13] H. Fujiwara, "A new class of sequential circuits with combinational test generation complexity," *IEEE Trans. on Computers*, Vol. 49, No. 9, pp. 895–905, Sep. 2000.
- [14] I. Ghosh, A. Raghunath and N. K. Jha, "Design for hierarchical testability of RTL circuits obtained by behavioral synthesis," *IEEE Trans. on CAD*, Vol. 16, No. 9, pp. 1001–1014, Sep. 1997.
- [15] R. Gupta, R. Gupta and M. A. Breuer, "The BALLAST methodology for structured partial scan design," *IEEE Trans. on Computers*, Vol. 39, No. 4, pp. 538–544, Apr. 1990.
- [16] K. Heragu and V. D. Agrawal, "Segment delay faults: a new fault model," *Proc. 14th IEEE VLSI Test Symp.*, pp. 32–39, 1996.
- [17] T. Hosokawa, T. Inoue, T. Hiraoka and H. Fujiwara, "Static and dynamic test sequence compaction methods for acyclic sequential circuits using a time expansion model," *Proc. 8th IEEE Asian Test Symp.*, pp. 192–199, 1999.
- [18] T. Inoue, T. Hosokawa, T. Mihara and H. Fujiwara, "An optimal time expansion model based on combinational ATPG for RT level circuits," *Proc. 7th IEEE Asian Test Symp.*, pp. 190–197, 1998.

- [19] Y. C. Kim, V. D. Agrawal and K. K. Saluja, "Combinational test generation for various classes of acyclic sequential circuits," *Proc. Int. Test Conf.*, pp. 1078–1087, 2001.
- [20] A. Krstić and K.-T. Cheng, *Delay fault testing for VLSI circuits*, Kluwer Academic Publishers, 1998.
- [21] A. Kunzmann and H.-J. Wunderlich, "An analytical approach to the partial scan problem," *Journal of Electronic Testing: Theory and Applications*, Vol. 1, No. 2, pp. 163–174, May 1990.
- [22] M. T.-C. Lee, *High-level test synthesis of digital VLSI circuits*, Artech House, 1997.
- [23] S. Majumder, V. D. Agrawal and M. L. Bushnell, "Path delay testing: variable-clock versus rated-clock," *Proc. 11th Int. Conf. on VLSI Design*, pp. 470–475, 1998.
- [24] S. Ohtake, T. Masuzawa and H. Fujiwara, "A non-scan approach to DFT for controllers achieving 100% fault efficiency," *Journal of Electronic Testing: Theory and Applications*, Vol. 16, No. 5, pp. 553–566, Oct. 2000.
- [25] S. Ohtake, S. Miwa and H. Fujiwara, "A method of test generation for path delay faults in balanced sequential circuits," *Proc. 20th IEEE VLSI Test Symp.*, pp. 321–327, 2002.
- [26] G. L. Smith, "Model for delay faults based upon paths," *Proc. Int. Test Conf.*, pp. 342–349, 1985.
- [27] J. Savir and S. Patil, "Scan-based transition test," *IEEE Trans. on CAD*, Vol. 12, No. 8, pp. 1232–1241, Aug. 1993.
- [28] J. Savir and S. Patil, "Broad-side delay test," *IEEE Trans. on CAD*, Vol. 13, No. 8, pp. 1057–1064, Aug. 1994.
- [29] J. Saxena, K. M. Butler, V. B. Jayaram and S. Kundu, "A case study of IR-drop in structured at-speed testing," *Proc. Int. Test Conf.*, pp. 1098–1104, 2003.

- [30] T. Villa, T. Kam, R.K. Brayton and A. Sangiovanni-Vincentelli, *Synthesis of finite state machines: logic optimization*, Kluwer Academic Publishers, 1997.
- [31] H. Wada, T. Masuzawa, K. K. Saluja and H. Fujiwara, “Design for strong testability of RTL data paths to provide complete fault efficiency,” *Proc. 13th Int. Conf. on VLSI Design*, pp. 300–305, 2000.