Acceleration of Transition Test Generation for Acyclic Sequential Circuits Utilizing Constrained Combinational Stuck-at Test Generation

Tsuyoshi Iwagaki Satoshi Ohtake Hideo Fujiwara

Graduate School of Information Science, Nara Institute of Science and Technology
Kansai Science City 630-0192, Japan
E-mail: {*tsuyo-i, ohtake, fujiwara*}*@is.naist.jp*

Abstract

This paper presents a transition test generation method for acyclic sequential circuits. In this method, to generate test sequences for transition faults in a given acyclic sequential circuit, constrained combinational stuck-at test generation is performed on its double time-expansion model that is composed of two copies of a time-expansion model of the given circuit. This method is complete, i.e., this method can generate test sequences for all the testable transition faults and can identify all the untestable transition faults in a given acyclic sequential circuit. Experimental results show that our method can achieve higher fault efficiency with drastically shorter test generation time than that achieved by a conventional method.

1 Introduction

Test generation for sequential circuits is generally a hard problem even if their circuit sizes are not so large. During sequential test generation, much time is consumed by the task of identifying sequentially untestable faults. It is impossible to identify all the sequentially untestable faults in a large circuit. A concept of *circuit pseudo-transformation (CPT)* has been proposed to facilitate test generation for sequential circuits [1]. The CPT changes a given sequential circuit into a different circuit that is easily testable compared to the original one only when test generation is performed. Several test generation methods based on this concept have been proposed for stuck-at faults [2, 3, 4, 5] and for delay faults [6, 7].

For the stuck-at fault model, *internally balanced structure* has been proposed as an easily testable circuit structure in [4]. The stuck-at fault testability of a sequential circuit with internally balanced structure is preserved in its combinationally equivalent circuit. That is, test sequences for all the testable stuck-at faults in a sequential circuit with internally balanced structure can be generated, and all the untestable stuck-at faults in the circuit can be identified by applying a combinational stuck-at fault test generation algorithm (ATPG) to its combinationally equivalent circuit. Test generation methods for an *acyclic sequential circuit*, which is a super class of sequential circuits with internally balanced structure, have been proposed [2, 3, 5]. To generate test sequences for a given acyclic sequential circuit, these methods use a transformed combinational circuit in which the function and timing behavior of the given circuit are simulated. Under the CPTs used in [2, 3, 5], the stuck-at fault testability of an acyclic sequential circuit is also preserved in its transformed combinational circuit.

Similarly, for the delay fault model, two test generation methods based on CPT have been proposed [6, 7]. In [6], it was shown that the delay fault testability of a *balanced sequential circuit* is preserved in its combinationally equivalent circuit. For an acyclic sequential circuit, unlike the case of the stuck-at fault model, a sequential delay fault ATPG is required to generate test sequences [8]. This implies that it is hard to achieve high fault efficiency with short test generation time. In [7], it was shown that the delay fault testability of an acyclic sequential circuit is not preserved in its *time-expansion model* [3]. Moreover, Iwagaki *et al.* proposed a

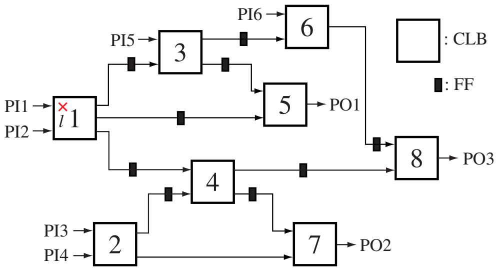

Figure 1: Acyclic sequential circuit: S

subclass of acyclic sequential circuits whose delay fault testability is preserved in its time-expansion model [7]. The circuit structure is called *discontinuous reconvergence (DR) structure*. In consequence of restricting circuit structure, we can use a combinational delay fault ATPG.

For stuck-at faults and delay faults, *partial scan technique* and *Partially enhanced scan technique* can be used as straightforward methods to apply the above test generation methods based on CPT to a general sequential circuit, respectively. Given a sequential circuit, in partial scan technique, some flip-flops (FFs) are replaced by scan FFs such that its *kernel*, which is the circuit excluding the scan path, become a desired circuit structure. In partially enhanced scan technique, enhanced scan FFs [9] are used instead of scan FFs. Hardware overheads in these design methods depend on which circuit structure is used for a kernel. If acyclic structure is used for a kernel, its hardware overhead is the smallest of the circuit structures mentioned above.

In this paper, we present a transition test generation method for acyclic sequential circuits. In this method, to generate tests for transition faults in a given acyclic sequential circuit, constrained combinational stuck-at test generation is performed on its double time-expansion model that is composed of two copies of a time-expansion model of the given circuit. That is, unlike the method proposed in [7], we do not restrict circuit structure but modify its test generation model for a given acyclic sequential circuit. As a result, an increase of hardware overhead is not incurred. This method is complete, i.e., this method can generate tests for all the testable transition faults, and can identify all the untestable transition faults. We show that, by some experiments, our method can achieve higher fault efficiency with drastically shorter test generation time than that obtained by a conventional sequential test generation method.

2 Preliminaries

2.1 Target Circuit and Fault Model

A sequential circuit generally consists of combinational logic blocks (CLBs) connected with each other directly or through FFs. A CLB is a region of connected combinational logic gates. This paper treats *acyclic sequential circuits* in which there is no cyclic path. For example, an acyclic sequential circuit S is shown in Figure 1. In this paper, we assume that FFs are of D-type. This assumption does not impose restrictions on circuit representation because the other types of FFs can be modeled by a D-type FF and some logic gates. Note that although a general sequential circuit is not acyclic, the circuit can be made acyclic by using some techniques, e.g., enhanced scan technique.

Our target faults are *transition faults* in acyclic sequential circuits. There are two transition faults associated with each line in an acyclic sequential circuit: 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 through the faulty site from reaching any FF or any primary output within a specified time. In this paper, it is assumed that transition faults in acyclic sequential circuits are tested in the slow-fast-slow testing manner [10]. Under this assumption, we can consider a sequential circuit to be delay fault-free in both the fault initialization

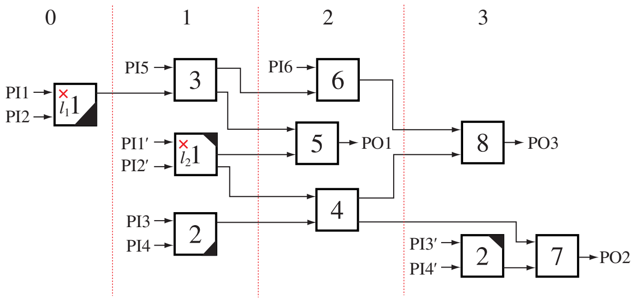

Figure 2: Time-expansion model of S: $C(S)$

and the fault effect propagation phases. Note that if a transition fault is testable under the at-speed testing manner, the fault is also testable under the slow-fast-slow testing manner [10]. Hence, the slow-fast-slow testing never misses any testable fault in the at-speed testing.

2.2 Time-Expansion Model

In this subsection, we mention a *time-expansion model (TEM)* [3], which is used in our test generation method. Then, we describe the correspondence between a transition fault in an acyclic sequential circuit and a transition fault in its TEM. We also explain the correspondence between a two-pattern test for a TEM and a test sequence for its original circuit.

A TEM of a given acyclic sequential circuit is a combinational circuit in which the function and timing behavior of the given circuit are simulated. Figure 2 is a TEM $C(S)$ of the circuit S shown in Figure 1. TEM $C(S)$ is a combinational circuit derived by connecting CLBs according to their sequential depths. A sequential depth between two CLBs is defined as the number of FFs on a path between them. If a CLB has paths to another CLB in S whose sequential depths are different, the CLB is duplicated in $C(S)$. For example, in Figure 1, since CLB 1 has two paths to CLB 5 whose sequential depths (one and two) are different, CLB 1 is duplicated in $C(S)$ (Figure 2). A shaded part of a CLB in Figure 2 represents a portion of the lines and gates removed. There is no path from the portion to any input of CLBs or any primary output of $C(S)$. The number placed at the top of each column in Figure 2 is the label of CLBs in the column. The label of a CLB v is denoted as $t(v)$.

A transition fault in an acyclic sequential circuit is mapped into a single or a multiple transition fault in its TEM. For example, a transition fault associated with a line l in CLB 1 of S (Figure 1) is mapped into a multiple fault whose respective faults exist in the duplicated CLBs of $C(S)$ (Figure 2) if the corresponding lines l_1, l_2 are not removed. Note that since we use the slow-fast-slow testing manner during test application, we can handle respective transition faults in a multiple transition fault one by one.

Here, we briefly describe how a two-pattern test for a TEM is transformed into a test sequence for its original circuit. A two-pattern test for a TEM is transformed into a test sequence for its original circuit on the basis of the information about the label of each primary input in the TEM. For example, suppose that a two-pattern test for $C(S)$ shown in Figure 2: $\mathrm{PI1} = (v_1^{\mathrm{PI1}}, v_2^{\mathrm{PI1}})$, $\mathrm{PI2} = (v_1^{\mathrm{PI2}}, v_2^{\mathrm{PI2}})$, $\mathrm{PI1}' = (v_1^{\mathrm{PI1}'}, v_2^{\mathrm{PI1}'})$, $\mathrm{PI2}' = (v_1^{\mathrm{PI2}'}, v_2^{\mathrm{PI2}'})$, $\mathrm{PI3} = (v_1^{\mathrm{PI3}}, v_2^{\mathrm{PI3}})$, $\mathrm{PI4} = (v_1^{\mathrm{PI4}}, v_2^{\mathrm{PI4}})$, $\mathrm{PI5} = (v_1^{\mathrm{PI5}}, v_2^{\mathrm{PI5}})$, $\mathrm{PI6} = (v_1^{\mathrm{PI6}}, v_2^{\mathrm{PI6}})$, and the corresponding responses: $\mathrm{PO1} = (v_1^{\mathrm{PO1}}, v_2^{\mathrm{PO1}})$, $\mathrm{PO2} = (v_1^{\mathrm{PO2}}, v_2^{\mathrm{PO2}})$, $\mathrm{PO3} = (v_1^{\mathrm{PO3}}, v_2^{\mathrm{PO3}})$, are given. From the label information of $C(S)$, if $v_2^{\mathrm{PI1}} = v_1^{\mathrm{PI1}'}$ and $v_2^{\mathrm{PI2}} = v_1^{\mathrm{PI2}'}$, this two-pattern test is transformed into the test sequence for S (Figure 1) shown in Table 1. Note that the above transformation is formally defined in [7].

Table 1: Input and output sequences

Time	0	1	2	3	4
PI1	v_1^{PI1}	$v_2^{\mathrm{PI1}} = v_1^{\mathrm{PI1}'}$	$v_2^{\mathrm{PI1}'}$	X	X
PI2	v_1^{PI2}	$v_2^{\mathrm{PI2}} = v_1^{\mathrm{PI2}'}$	$v_2^{\mathrm{PI2}'}$	X	X
PI3	X	v_1^{PI3}	v_2^{PI3}	$v_1^{\mathrm{PI3}'}$	$v_2^{\mathrm{PI3}'}$
PI4	X	v_1^{PI4}	v_2^{PI4}	$v_1^{\mathrm{PI4}'}$	$v_2^{\mathrm{PI4}'}$
PI5	X	v_1^{PI5}	v_2^{PI5}	X	X
PI6	X	X	v_1^{PI6}	v_2^{PI6}	X
PO1	X	X	v_1^{PO1}	v_2^{PO1}	X
PO2	X	X	X	v_1^{PO2}	v_2^{PO2}
PO3	X	X	X	v_1^{PO3}	v_2^{PO3}

3 Proposed Method

3.1 Motivation and Main Ideas

As shown in Table 1, we can obtain a test sequence for an acyclic sequential circuit only if, for its TEM, the second vector of a primary input u and the first vector of a primary input v that satisfy $t(v) - t(u) = 1$ are the same value. This limitation is induced by the fact that a TEM of an acyclic sequential circuit does not include information about its pattern dependency such as $v_2^{\mathrm{PI1}} = v_1^{\mathrm{PI1}'}$ and $v_2^{\mathrm{PI2}} = v_1^{\mathrm{PI2}'}$ in Table 1. Thus, it is not sufficient to use only a TEM to generate test sequences for its original circuit that is acyclic.

In order to generate transition tests for an acyclic sequential circuit, we define the following test generation model that includes information about its pattern dependency.

Definition: Let S be an acyclic sequential circuit, and $C(S)$ be a TEM of S. Then, a combinational circuit obtained by the following procedure is said to be a *double time-expansion model (DTEM)* $C^*(S)$ of S.

S1: Make two copies of $C(S)$: $C^*_{V1}(S)$, $C^*_{V2}(S)$.

S2: Connect any pair of primary inputs u in $C^*_{V1}(S)$ and v in $C^*_{V2}(S)$ such that the difference between those label values, $t(v) - t(u)$, with each other, and feed a new primary input w into them. □

Figure 3 shows a DTEM $C^*(S)$ of S (Figure 1). In Figure 3, $C^*(S)$ is composed of two copies of $C(S)$ (Figure 2): $C^*_{V1}(S)$, $C^*_{V2}(S)$, and $\mathrm{PI1}_{V2=V1}$ and $\mathrm{PI2}_{V2=V1}$ are created according to **S2** of "Definition". A single pattern for $C^*_{V1}(S)$ (resp. $C^*_{V2}(S)$) corresponds to the first (resp. second) vector of a vector pair for $C(S)$. Note that, in Figure 3, a single pattern for $\mathrm{PI1}_{V2=V1}$ and $\mathrm{PI2}_{V2=V1}$ corresponds to both the second vector for PI1 and PI2 and the first vector for PI1$'$ and PI2$'$ in Figure 2.

Transition test generation is similar to stuck-at test generation. A two-pattern test (V_1, V_2) for the slow-to-rise (resp. slow-to-fall) fault associated with a line l in a combinational circuit has the following two properties:

1. the value of 0 (resp. 1) is justified to l by V_1, and
2. the stuck-at 0 (resp. 1) fault associated with l is detected by V_2.

According to these properties, we can generate the first vector and the second vector of a two-pattern test separately. Here, let us consider applying the above properties to a DTEM. If we generate a two-pattern test for the slow-to-rise (resp. slow-to-fall) fault associated with a line l in a TEM $C(S)$, we perform stuck-at test generation for the stuck-at 0 (resp. 1) fault associated with a corresponding line l_{V2} in $C^*_{V2}(S)$ under the following constraint: the value of the corresponding line l_{V1} in $C^*_{V1}(S)$ is set to 0 (resp. 1). For example, in

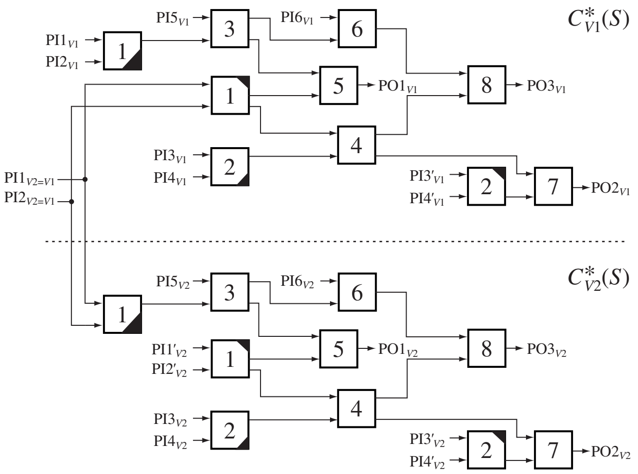

Figure 3: Double time-expansion model of S: $C^*(S)$

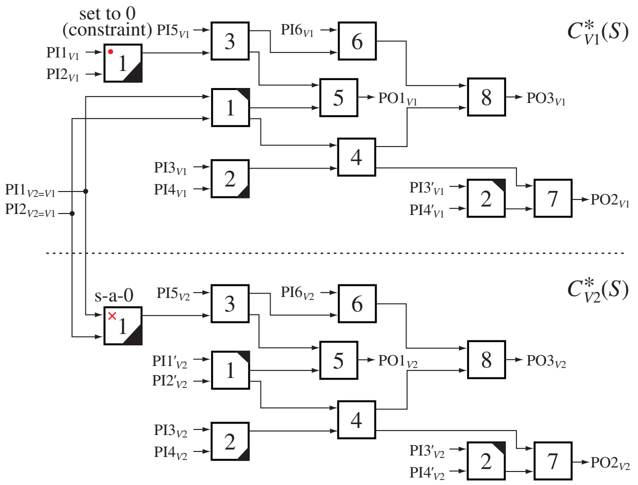

Figure 4: Double time-expansion model with a constraint

order to generate a two-pattern test for the slow-to-rise fault associated with a line l_1 in CLB 1 (Figure 2), we perform stuck-at test generation for the DTEM with a constraint shown in Figure 4. From the above discussion, we see that a test sequence for a transition fault in an acyclic sequential circuit can be generated by performing constrained combinational stuck-at test generation on its DTEM.

The correctness of the above test generation is guaranteed by the following theorem.

Theorem: Let S and $C^*(S)$ be an acyclic sequential circuit and a DTEM of S, respectively. Let f_t^S be the slow-to-rise (resp. slow-to-fall) fault associated with a line l in S. Let $F_s^{C^*(S)}$ be the set of stuck-at 0 (resp. 1) faults associated with the set of lines L_{V2} in $C^*_{V2}(S)$ corresponding to l. Then, (i) f_t^S is testable if and only if at least one $f_s^{C^*(S)} \in F_s^{C^*(S)}$ associated with $l_{V2} \in L_{V2}$ is testable under the following condition:

- an objective $(0, l_{V1})$ (resp. $(1, l_{V1})$) in $C^*_{V1}(S)$ must be satisfied, i.e., the value of 0 (resp. 1) has to be justified to l_{V1} in $C^*_{V1}(S)$, where l_{V1} is the line corresponding to l_{V2}.

Furthermore, (ii) a test pattern generated for $f_s^{C^*(S)}$ can always be transformed into a test sequence for f_t^S.

Sketch of proof: We demonstrate (i) of "Theorem" first. Under the slow-fast-slow testing manner, we can treat stuck-at faults in $F_s^{C^*(S)}$ one by one. Hence, it is sufficient to consider whether at least one $f_s^{C^*(S)}$ is testable. The transition fault testability of S is not preserved in $C(S)$ [7]. This is because $C(S)$ does not include information about the pattern dependency in S. Unlike $C(S)$, $C^*(S)$ includes information about the pattern dependency. Consequently, $C^*(S)$ can simulate the function and timing behavior of S. By the similar discussion in [7], (i) of "Theorem" can be demonstrated.

Then, we can easily see that (ii) of "Theorem" is true because $C^*(S)$ includes information about the pattern dependency in S. □

3.2 Test Generation Procedure

All the testable transition faults can be tested, and all the untestable transition faults can be identified by performing test generation based on only "Theorem". Furthermore, in order to perform test generation more efficiently, we use the following test generation procedure. In the following procedure, for the sake of efficiency, we make use of the fact that a necessary condition to detect the transition fault associated with a line is that the corresponding stuck-at fault on the line is detectable. That is, we perform stuck-at test generation for a TEM of a given acyclic sequential circuit, then construct vector pairs for transition faults in the TEM from test patterns for stuck-at faults. This additional step aims to detect many transition faults before performing test generation based on "Theorem".

Given an acyclic sequential circuit S, our method is performed as follows.

Main Procedure

S1: Create a transition fault list F_t^S of S.

S2: Construct a TEM $C(S)$ of S and a DTEM $C^*(S)$ of S.

S3: Perform a sub procedure *FAULT_DROPPING*.

Until F_t^S is empty, iterate **S4**.

S4: For each remaining fault in F_t^S, the following steps are performed.

- **(a):** Perform test generation based on (i) of "Theorem".
- **(b):** Convert a test pattern $t_s^{C^*(S)}$ generated in **S4(a)** into a two-pattern test $t_t^{C(S)}$ for $C(S)$.
- **(c):** Perform transition fault simulation by applying $t_t^{C(S)}$ to $C(S)$.
- **(d):** Add $t_t^{C(S)}$ to $T_t^{C(S)}$.
- **(e):** Drop all the corresponding transition faults detected in **S4(c)** from F_t^S.

S5: Convert $T_t^{C(S)}$ into a transition test set T_t^S for S.

Sub Procedure *FAULT_DROPPING*

(a): Create a stuck-at fault list $F_s^{C(S)}$ of $C(S)$.

(b): Generate a stuck-at test set $T_s^{C(S)}$ for $F_s^{C(S)}$ by using a combinational stuck-at fault ATPG.

(c): Convert $T_s^{C(S)}$ into a vector pair set $V_t^{C(S)}$ for $C(S)$.

(d): Perform transition fault simulation by applying $V_t^{C(S)}$ to $C(S)$.

Table 2: Circuit characteristics

Circuit name	#PIs	#POs	#FFs	Area
C1*	32	48	56	5,292
C2*	48	56	88	5,911
C3*	256	224	160	19,923

(e): Add valid vector pairs, which detect some transition faults in **(d)**, to a transition test set $T_t^{C(S)}$ for $C(S)$.

(f): Drop all the corresponding transition faults detected in **(d)** from F_t^S.

Here, we explain some steps in the above procedure in details. In **S3(b)**, stuck-at test generation is performed. As mentioned above, a necessary condition to detect the transition fault associated with a line is that the corresponding stuck-at fault on the line is detectable. Hence, if all the stuck-at faults corresponding to a transition fault in S are identified as untestable, the transition fault is also untestable. This step is useful in achieving efficient test generation. Note that, for a given sequential circuit, if stuck-at fault testing is performed by using the method proposed in [3] in advance, $T_s^{C(S)}$ can be obtained without performing test generation in **S3(b)**. In **S3(c)**, we convert $T_s^{C(S)}$ into $V_t^{C(S)}$ for $C(S)$ such that pattern conflict does not occur. During pattern conversion, each first vector in $V_t^{C(S)}$ is basically derived as the complement of its corresponding pattern in $T_s^{C(S)}$ such that a transition occurs. In [11], Liu *et al.* have proposed an efficient method to obtain two-pattern tests for transition faults from test patterns for stuck-at faults. This method could be applied to **S3(c)** with some modification. In **S4**, if all the duplicated stuck-at faults corresponding to a transition fault in the original circuit are identified as untestable, the transition fault is also untestable. Notice that if one of all the duplicated stuck-at faults corresponding to a transition fault in the original circuit is detected, we do not need to consider the other duplicated faults. Although, in this step, we need to perform test generation under constraints, commercial ATPG tools can usually support such kind of constraints.

4 Experimental Results

In this section, we evaluate the proposed method in terms of hardware overhead, test generation time, test application time and fault efficiency.

The following experiment was carried out on a Sun Blade 2000 workstation. To perform test generation and fault simulation, TetraMAX ATPG (Synopsys) was used as a combinational and sequential ATPG tool with a backtrack limit of 100. We used acyclic versions of the circuits reported in [7]. The characteristics of these circuits are shown in Table 2. In Table 2, Columns “#PIs”, “#POs” and “#FFs” denote the number of primary inputs, primary outputs and FFs, respectively. Column “Area” denotes the area of a circuit estimated by Design Compiler (Synopsys).

First, we evaluate the hardware overhead needed to make a given circuit acyclic by using enhanced scan technique. The result of hardware overheads reported in [7] shows that, by using acyclic structure as a kernel structure, an about 9.5% reduction on average (a 11.2% reduction in the best case) was achieved compared with the case of using DR-structure. Thus, our method is effective in hardware overhead compared to the previous method.

Next, we show the test generation results. Table 3 lists the test generation results obtained by sequential test generation for transition faults in the respective circuits, denoted by “Sequential ATPG”, and by our proposed test generation procedure, denoted by “Our method”. Columns “#faults”, “#det”, “#unt” and “#abt” give the number of target transition faults, detected faults, identified untestable faults and aborted faults, respectively. Columns “TGT [s]”, “FE [%]” and “TAT [clock cycles (CC)] denote test generation time which includes fault simulation time, fault efficiency and test application time, respectively. The fault efficiency is defined as $100 \times$

Table 3: Test generation results

Circuit name	Method	#faults	#det	#unt	#abt	TGT [s]	FE [%]	TAT [CC]
C1*	Sequential ATPG	12,238	11,724	26	488	131.45	96.01	487
	Our method		12,194	41	3	6.76	99.98	1,190
C2*	Sequential ATPG	13,018	10,900	23	2095	1,122.03	84.01	221
	Our method		12,875	26	117	13.76	99.10	1,008
C3*	Sequential ATPG	48,032	43,658	0	4,874	4,709.04	90.91	189
	Our method		45,885	0	2,147	195.40	95,53	1,737

Table 4: Disaggregated results of our method

Circuit name	**S3** (*FAULT_DROPPING*)		**S4** ("Theorem")		Overall	
	TGT [s]	FE [%]	TGT [s]	FE [%]	TGT [s]	FE [%]
C1*	4.21	97.21	2.55	99.98	6.76	99.98
C2*	3.98	90.96	9.78	99.10	13.76	99.10
C3*	38.09	54.44	157.31	95.53	195.40	95.53

$(\#\text{det} + \#\text{unt})/\#\text{faults}$. The test application time of our method is calculated by $(d+1)\cdot n_{\text{TPT}}$, where n_{TPT} is the number of generated two-pattern tests and d is the maximum sequential depth of a given acyclic sequential circuit. In the conventional method ("Sequential ATPG"), the test application time is the length of a generated test sequence. Note that we do not consider scan shift operation here. From Table 3, we can see that our method outperformed the conventional method in test generation time as well as fault efficiency. In the best case, our method achieved 15.09% higher fault efficiency with about $1/82$ shorter test generation time than that of the conventional method. Table 3 also shows our method resulted in long test application time compared to that of the conventional method. However, this dose not mean that our method is ineffective in test application time because our method detected more faults compared with the conventional method. Furthermore, if we use the method reported in [12], we can reduce the test application time.

Finally, we analyze the test generation results of our method. Table 4 shows the disaggregated results of the proposed method. Columns "**S3** (*FAULT_DROPPING*)" and "**S4** ("Theorem")" give the intermediate results in **S3** of our method and **S4**, respectively. In this table, "FE [%]" denotes the cumulative result of fault efficiency. In columns "**S3** (*FAULT_DROPPING*)" and "**S4** ("Theorem")" "TGT [s]" denotes test generation time which includes fault simulation time in each step. For C1* and C2*, **S3** worked well, i.e., a large number of transition faults were detected in **S3**. Furthermore, even if some faults are missed in **S3**, we can detect almost all the faults by performing **S4**.

5 Conclusions and Future Work

In this paper, we presented a transition test generation method for acyclic sequential circuits. In this method, to generate test sequences for transition faults in a given acyclic sequential circuit, constrained combinational stuck-at test generation is performed on its double time-expansion model that is composed of two copies of a time-expansion model of the given circuit. This method can generate test sequences for all the testable transition faults and can identify all the untestable transition faults in a given acyclic sequential circuit. In this paper, on the basis of circuit pseudo-transformation (CPT), we accelerated test generation. Moreover, CPT made the use of an efficient procedure such as *FAULT_DROPPING* possible. We showed that our method is effective in test generation time, fault efficiency and hardware overhead by experiments.

Our future work is to extend the proposed method to be able to handle the path delay fault model.

Acknowledgments

This work was supported in part by JSPS (Japan Society for the Promotion of Science) under Grants-in-Aid for Scientific Research B(2) (No. 15300018).

References

[1] S. Ohtake, T. Inoue and H. Fujiwara, "Sequential test generation based on circuit pseudo-transformation," *Proc. IEEE 6th Asian Test Symp.*, pp. 62–67, 1997.

[2] A. Kunzmann and H.-J. Wunderlich, "An analytical approach to the partial scan problem," *J. of Electronic Testing: Theory and Appl.*, Vol. 1, No. 2, pp. 163–174, May 1990.

[3] T. Inoue, T. Hosokawa, T. Mihara and H. Fujiwara, "An optimal time expansion model based on combinational ATPG for RT level circuits," *Proc. 7th Asian Test Symp.*, pp. 190–197, 1998.

[4] 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.

[5] 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.

[6] 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.

[7] T. Iwagaki, S. Ohtake and H. Fujiwara, "Reducibility of sequential test generation to combinational test generation for several delay fault models," *Proc. IEEE 12th Asian Test Symp.*, pp. 58–63, 2003.

[8] T. J. Chakraborty, V. D. Agrawal and M. L. Bushnell, "Improving path delay testability of sequential circuits," *IEEE Trans. VLSI Systems*, Vol. 8, No. 6, pp. 736–741, Dec. 2000.

[9] Y. K. Malaiya and R. Narayanaswamy, "Testing for timing faults in synchronous sequential integrated circuits," *Proc. Int. Test Conf.*, pp. 560–571, 1983.

[10] A. Krstić and K.-T. Cheng, *Delay fault testing for VLSI circuits*, Kluwer Academic Publishers, 1998.

[11] X. Liu, M. S. Hsiao, S. Chakravarty and P. J. Thadikaran, "Efficient transition fault ATPG algorithms based on stuck-at test vectors," *J. of Electronic Testing: Theory and Appl.*, Vol. 19, No. 4, pp. 437–445, Aug. 2003.

[12] 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.

[13] M. Abadir and J. Zhu, "Transition test generation using replicate-and-reduce transform for scan-based designs," *Proc. 21st VLSI Test Symp.*, pp. 22–27, 2003.